Published in ICLR2021

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

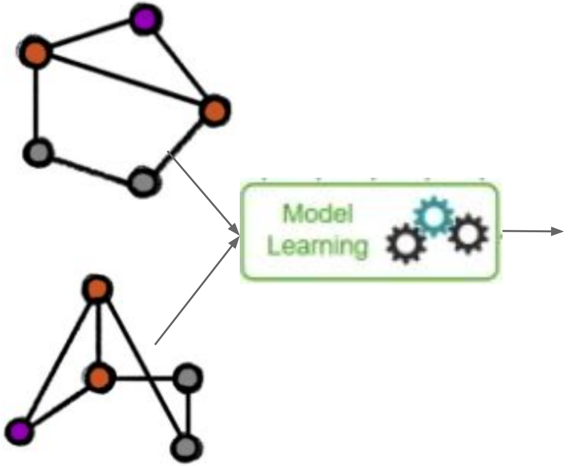
Muhammet Balcilar, Guillaume Renton, Pierre Héroux,

Benoit Gaüzère, Sébastien Adam, Paul Honeine

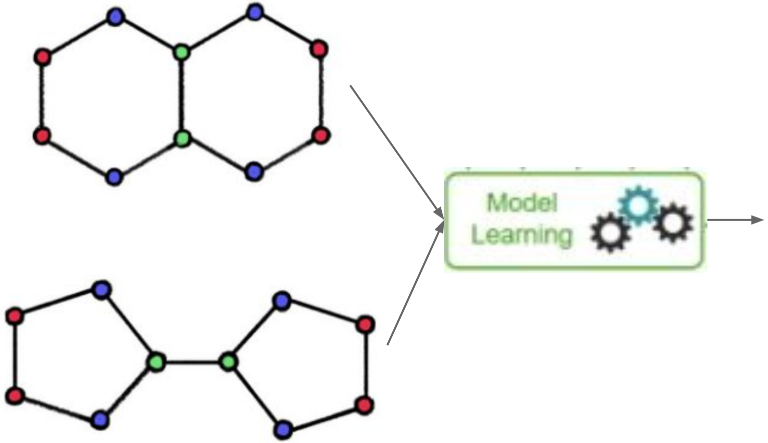
Table of Contents

Expressive Power of GNN

- Universality of the GNN depends on
  - ability to produce same output for isomorphic graphs (invariance).
  - ability to produce different output for non-isomorphic graphs.



should be same



should be different

Graphs are taken from Expressive power of graph neural networks and the Weisfeiler-Lehman test By M. Bronstein

Expressive Power of GNN

- WL test iteratively passes the node color to its neighborhood.
- 1-WL=2-WL <3-WL<4-WL<......<k-WL
- We can classify GNN by equivalence of WL test order
- k>2, k-WL GNN needs
  - O(n^(k-1)) memory
  - O(n^k) CPU time

MPNN (i.e 1-WL equivalent GNNs)

- MPNN are still attractive because of ;
  - Linear memory&time complexity.
  - Natural problems consist of graphs can be distinguishable by 1-WL.
  - 300 out of 61M graphs pairs are not indistinguishable by 1-WL GNN.
  - Their results are still state of the art!

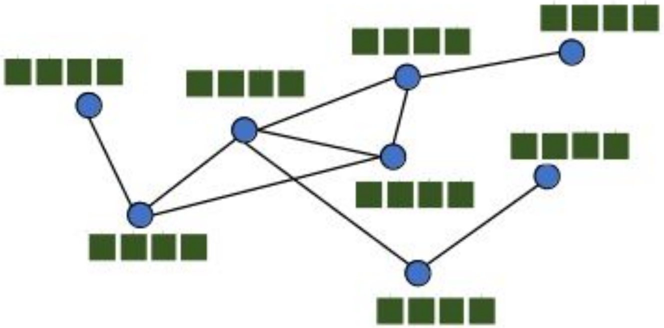
- WL test order cannot tell any superiority between MPNNs
- We need another perspective to evaluate MPNN's expressive power.

What is MPNN

$X \in \mathbb{R}^{n \times d}$ is node features

$E \in \mathbb{R}^{n \times n \times e}$ is edge features

$A \in \mathbb{R}^{n \times n}$ is adjacency matrix.



Adjacency Matrix

$$X \rightarrow \begin{bmatrix} \mathrm{x}(1) \\ \mathrm{x}(2) \\ \mathrm{x}(3) \\ \mathrm{x}(4) \\ \mathrm{x}(5) \\ \mathrm{x}(6) \\ \mathrm{x}(7) \\ \mathrm{x}(8) \end{bmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

A

MPNN finds new representation

$$H^{(l+1)} = f(H^{(l)}, A, E)$$

$$H^{(0)} = X$$

Spatial MPNN

- Forward calculation of one layer Spatial MPNN

agg aggregates the neighborhood nodes.

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

upd updates the concerned node

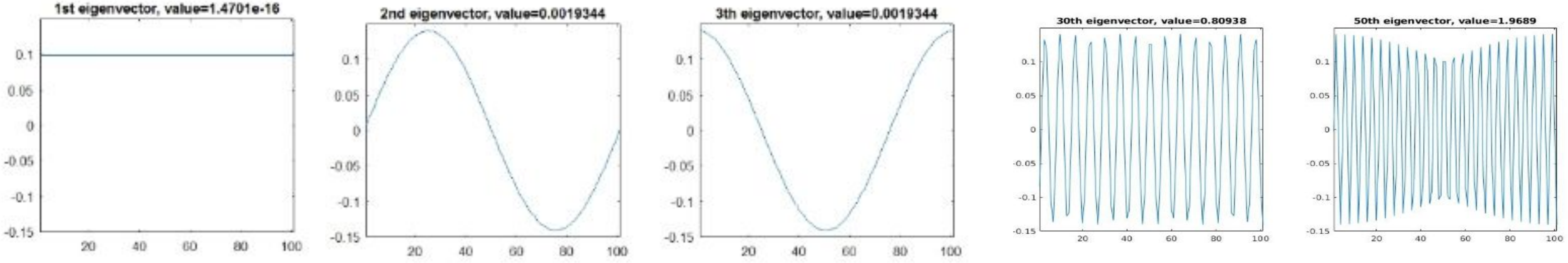
$g_0, g_1 : \mathbb{R}^{n \times f_l} \to \mathbb{R}^{n \times f_{l+1}}$ trainable models. $\mathcal{N}(v)$ is the set of neighborhood nodes

Graph Signal Processing

Laplacian of Graph $L = I - D^{-1/2} A D^{-1/2}$

Eigenvector and eigenvalues $U^{-1} L U = diag(\lambda_1, \lambda_2, \dots, \lambda_n)$

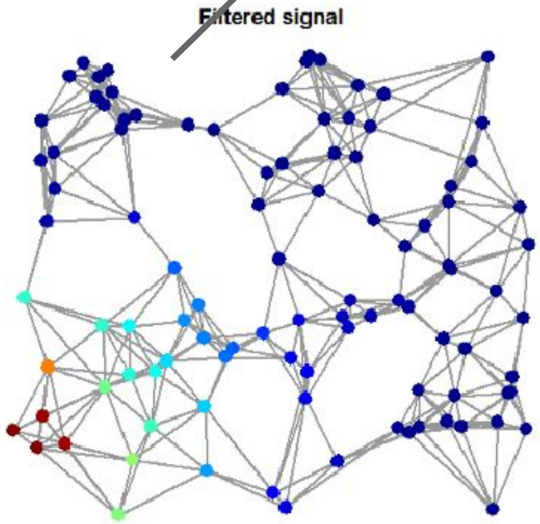
$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

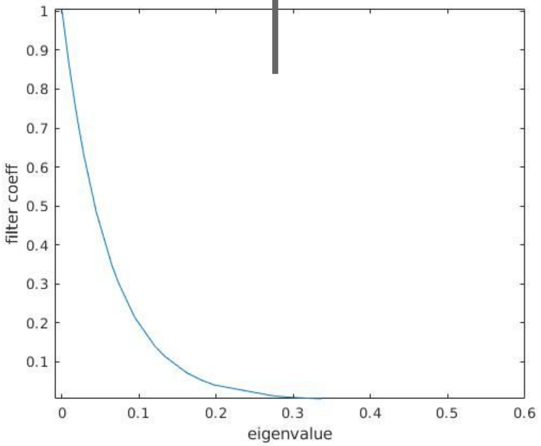


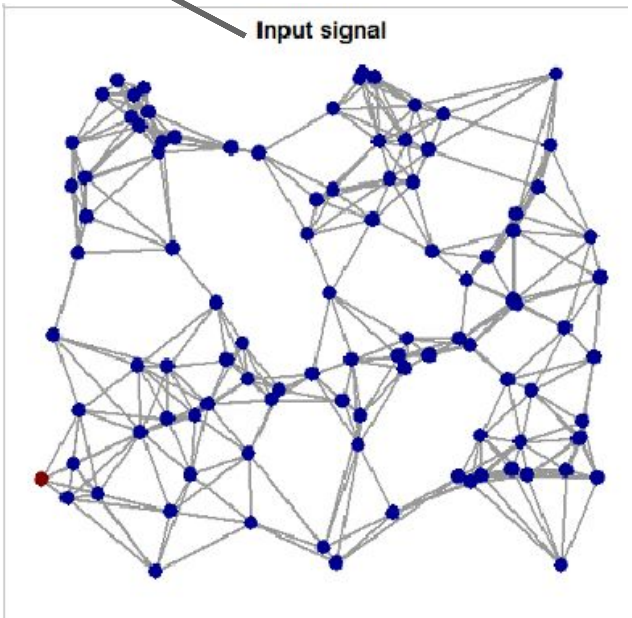
Any signal can be written by weighted sum of these base functions

Spectral Graph Filtering

$$x_{filtered} = U diag(F(\boldsymbol{\lambda}))U^{\top} x,$$







Spectral MPNN

Forward calculations of one layer non-parametric Spectral MPNN

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathrm{diag}(F_i^{(l,j)}) U^\top H_i^{(l)}\right), \qquad \text{for } j \in \{1, \ldots, f_{l+1}\}.$$

$F^{(l,j)} \in \mathbb{R}^{n \times f_l}$ is the corresponding weight vector to be tuned

Forward calculations of one layer parametric Spectral MPNN

$$H_j^{(l+1)} = \sigma\Big(\sum_{i=1}^{f_l} U \,\mathrm{diag}\left(B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top\right) U^\top H_i^{(l)}\Big),$$

Bridging the Gap Between Spectral and Spatial MPNN

- We proposed new framework to generalize both approaches.

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

Convolution support   Node Representation   Trainable Params

- Spatial Methods are defined by C matrices
- Spectral Method defined by $B_{i,j} = \Phi_j(\lambda_i)$.

Spectral to Spatial transition $C^{(s)} = U \operatorname{diag}(\Phi_s(\boldsymbol{\lambda}))U^\top.$

Bridging the Gap Between Spectral and Spatial MPNN

- Trainable or Fixed supports

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.*

In the trainable support case, supports can be different in each layer, which can be shown by $C^{(l,s)}$ for the s-th support in layer l. Formally, we can define a trainable support by:

$$\left(C^{(l,s)}\right)_{v,u} = h_{s,l}\left(H_{:v}^{(l)}, H_{:u}^{(l)}, E_{v,u}^{(l)}, A\right),$$

where $E_{v,u}^{(l)}$ shows edge features on layer l from node v to node u if it is available and $h(.)$ is any trainable model parametrized by (s, l).

Bridging the Gap Between Spectral and Spatial MPNN

- Spatial to Spectral transition

Corollary 1.1. *The frequency profile* $$\Phi_s(\boldsymbol{\lambda}) = diag^{-1}(U^\top C^{(s)} U).$$

- Definition of Spatial & Spectral MPNN

Definition 2. *A* ***Spectral-designed*** *graph convolution refers to a convolution where supports are written as a function of eigenvalues ($\Phi_s(\boldsymbol{\lambda})$) and eigenvectors ($U$) of the corresponding graph Laplacian (equation $\boxed{6}$). Thus, each convolution support $C^{(s)}$ has the same frequency response $\Phi_s(\boldsymbol{\lambda})$ over different graphs. Graph convolution out of this definition is called* ***spatial-designed*** *graph convolution.*

Some MPNN Models

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

- Vanilla Graph Convolution

$$C = A + I$$

- Molecular fingerprints[1], NIPS. 2015

$$C^{(1)} = I, \quad C^{(n)} = subset\ of\ A\ according\ to\ node\ type$$

- Patchy-San [2], ICML. 2016
  - Nauty: Find cardinal ordering of neighbors

$$C^{(1)} = I, C^{(n)} = (n-1)^{th} \text{ neighbours connections.}$$

- GraphSage [3], NIPS 2017

$$C^{(1)} = I \quad C^{(2)} = \tilde{A}$$ Row normalized adjacency

Some MPNN Models

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

- GCN[4], ICLR. 2017

$$C = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}$$ where $$\widetilde{A} = (A + I).\quad \widetilde{D}_{i,i} = \sum_j \widetilde{A}_{i,i}$$

- ChebNet[5] NIPS 2017

$$C^{(1)} = I \qquad C^{(2)} = 2L/\lambda_{\max} - I \quad C^{(k)} = 2C^{(2)}C^{(k-1)} - C^{(k-2)}.$$

- GAT[6], ICLR, 2018

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

Some MPNN Models

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

- Monet[7], CVPR2017

$$C^{(l,s)}_{v,u} = \widetilde{A} \circ e_{s,l}\left(E_{v,u}\right), \quad \text{where } e_{s,l}(.) \text{ is trainable function for } s\text{-th support's } l\text{-th layer.}$$

- SplineCNN[8], CVPR2018

$$C^{(l,s,i)}_{v,u} = \widetilde{A} \circ g_{s,l,i}\left(E_{v,u}\right), \quad \text{where } g_{s,l,i}(.) \text{ is trainable function for } i\text{-th input band, } s\text{-th support's } l\text{-th layer.}$$

- Gated GCN[9], 2018

$$C^{(1)} = I, \quad C^{(l,2)}_{v,u} = A \circ \eta\left(H^{(l)}_{:v}, H^{(l)}_{:u}, E^{(l)}_{v,u}\right)$$

Some MPNN Models

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

- CayleyNet[10], IEEE Transaction on Signal Proc, 2019

$$C^{(1)} = I$$
$$C^{(2r)} = Re(\rho(hL)^r)$$
$$C^{(2r+1)} = Re(\mathbf{i}\rho(hL)^r)$$

- GIN[11], ICLR 2019

$$C = A + (1+\epsilon)I \qquad \epsilon \text{ is trainable parameter}$$

- Custom Designed Spectral GNN[12], ICML 2020 Workshop

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

Where $\Phi_s(\boldsymbol{\lambda})$ designed by custom manner (problem specific).

Spectral Analysis of some MPNNs

Theorem 2. *The theoretical frequency response of each support of ChebNet can be defined as*

$$\Phi_1(\boldsymbol{\lambda}) = \mathbf{1}, \;\; \Phi_2(\boldsymbol{\lambda}) = \frac{2\boldsymbol{\lambda}}{\lambda_{\max}} - \mathbf{1}, \;\; \Phi_k(\boldsymbol{\lambda}) = 2\Phi_2(\boldsymbol{\lambda})\Phi_{k-1}(\boldsymbol{\lambda}) - \Phi_{k-2}(\boldsymbol{\lambda}),$$

where $\mathbf{1}$ *is the vector of ones and* $\lambda_{\max}$ *is the maximum eigenvalue.*

Theorem 3. *The theoretical frequency response of each support of CayleyNet can be defined as*

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

where h *is a trainable scalar and* $\theta(x) = atan2(-1, x) - atan2(1, x)$.

Spectral Analysis of some MPNNs

Theorem 4. *The theoretical frequency response of GCN support can be approximated as*

$$\Phi(\boldsymbol{\lambda}) \approx \mathbf{1} - \boldsymbol{\lambda}\overline{p}/(\overline{p}+1),$$

where $\overline{p}$ is the average node degree in the graph.

Theorem 5. *The theoretical frequency response of GIN support can be approximted as*

$$\Phi(\boldsymbol{\lambda}) \approx \overline{p}\left(\frac{1+\epsilon}{\overline{p}} + \mathbf{1} - \boldsymbol{\lambda}\right)$$

where ϵ is a trainable scalar.

Table 1: Summary of the studied GNN models.

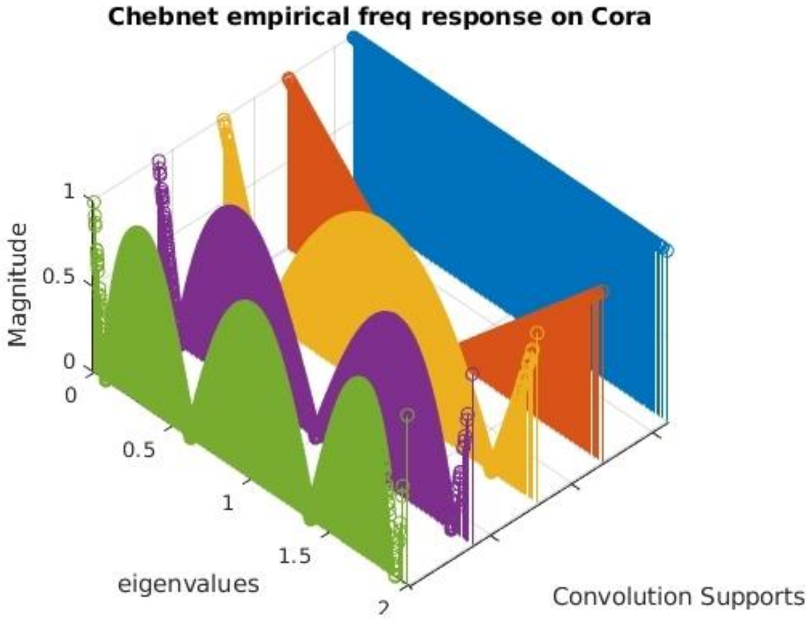
	Design	Support Type	Convolution Matrix	Frequency Response
MLP	Spectral	Fixed	$C = I$	$\Phi(\boldsymbol{\lambda}) = \mathbf{1}$
GCN	Spatial	Fixed	$C = \tilde{D}^{-0.5}\tilde{A}\tilde{D}^{-0.5}$	$\Phi(\boldsymbol{\lambda}) \approx \mathbf{1} - \boldsymbol{\lambda}\overline{p}/(\overline{p}+1)$
GIN	Spatial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

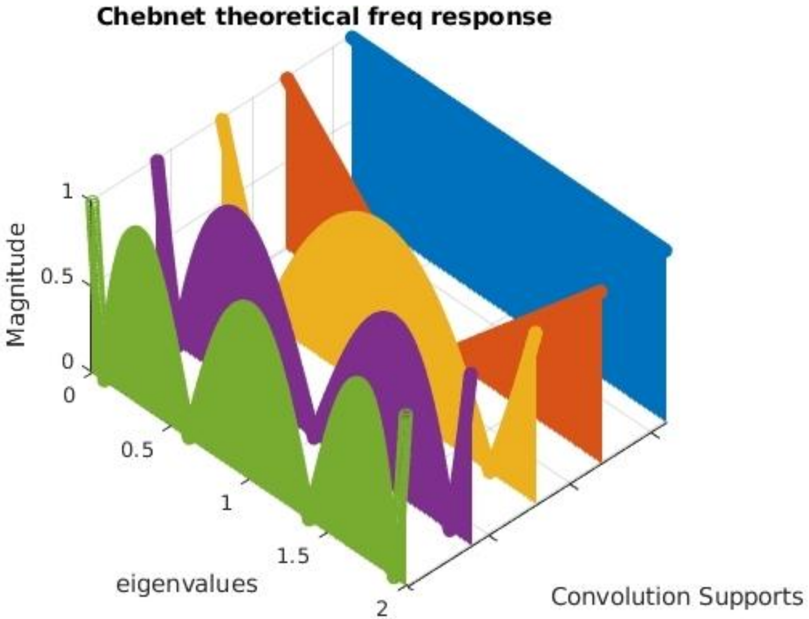
[a] $\rho(x) = (x - \mathbf{i}I)/(x + \mathbf{i}I)$

Chebnet[5], Spectral Designed, Fixed Support

$$C^{(1)} = I$$
$$C^{(2)} = 2L/\lambda_{\max} - I$$
$$C^{(s)} = 2C^{(2)}C^{(s-1)} - C^{(s-2)}$$

$$\Phi_1(\boldsymbol{\lambda}) = \mathbf{1}$$
$$\Phi_2(\boldsymbol{\lambda}) = 2\boldsymbol{\lambda}/\lambda_{\max} - \mathbf{1}$$
$$\Phi_s(\boldsymbol{\lambda}) = 2\Phi_2(\boldsymbol{\lambda})\Phi_{s-1}(\boldsymbol{\lambda}) - \Phi_{s-2}(\boldsymbol{\lambda})$$

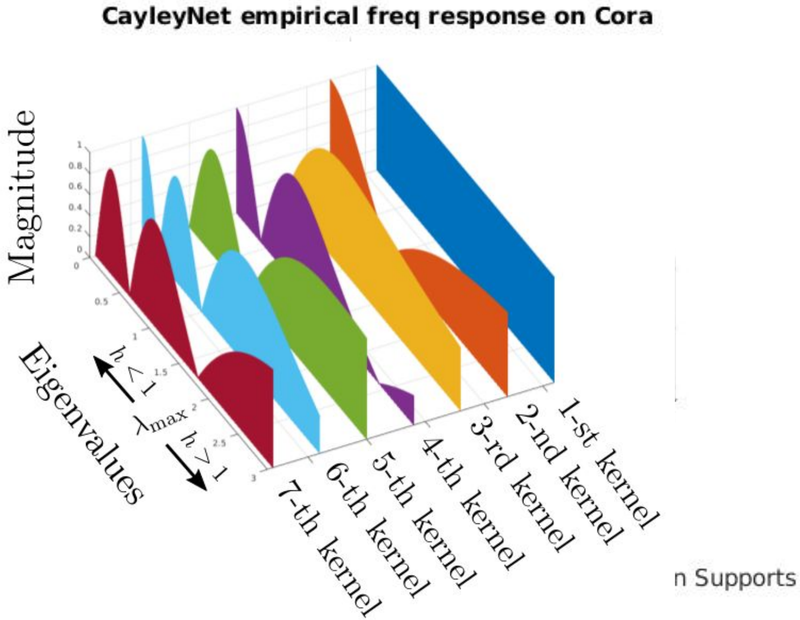


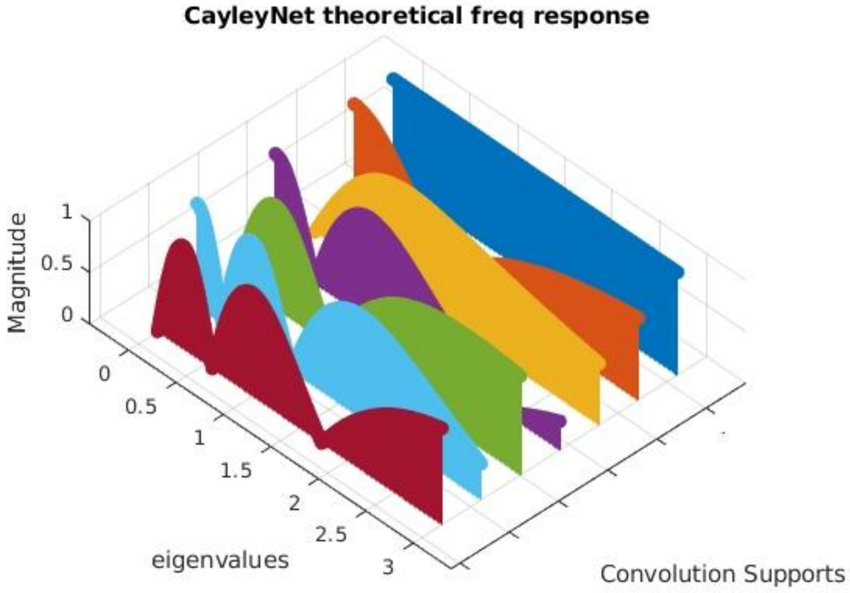


CayleyNet[10] Spectral Designed, Trainable Support

$$C^{(1)} = I$$
$$C^{(2r)} = Re(\rho(hL)^r)$$
$$C^{(2r+1)} = Re(\mathbf{i}\rho(hL)^r)$$

$$\Phi_1(\boldsymbol{\lambda}) = \mathbf{1}$$
$$\Phi_{2r}(\boldsymbol{\lambda}) = \cos(r\theta(h\boldsymbol{\lambda}))$$
$$\Phi_{2r+1}(\boldsymbol{\lambda}) = -\sin(r\theta(h\boldsymbol{\lambda}))$$

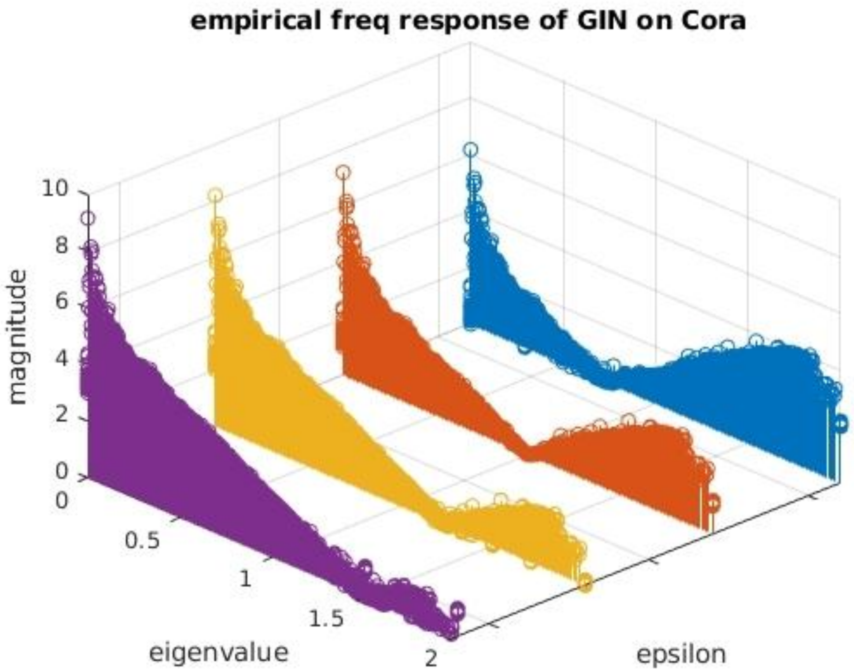


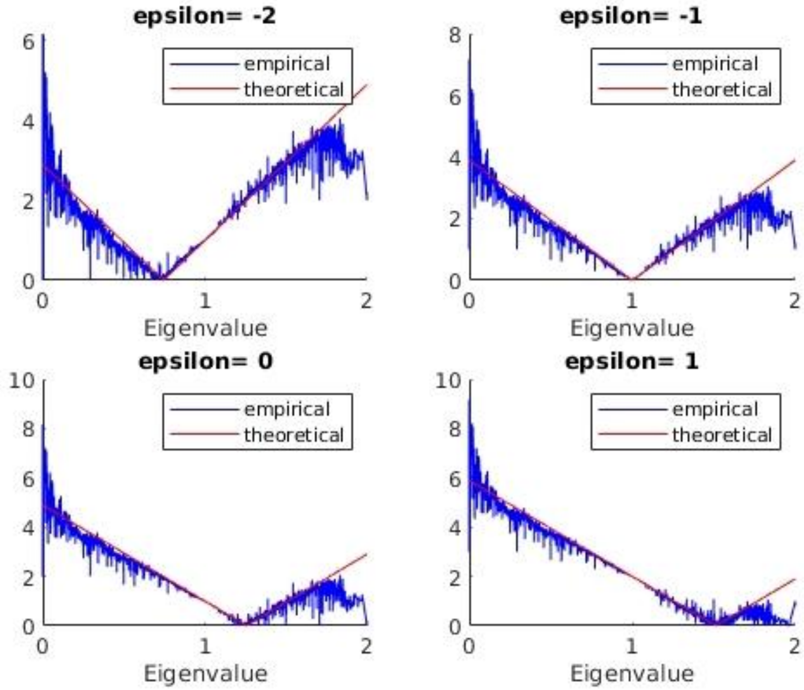


GIN[11] Spatial Designed, Trainable Support

$$C = A + (1+\epsilon)I$$

$$\Phi(\boldsymbol{\lambda}) \approx \bar{p}\left(\frac{1+\epsilon}{\bar{p}} + \mathbf{1} - \boldsymbol{\lambda}\right)$$

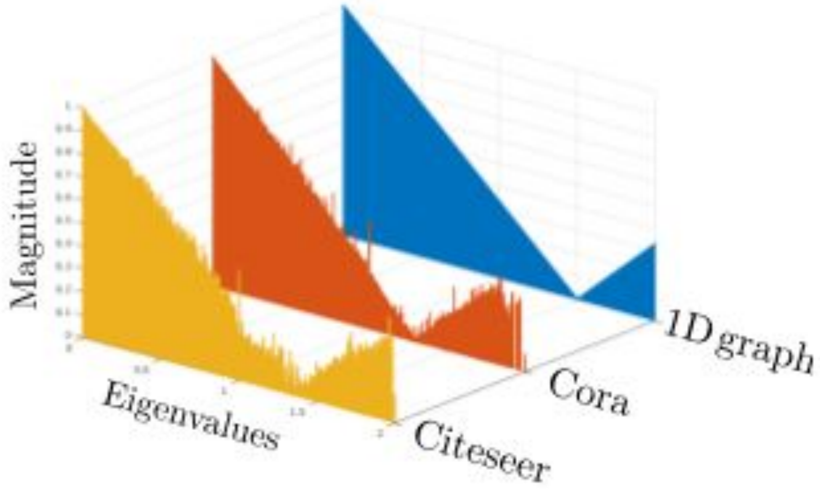




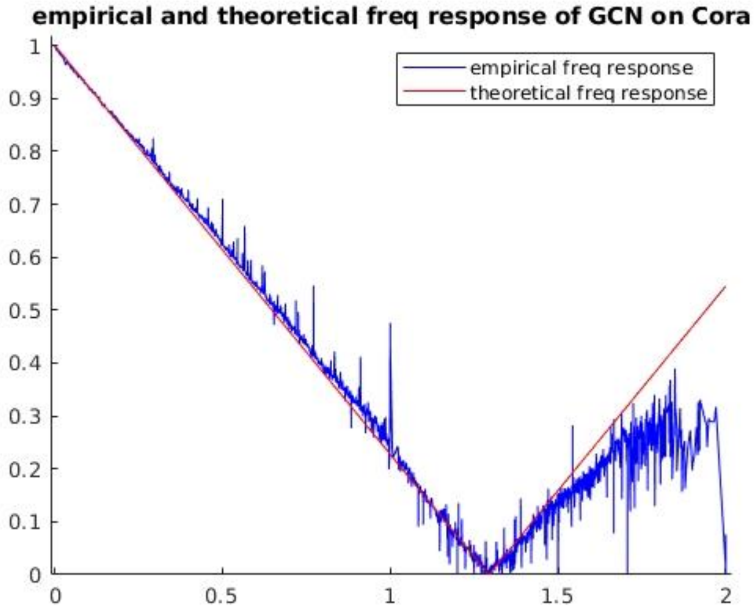
GCN[4] Spatial Designed, Fixed Support

$$C = \tilde{D}^{-0.5}\tilde{A}\tilde{D}^{-0.5}$$

$$\Phi(\boldsymbol{\lambda}) \approx \mathbf{1} - \boldsymbol{\lambda}\overline{p}/(\overline{p}+1)$$

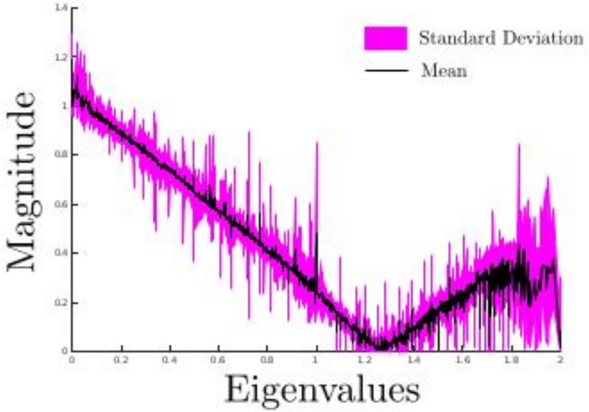


(a) GCN frequency profiles

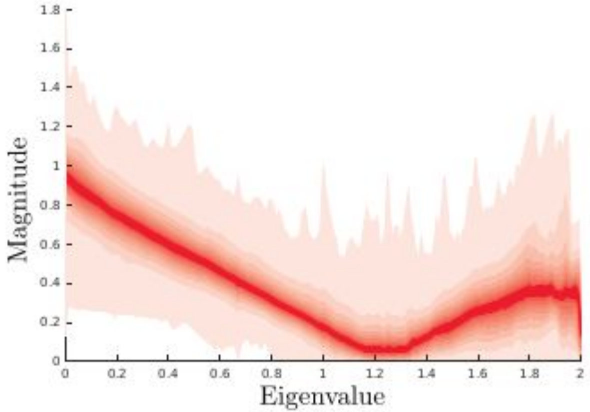


GAT[6] Spatial Designed, Trainable Support

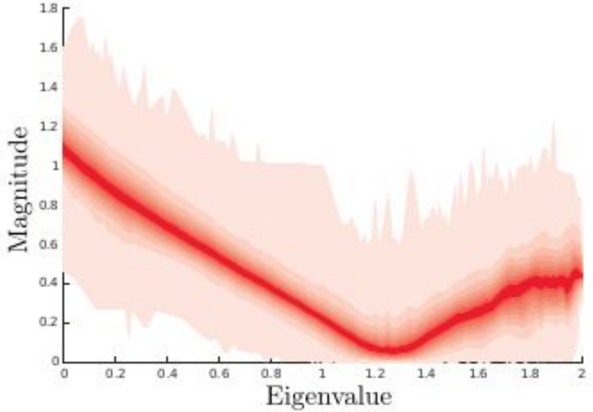
$$C^{(s)}_{v,u} = e_{v,u} / \sum_{k \in \tilde{\mathcal{N}}(v)} e_{v,k}$$



(a) Expected frequency response from Simulation on Cora



(b) Heat density map of learned frequency response on ENZYMES



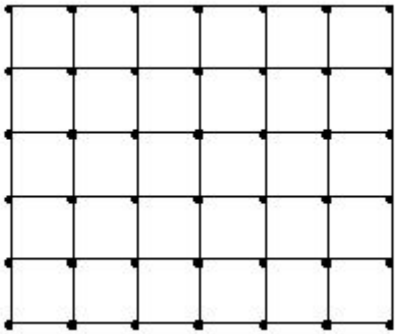
(c) Heat density map of learned frequency response on PROTEINS

Our Conclusion on Frequency Responses

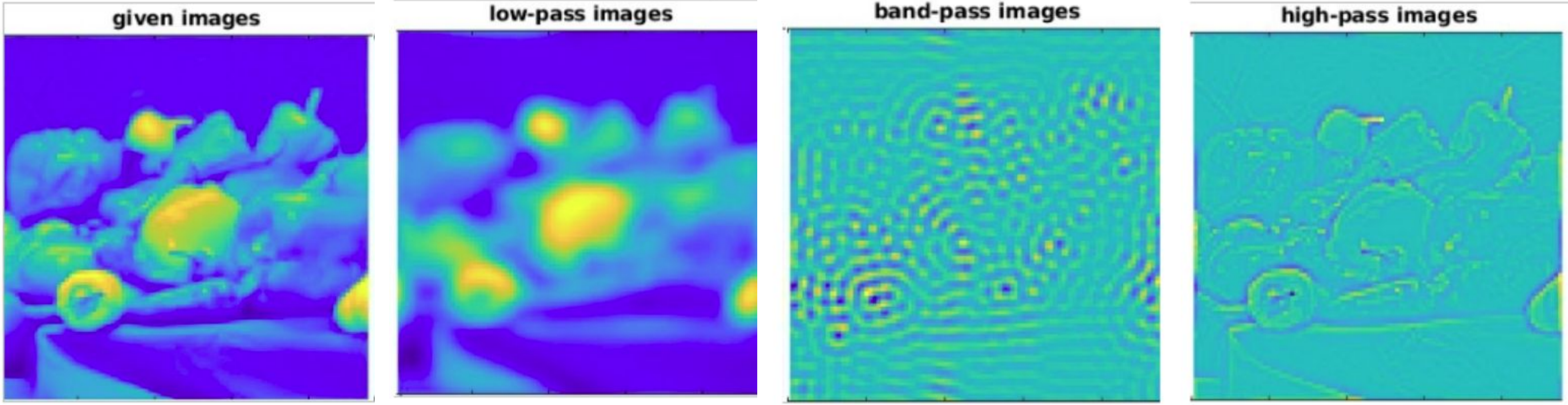
- Spatial MPNN is nothing but just low-pass filter!
- Spectral MPNN cover the spectrum well but not have band specific filters
- Most of the natural graph problems need low-pass effects.
- If the signal on graph matters, spectral methods are best!

Experiments

- 2DGrid graph

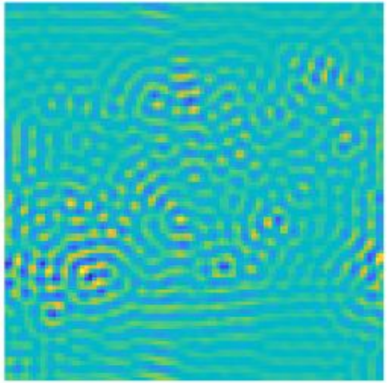


$\Phi_1(\rho) = \exp(-100\rho^2)$ $\Phi_2(\rho) = \exp(-1000(\rho - 0.5)^2)$ $\Phi_3(\rho) = 1 - \exp(-10\rho^2)$



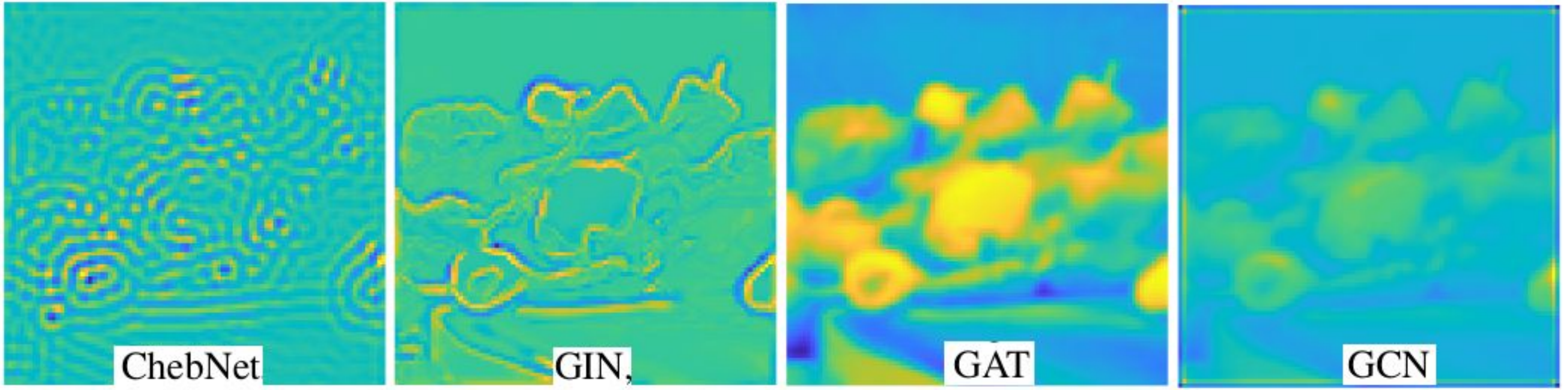
Experiments

Ground truth



Prediction Target	GCN	GIN	GAT	ChebNet
Low-pass filter (Φ_1)	15.55	11.01	10.50	3.44
Band-pass filter (Φ_2)	79.72	63.24	79.68	17.30
High-pass filter (Φ_3)	29.51	14.27	29.10	2.04

Prediction



Experiments

- BandClass

Table 5: Test set accuracy and binary cross entropy loss.

	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

Experiments

- MNIST-75

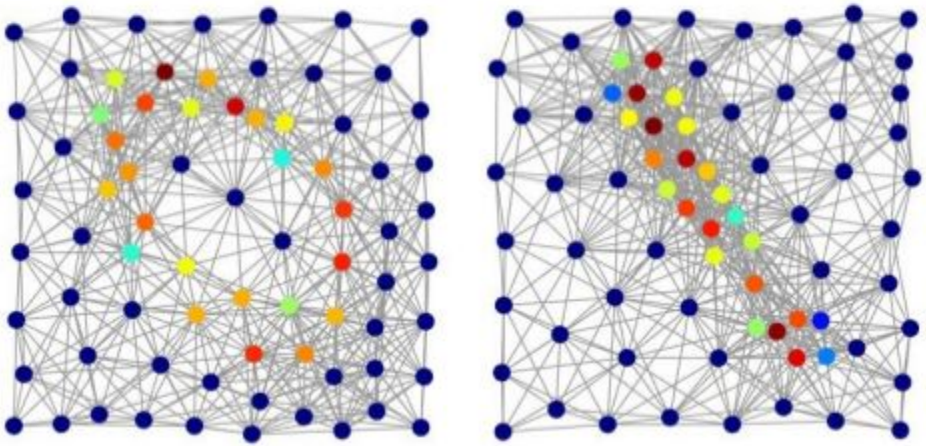


Table 2: Test set accuracies on MNIST superpixel dataset

Node feature	MLP	GCN	GIN	GAT	CayleyNet	ChebNet
Node degree	11.29±0.5	15.81±0.8	32.45±1.2	31.72±1.5	45.61±1.7	46.23±1.8
Pixel value	12.11±0.5	11.35±1.1	64.96±3.9	62.61±2.9	88.41±2.1	91.10±1.9
Both	25.10±1.2	52.98±3.1	75.23±4.1	82.73±2.1	90.31±2.3	92.08±2.2

Summary of Our Contributions

- Bridging the gap between Spatial-Spectral MPNN
- Show how to do spectral analysis of GNN.
- Show spatial MPNN is nothing but low-pass filter.
- Propose new taxonomy on GNN.
- Put a new criteria on theoretical evaluation of expressive power of GNN.

https://github.com/balcilar/gnn-spectral-expressive-power

References

1. D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell,T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” in NIPS. 2015.
2. M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” in ICML2016.
3. Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In Advances in neural information processing systems (pp. 1024-1034).
4. T. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” ICLR2017.
5. M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in NIPS2017.
6. P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, ́and Y. Bengio, “Graph attention networks,” ICLR2018,
7. Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5115–5124, 2017
8. Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
9. Xavier Bresson and Thomas Laurent. Residual gated graph convnets, 2018. URL https://openreview.net/forum?id=HyXBcYg0b
10. R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, “Cayleynets: Graph convolutional neural networks with complex rational spectral filters,” IEEE Transactions on Signal Processing, 2019.
11. Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In International Conference on Learning Representations, 2019
12. Balcilar, M., Renton, G., Héroux, P., Gauzere, B., Adam, S., & Honeine, P. (2020). Bridging the Gap Between Spectral and Spatial Domains in Graph Neural Networks. arXiv preprint arXiv:2003.11702.
13. Analyzing the Expressive Power of Graph Neural Networks in a Spectral Perspective. Balcilar, M.; Renton, G.; Héroux, P.; Gaüzère, B.; Adam, S.; and Honeine, P. In *International Conference on Learning Representations*, Vienna, Austria, 4~May 2021