Analyzing The Expressive Power of GNNs

A Spectral Perspective

Shichang Zhang

01-12-2021

Outline

- Introduction and motivation
- Convolution, graph signal, and graph Fourier transformation
- Spectral and spatial GNNs
- Frequency profile analysis

Brief Introduction and Motivation

- Understand the expressiveness of GNNs
  - Spectral perspective
- Reformulate and analyze GNNs under one framework
  - ChebNet, CayleyNet, GCN, GAT, and GIN

Notations

$A \in \{0,1\}^{n\times n}$	: adjacency matrix	$\tilde{A} = A + I$	: adjacency matrix with self-loop
$D \in \mathbb{R}^{n\times n}$	: diagonal degree matrix	$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$	: degree matrix with self-loop
$L = I - D^{-1/2}AD^{-1/2}$	: normalized Laplacian	$L = U\mathbf{diag}(\boldsymbol{\lambda})U^T$	: eigendecomposition $U \in \mathbb{R}^{n\times n}$ $\boldsymbol{\lambda} \in \mathbb{R}^n$
$\mathbf{diag}(.)$	: vector -> diagonal matrix	$diag^{-1}(.)$	: diagonal matrix -> vector
$X \in \mathbb{R}^{n\times f_0}$	: node features	$X_i \in \mathbb{R}^n$	: the i-th column (feature)
$H^{(l)} \in \mathbb{R}^{n\times f_l}$	: node representations in layer l	$H^{(0)} = X$	

Notations

$A \in \{0,1\}^{n\times n}$	: adjacency matrix	$\tilde{A} = A + I$	: adjacency matrix with self-loop
$D \in \mathbb{R}^{n\times n}$	: diagonal degree matrix	$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$	: degree matrix with self-loop
$L = I - D^{-1/2}AD^{-1/2}$	: normalized Laplacian	$L = U\mathbf{diag}(\boldsymbol{\lambda})U^T$	: eigendecomposition $U \in \mathbb{R}^{n\times n}$ $\boldsymbol{\lambda} \in \mathbb{R}^n$
$\mathbf{diag}(.)$	: vector -> diagonal matrix	$diag^{-1}(.)$	: diagonal matrix -> vector
$X \in \mathbb{R}^{n\times f_0}$	: node features	$X_i \in \mathbb{R}^n$	: the i-th column (feature)
$H^{(l)} \in \mathbb{R}^{n\times f_l}$	: node representations in layer l	$H^{(0)} = X$	

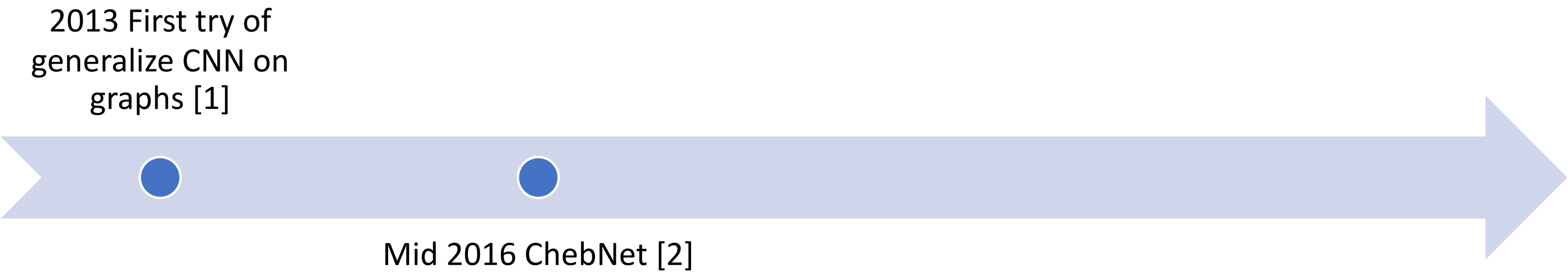
GCN layer: $H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$

$W^{(l)}$: parameters in layer l

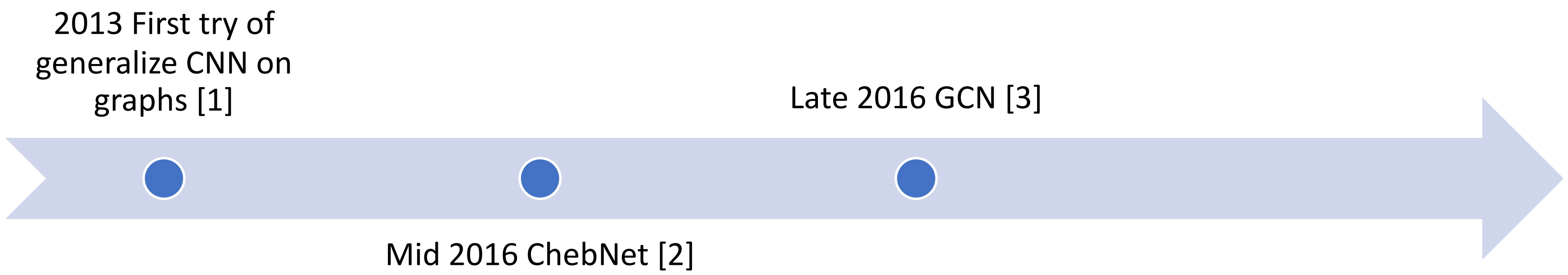
Timeline of GNNs

2013 First try of generalize CNN on graphs [1]

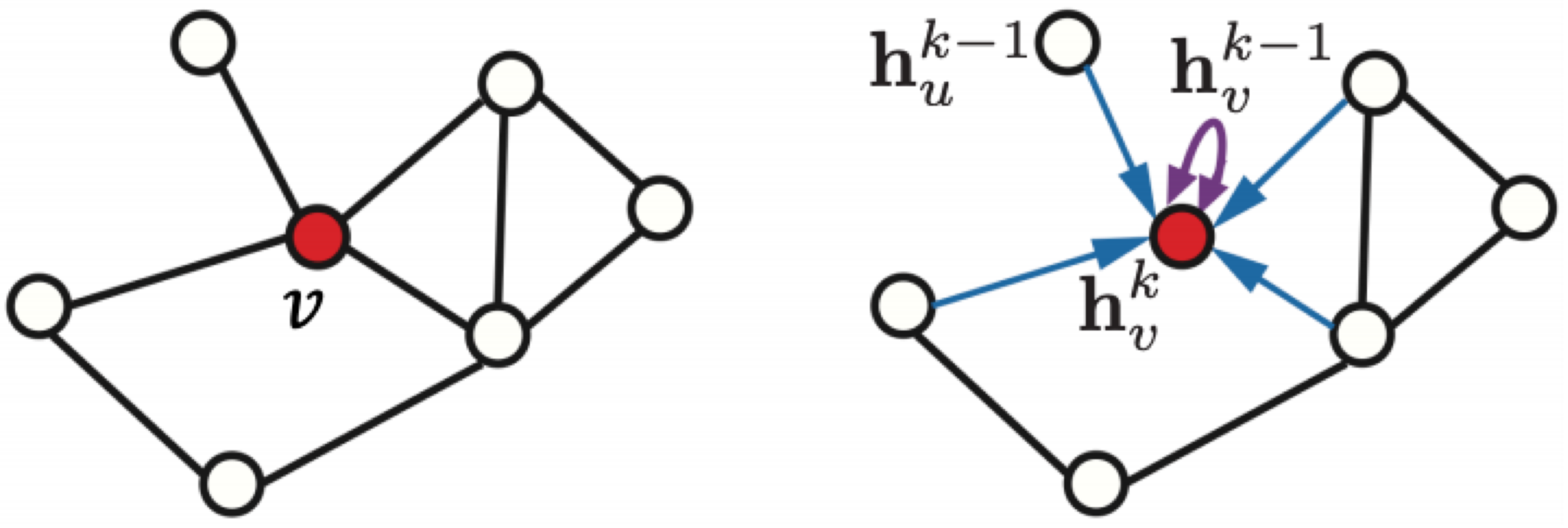
Timeline of GNNs



Timeline of GNNs



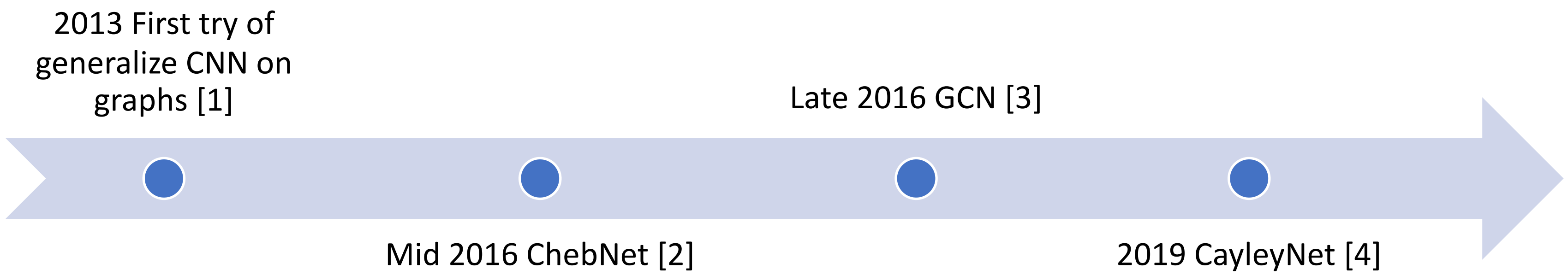
Graph Convolutional Network



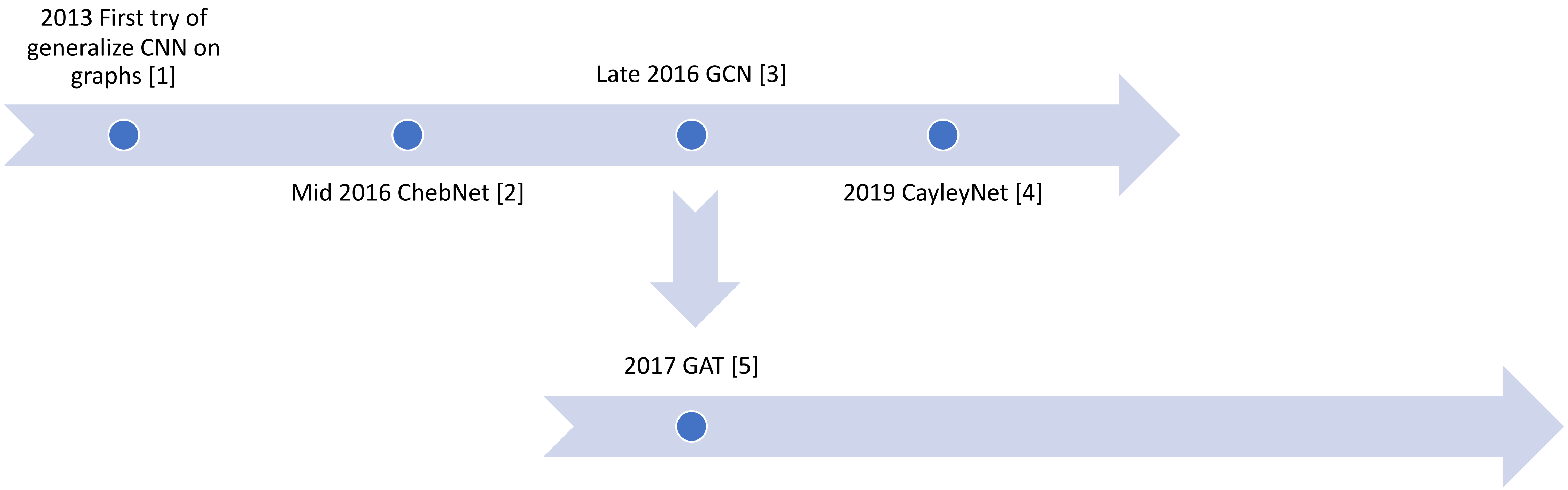
GCN layer: $H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$

$W^{(l)}$: parameters in layer l

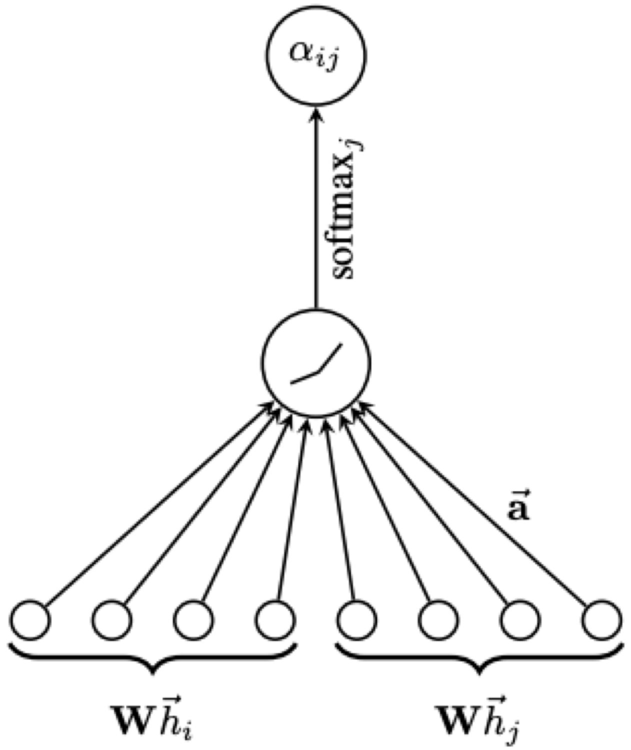
Timeline of GNNs

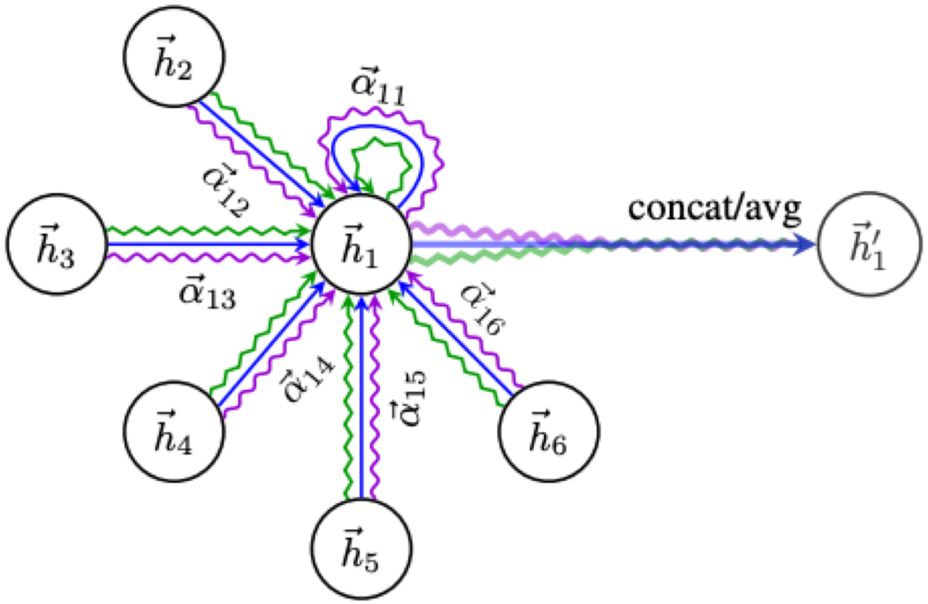


Timeline of GNNs



Graph Attention Network



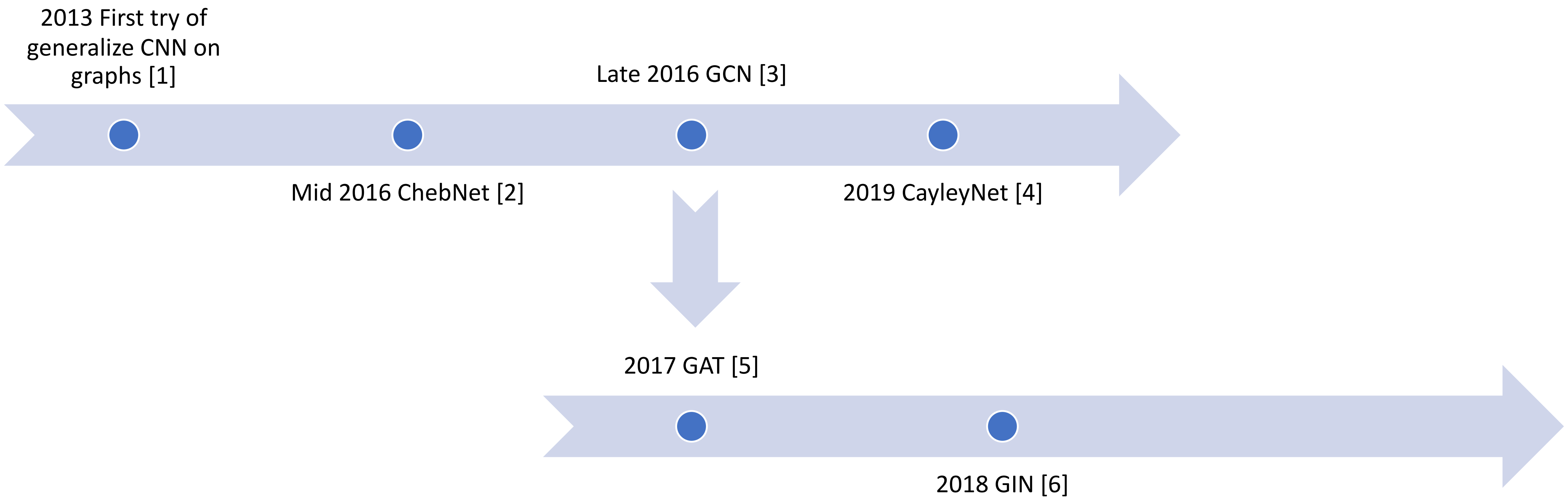


GCN layer: $$H_{:i}^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \alpha_{i,j} W^{(l)} H_{:j}^{(l)}\right)$$

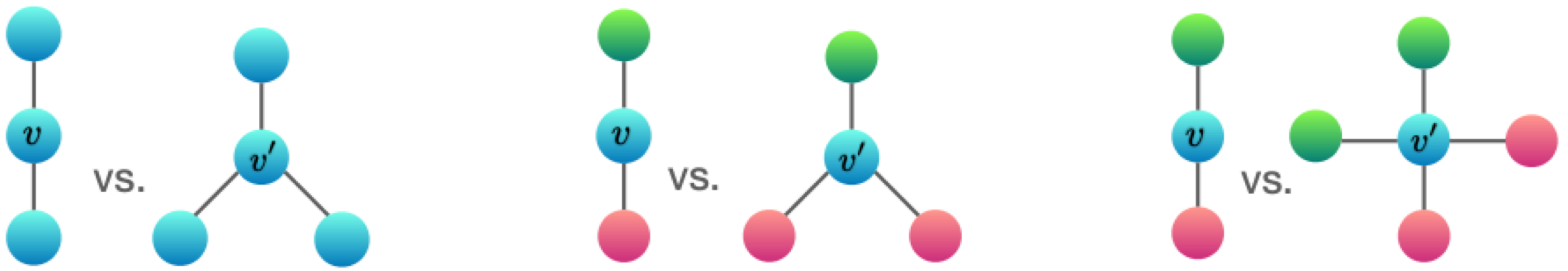
$W^{(l)}$: parameters in layer l

$\alpha_{i,j}$: attention score

Timeline of GNNs



Graph Isomorphism Network

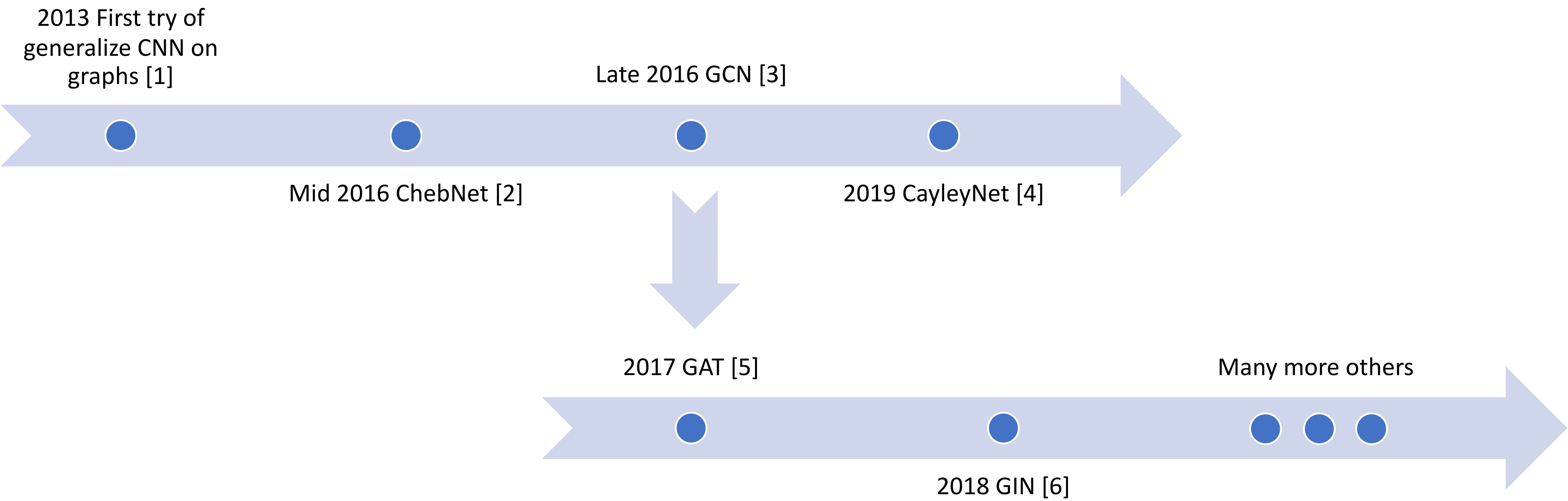


(a) Mean and Max both fail (b) Max fails (c) Mean and Max both fail

GIN layer: $H^{(l+1)} = \sigma((A + (1+\epsilon)I)H^{(l)}W^{(l)})$

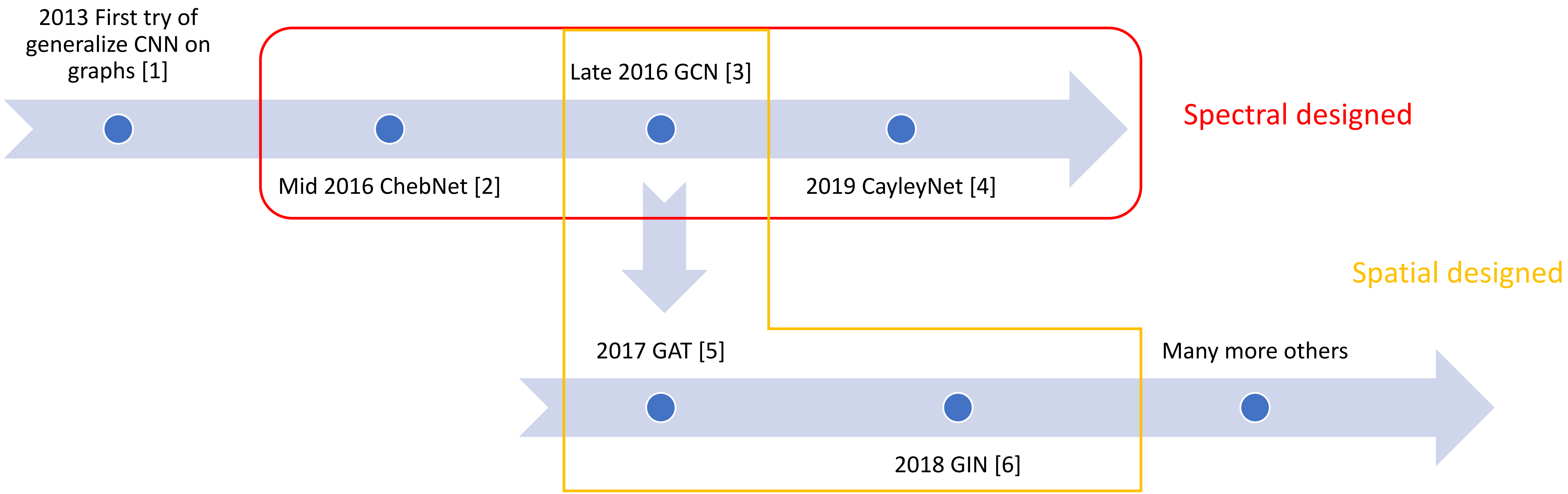
$W^{(l)}$: parameters in layer l

Timeline of GNNs



[1] Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203.*
[2] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, *29*, 3844-3852.
[3] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907.*
[4] Levie, R., Monti, F., Bresson, X., & Bronstein, M. M. (2018). Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Transactions on Signal Processing, 67(1), 97-109.
[5] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903
[6] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks?. arXiv preprint arXiv:1810.00826.

Timeline of GNNs

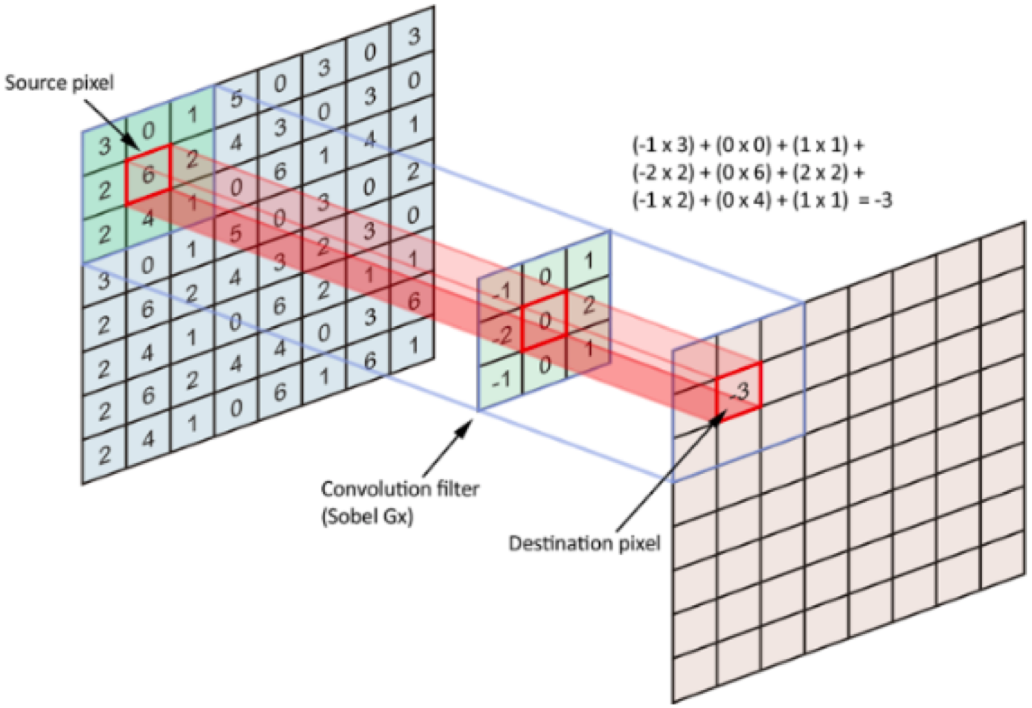


[1] Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203.*
[2] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, *29*, 3844-3852.
[3] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907.*
[4] Levie, R., Monti, F., Bresson, X., & Bronstein, M. M. (2018). Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Transactions on Signal Processing, 67(1), 97-109.
[5] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903
[6] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks?. arXiv preprint arXiv:1810.00826.

The Idea of Convolution

- Convolution of 2-d matrices in computer vision

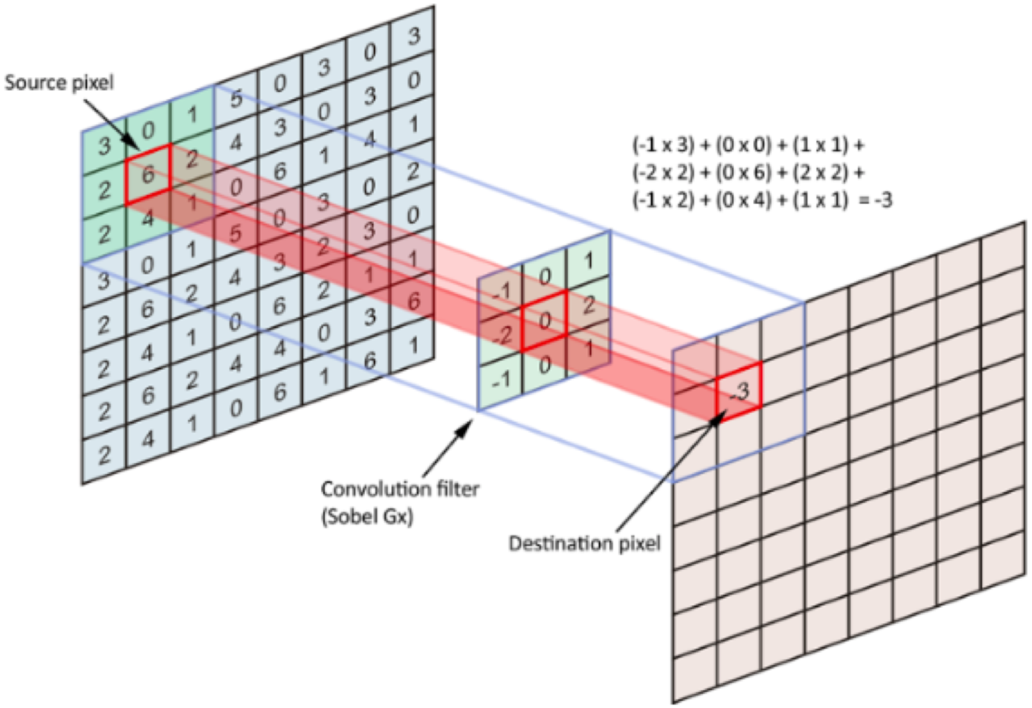
$$(f * g)(a, b) = \sum_h \sum_w f(h, w) g(a - h, b - w)$$



The Idea of Convolution

- Convolution of 2-d matrices in computer vision

$$(f * g)(a,b) = \sum_h \sum_w f(h,w)g(a-h, b-w)$$



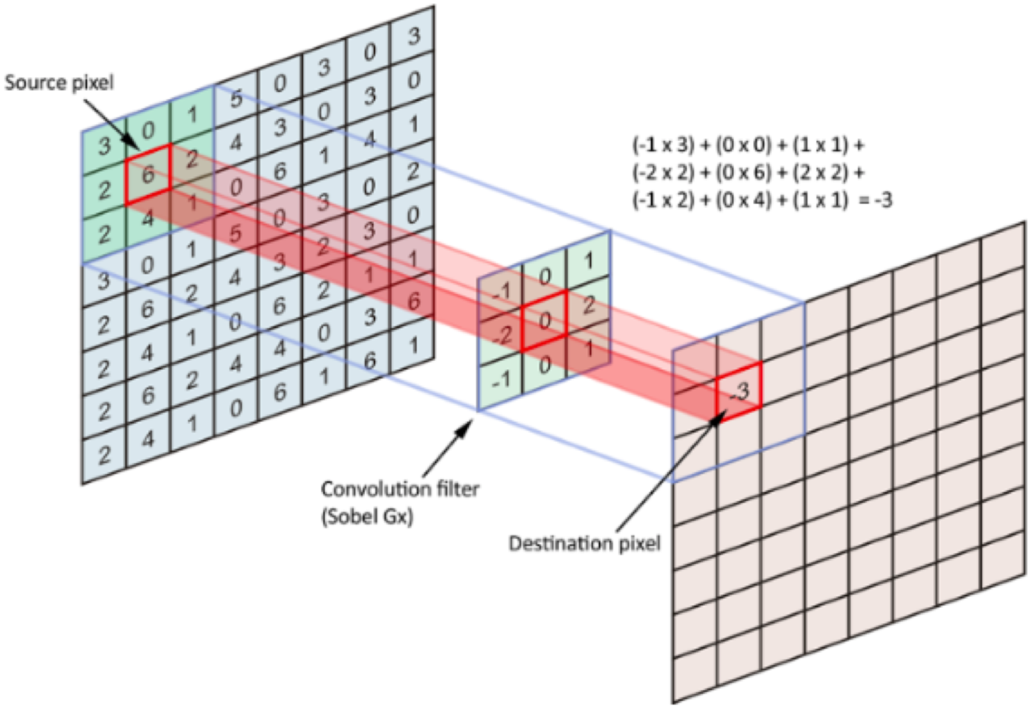
$$f : \mathbb{Z}^2 \to \mathbb{R}$$

$$g : \mathbb{Z}^2 \to \mathbb{R}$$

The Idea of Convolution

- Convolution of 2-d matrices in computer vision

$$(f * g)(a, b) = \sum_h \sum_w f(h, w) g(a - h, b - w)$$



$$f : \mathbb{Z}^2 \to \mathbb{R}$$

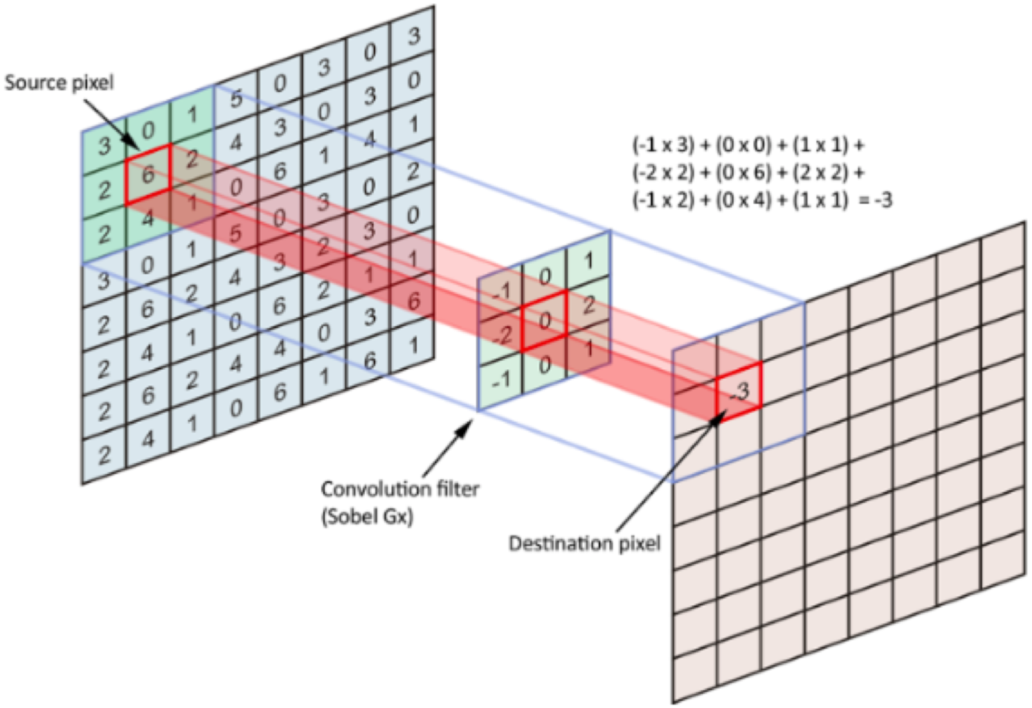
$$g : \mathbb{Z}^2 \to \mathbb{R}$$

What is the corresponding definition of convolution on graphs?

The Idea of Convolution

- Convolution of 2-d matrices in computer vision

$$(f * g)(a,b) = \sum_h \sum_w f(h,w)g(a-h,b-w)$$



$$f : \mathbb{Z}^2 \rightarrow \mathbb{R}$$

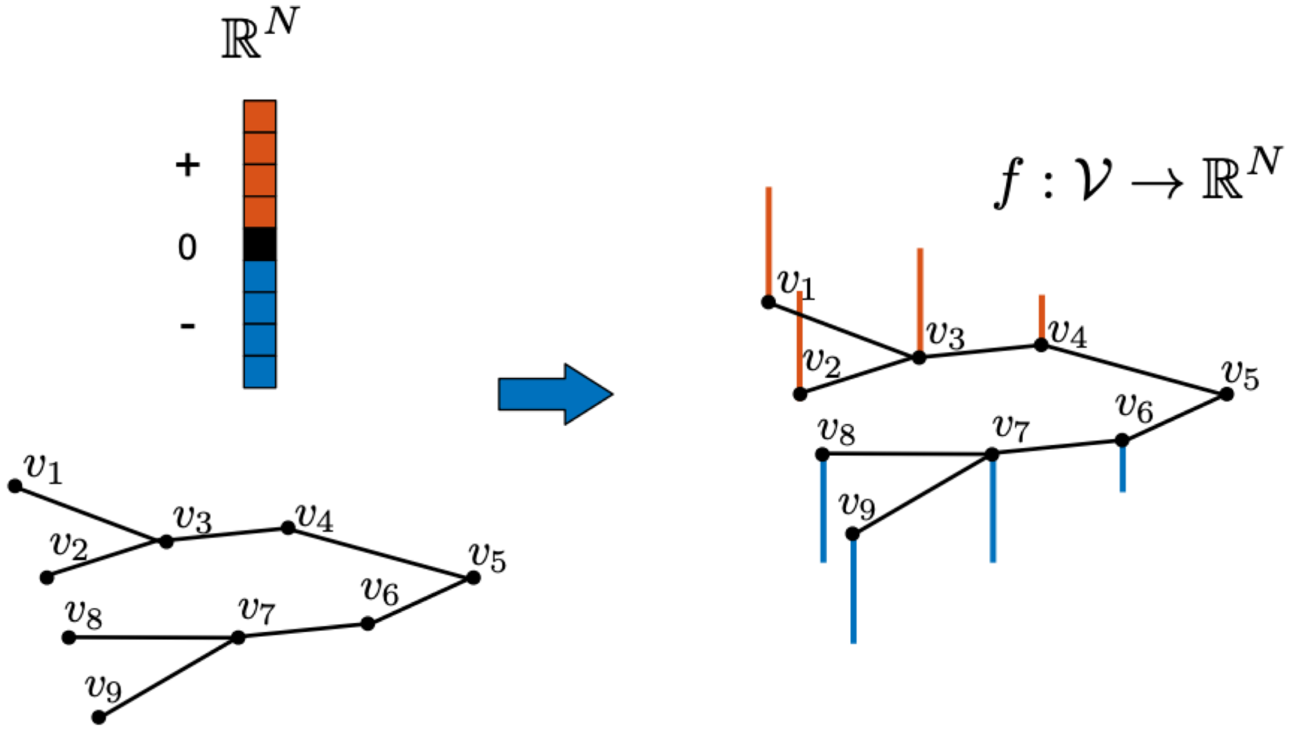
$$g : \mathbb{Z}^2 \rightarrow \mathbb{R}$$

What is the corresponding definition of convolution on graphs?

What is the corresponding of functions definition on graphs?

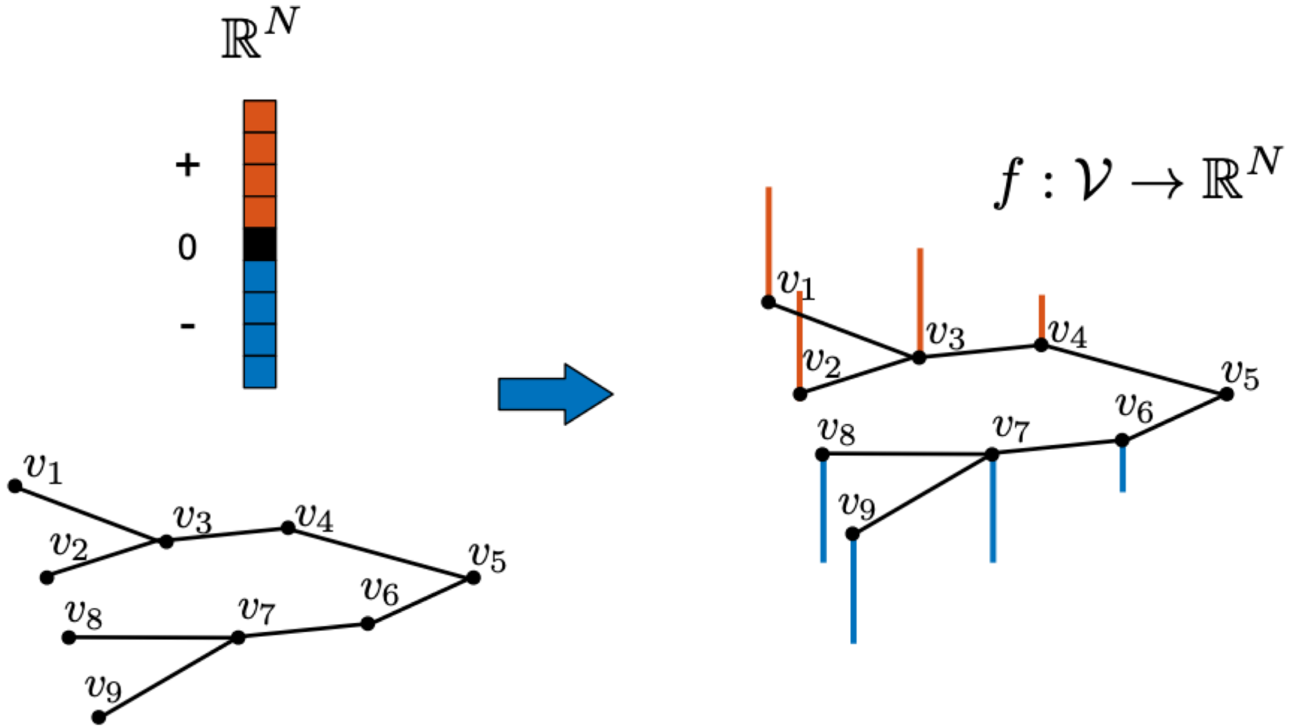
Graph Signals

- A function defined on the vertices of a graph



Graph Signals

- A function defined on the vertices of a graph



Node features:

$$X \in \mathbb{R}^{n \times f_0}$$

Each column (feature) is a signal:

$$X_i \in \mathbb{R}^n$$

Graph Laplacian

- A function that measures smoothness of a graph signal

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{N} W_{ij} \left(f(i) - f(j)\right)^2$$

Weighted adjacency matrix:

$$W \in \mathbb{R}^{n \times n}$$

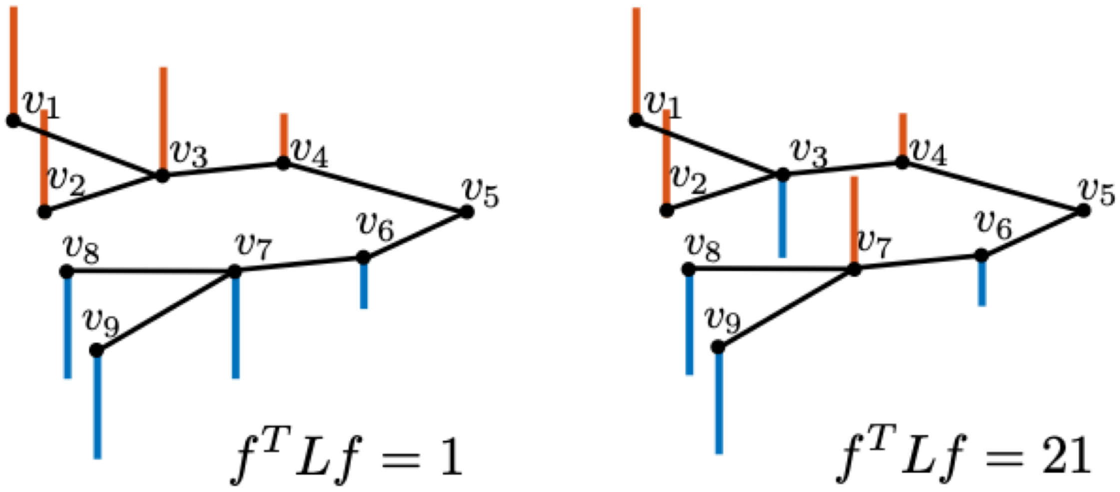
Graph Laplacian

- A function that measures smoothness of a graph signal

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{N} W_{ij} \left(f(i) - f(j)\right)^2$$

Weighted adjacency matrix:

$$W \in \mathbb{R}^{n \times n}$$



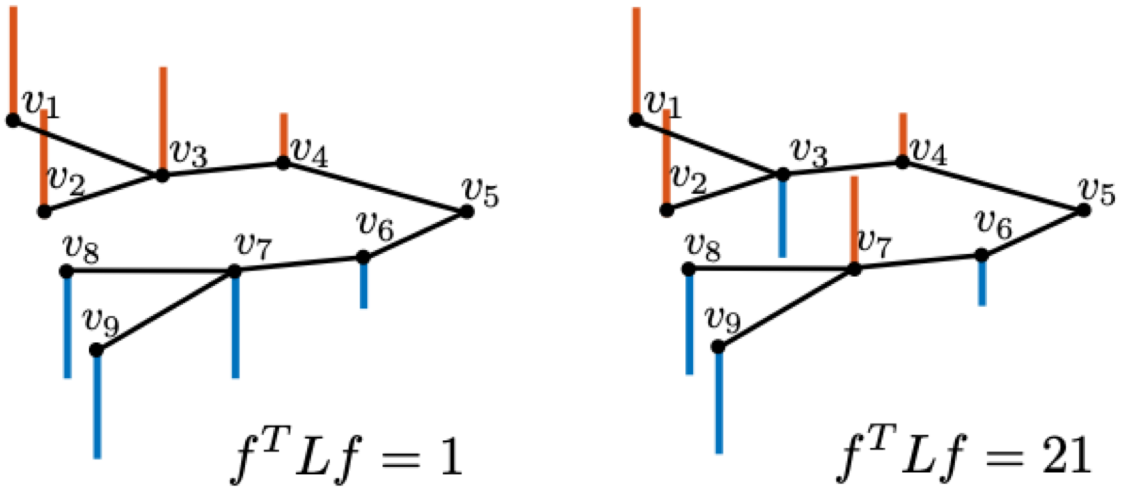
Graph Laplacian

- A function that measures smoothness of a graph signal

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{N} W_{ij} \left(f(i) - f(j)\right)^2$$

Weighted adjacency matrix:

$$W \in \mathbb{R}^{n \times n}$$



How to define convolution on graph signals?

Fourier Transformation

- Classical Fourier transformation
  - From time/space domain to frequency domain
  - Formula:

$$\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{2\pi i \xi t} dt$$

Fourier Transformation

- Classical Fourier transformation
  - From time/space domain to frequency domain
  - Formula:

$$\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{2\pi i \xi t} dt$$

- Graph Fourier transformation
  - From vertex domain to graph spectral domain

Fourier Transformation

- Classical Fourier transformation
  - From time/space domain to frequency domain
  - Formula:

$$\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{2\pi i \xi t} dt$$

- Graph Fourier transformation
  - From vertex domain to graph spectral domain
  - Formula:

$$\hat{f}(\lambda_l) = \langle f, u_l \rangle = \sum_{i=1}^{n} f(i) u_l^*(i)$$

λ_l : l-th eigenvalue of L

u_l : l-th eigenvector of L

Fourier Transformation

- Classical Fourier transformation
  - From time/space domain to frequency domain
  - Formula:

$$\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{2\pi i \xi t} dt$$

- Graph Fourier transformation
  - From vertex domain to graph spectral domain
  - Formula:

$$\hat{f}(\lambda_l) = \langle f, u_l \rangle = \sum_{i=1}^{n} f(i) u_l^*(i)$$

  - Inverse formula:

$$f(i) = \sum_{l=1}^{n} \hat{f}(\lambda_l) u_l(i)$$

λ_l : l-th eigenvalue of L

u_l : l-th eigenvector of L

Fourier Transformation

- Classical Fourier transformation
  - From time/space domain to frequency domain
  - Formula:

$$\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{2\pi i \xi t} dt$$

- Graph Fourier transformation
  - From vertex domain to graph spectral domain
  - Formula:

$$\hat{f}(\lambda_l) = \langle f, u_l \rangle = \sum_{i=1}^{n} f(i) u_l^*(i) \quad \longrightarrow \quad \hat{f} = U^T f$$

  - Inverse formula:

$$f(i) = \sum_{l=1}^{n} \hat{f}(\lambda_l) u_l(i) \quad \longrightarrow \quad f = U \hat{f}$$

λ_l : l-th eigenvalue of L
u_l : l-th eigenvector of L

Convolution and Fourier Transformation

- The Fourier transform of the convolution is given by the product of the Fourier transforms

Convolution and Fourier Transformation

- The Fourier transform of the convolution is given by the product of the Fourier transforms
- Generalize to graph Fourier transformation
  - Elementwise notation

$$(f * h)(i) = \sum_{l=1}^{n} \hat{f}(\lambda_l)\hat{h}(\lambda_l)u_l(i)$$

Convolution and Fourier Transformation

- The Fourier transform of the convolution is given by the product of the Fourier transforms
- Generalize to graph Fourier transformation
  - Elementwise notation

$$(f * h)(i) = \sum_{l=1}^{n} \hat{f}(\lambda_l)\hat{h}(\lambda_l)u_l(i)$$

  - Matrix notation

$$f * h = U(\hat{f} \odot \hat{h}) = U((U^T f) \odot (U^T h))$$

Convolution and Fourier Transformation

- The Fourier transform of the convolution is given by the product of the Fourier transforms
- Generalize to graph Fourier transformation
  - Elementwise notation

$$(f * h)(i) = \sum_{l=1}^{n} \hat{f}(\lambda_l)\hat{h}(\lambda_l)u_l(i)$$

  - Matrix notation

$$f * h = U(\hat{f} \odot \hat{h}) = U((U^T f) \odot (U^T h))$$

  - Further equals to

$$f * h = U diag(\hat{h}(\boldsymbol{\lambda}))U^T f$$

Spectral GNNs

- General formula of convolution in layer l (Bruna 2013)

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

Spectral GNNs

- General formula of convolution in layer l (Bruna 2013)

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

- Non-parametric spectral GNN
  - A full parameter matrix: $F^{(l,j)} \in \mathbb{R}^{n \times f_l}$

Spectral GNNs

- General formula of convolution in layer l (Bruna 2013)

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

- Non-parametric spectral GNN
  - A full parameter matrix: $F^{(l,j)} \in \mathbb{R}^{n \times f_l}$

- Function bases, e.g. B-splines, Chebyshev polynomials, Cayley polynomials
  - Formula: $F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top$

Spectral GNNs

- General formula of convolution in layer l (Bruna 2013)

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

- Non-parametric spectral GNN
  - A full parameter matrix: $F^{(l,j)} \in \mathbb{R}^{n \times f_l}$

- Function bases, e.g. B-splines, Chebyshev polynomials, Cayley polynomials
  - Formula: $F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top$
  - Base function of eigenvalues: $B \in \mathbb{R}^{n \times s_e} \quad B_{k,s} = \Phi_s(\lambda_k)$
  - Parameters: $W^{(l,s)} \in \mathbb{R}^{f_l \times f_{l+1}}$

s_e : total # filters

$k = 1, \ldots, n$

$s = 1, \ldots, s_e$

Spatial GNNs

- General formula

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

$H_{:v}^{(l)}$: v-th row,
all features of node v

Spatial GNNs

- General formula

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

$H_{:v}^{(l)}$: v-th row,
all features of node v

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

$C^{(s)} \in \mathbb{R}^{n \times n}$ is the s-th convolution support

Spatial GNNs

- General formula

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

$H_{:v}^{(l)}$: v-th row,
all features of node v

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

$C^{(s)} \in \mathbb{R}^{n\times n}$ is the s-th convolution support

- GCN $H^{(l+1)} = \sigma\Big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\Big)$

Spatial GNNs

- General formula

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

$H_{:v}^{(l)}$: v-th row, all features of node v

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

$C^{(s)} \in \mathbb{R}^{n \times n}$ is the s-th convolution support

- GCN $H^{(l+1)} = \sigma\Big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\Big)$
- GIN $H^{(l+1)} = \sigma((A + (1+\epsilon)I)H^{(l)}W^{(l)})$

Spatial GNNs

- General formula

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

$H_{:v}^{(l)}$: v-th row, all features of node v

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

$C^{(s)} \in \mathbb{R}^{n\times n}$ is the s-th convolution support

- GCN $H^{(l+1)} = \sigma\Big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\Big)$
- GIN $H^{(l+1)} = \sigma((A + (1+\epsilon)I)H^{(l)}W^{(l)})$
- GAT

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

$$e_{v,u} = \exp\left(\sigma(\mathbf{a}^{(l,s)}[H_{:v}^{(l)}W^{(l,s)}||H_{:u}^{(l)}W^{(l,s)}])\right)$$

Note: GAT also falls into this framework, but its convolution support is different from layer to layer

Rewrite Spectral GNNs

- From $$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathrm{diag}(F_i^{(l,j)}) U^\top H_i^{(l)}\right), \qquad \text{for } j \in \{1, \ldots, f_{l+1}\}.$$
$$F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top \qquad B_{k,s} = \Phi_s(\lambda_k)$$
- To $$H^{(l+1)} = \sigma\Big(\sum_s C^{(s)} H^{(l)} W^{(l,s)}\Big)$$

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

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathbf{diag}(F_i^{(l,j)}) U^\top H_i^{(l)}\right) \qquad F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top \quad B_{k,s} = \Phi_s(\lambda_k)$$

derivation

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

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathbf{diag}(F_i^{(l,j)}) U^\top H_i^{(l)}\right) \qquad F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top \quad B_{k,s} = \Phi_s(\lambda_k)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathbf{diag}\Big(\sum_{s=1}^{S} W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda})\Big) U^\top H_i^{(l)}\right)$$

derivation

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

$$H_j^{(l+1)} = \sigma \left(\sum_{i=1}^{f_l} U \mathbf{diag}(F_i^{(l,j)}) U^\top H_i^{(l)} \right) \qquad F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)} \right]^\top \quad B_{k,s} = \Phi_s(\lambda_k)$$

$$H_j^{(l+1)} = \sigma \left(\sum_{i=1}^{f_l} U \mathbf{diag}\Big(\sum_{s=1}^{S} W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda}) \Big) U^\top H_i^{(l)} \right)$$

$$H_j^{(l+1)} = \sigma \left(\sum_{s=1}^{S} \sum_{i=1}^{f_l} U \mathbf{diag}\Big(W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda}) \Big) U^\top H_i^{(l)} \right)$$

derivation

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

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathrm{diag}(F_i^{(l,j)}) U^\top H_i^{(l)}\right) \qquad F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top \quad B_{k,s} = \Phi_s(\lambda_k)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathrm{diag}\Big(\sum_{s=1}^{S} W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda})\Big) U^\top H_i^{(l)}\right)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{s=1}^{S}\sum_{i=1}^{f_l} U \mathrm{diag}\Big(W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda})\Big) U^\top H_i^{(l)}\right)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{s=1}^{S}\sum_{i=1}^{f_l} W_{i,j}^{(l,s)} U \mathrm{diag}(\Phi_s(\boldsymbol{\lambda})) U^\top H_i^{(l)}\right)$$

derivation

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

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathbf{diag}(F_i^{(l,j)}) U^\top H_i^{(l)}\right) \qquad F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top \quad B_{k,s} = \Phi_s(\lambda_k)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathbf{diag}\Big(\sum_{s=1}^{S} W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda})\Big) U^\top H_i^{(l)}\right)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{s=1}^{S}\sum_{i=1}^{f_l} U \mathbf{diag}\Big(W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda})\Big) U^\top H_i^{(l)}\right)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{s=1}^{S}\sum_{i=1}^{f_l} W_{i,j}^{(l,s)} U \mathbf{diag}(\Phi_s(\boldsymbol{\lambda})) U^\top H_i^{(l)}\right)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{s=1}^{S}\sum_{i=1}^{f_l} W_{i,j}^{(l,s)} C^{(s)} H_i^{(l)}\right) \qquad C^{(s)} = U \mathbf{diag}(\Phi_s(\boldsymbol{\lambda})) U^\top$$

derivation

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

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathbf{diag}(F_i^{(l,j)}) U^\top H_i^{(l)}\right) \qquad F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \ldots, W_{i,j}^{(l,s_e)}\right]^\top \quad B_{k,s} = \Phi_s(\lambda_k)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{i=1}^{f_l} U \mathbf{diag}\Big(\sum_{s=1}^{S} W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda})\Big) U^\top H_i^{(l)}\right)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{s=1}^{S}\sum_{i=1}^{f_l} U \mathbf{diag}\Big(W_{i,j}^{(l,s)} \Phi_s(\boldsymbol{\lambda})\Big) U^\top H_i^{(l)}\right)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{s=1}^{S}\sum_{i=1}^{f_l} W_{i,j}^{(l,s)} U \mathbf{diag}(\Phi_s(\boldsymbol{\lambda})) U^\top H_i^{(l)}\right)$$

$$H_j^{(l+1)} = \sigma\left(\sum_{s=1}^{S}\sum_{i=1}^{f_l} W_{i,j}^{(l,s)} C^{(s)} H_i^{(l)}\right) \qquad C^{(s)} = U \mathbf{diag}(\Phi_s(\boldsymbol{\lambda})) U^\top$$

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

derivation

Summary

- Looks like we just did a lot of matrix notation manipulation, what have we done?

Summary

- Looks like we just did a lot of matrix notation manipulation, what have we done?
- Spectral-designed GNNs: start with the definition of convolution and parametrize the filter function using a function basis

Summary

- Looks like we just did a lot of matrix notation manipulation, what have we done?
- Spectral-designed GNNs: start with the definition of convolution and parametrize the filter function using a function basis
- Spatial-designed GNNs: collect information from neighbors

Summary

- Looks like we just did a lot of matrix notation manipulation, what have we done?
- Spectral-designed GNNs: start with the definition of convolution and parametrize the filter function using a function basis
- Spatial-designed GNNs: collect information from neighbors

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

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 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.*

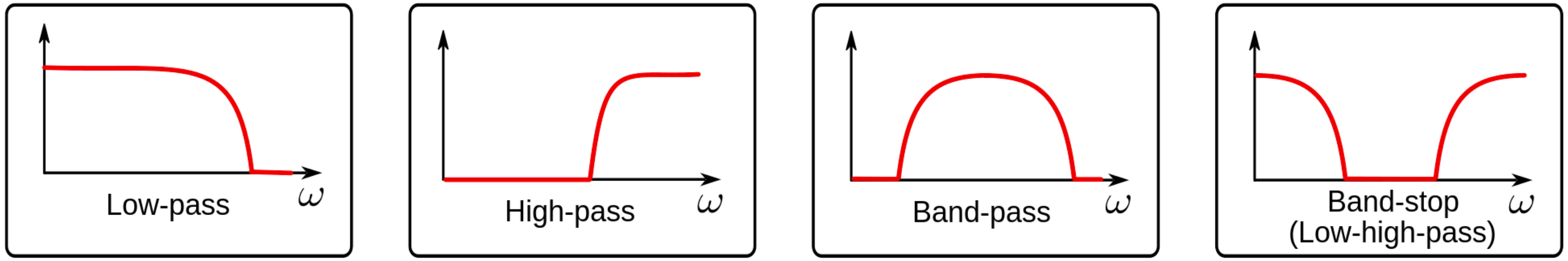
Frequency Response

- A measure of magnitude and phase as a function of frequency

Frequency Response

- A measure of magnitude and phase as a function of frequency

- Filters



Analyzing The Expressive Power of GNNs

A Spectral Perspective

Shichang Zhang

01-19-2021

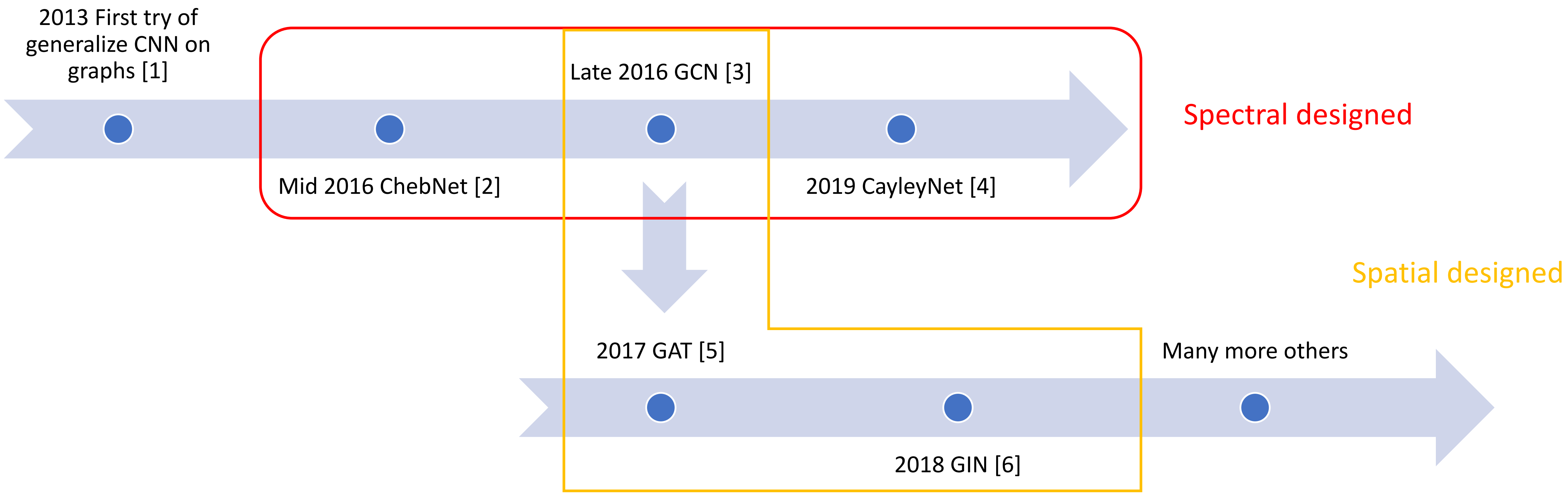
Outline

- Introduction and motivation ✓
- Convolution, graph signal, and graph Fourier transformation ✓
- Spectral and spatial GNNs ✓
- Frequency profile analysis

Review Notations

$A \in \{0,1\}^{n\times n}$	: adjacency matrix	$\tilde{A} = A + I$	: adjacency matrix with self-loop
$D \in \mathbb{R}^{n\times n}$	: diagonal degree matrix	$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$	: degree matrix with self-loop
$L = I - D^{-1/2}AD^{-1/2}$	: normalized Laplacian	$L = U\mathbf{diag}(\boldsymbol{\lambda})U^T$	: eigendecomposition $U \in \mathbb{R}^{n\times n}$ $\boldsymbol{\lambda} \in \mathbb{R}^n$
$\mathbf{diag}(.)$	: vector -> diagonal matrix	$diag^{-1}(.)$	: diagonal matrix -> vector
$X \in \mathbb{R}^{n\times f_0}$	: node features	$X_i \in \mathbb{R}^n$	: the i-th column (feature)
$H^{(l)} \in \mathbb{R}^{n\times f_l}$	: node representations in layer l	$H^{(0)} = X$	

Review Timeline of GNNs



[1] Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203.*

[2] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems, 29*, 3844-3852.

[3] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907.*

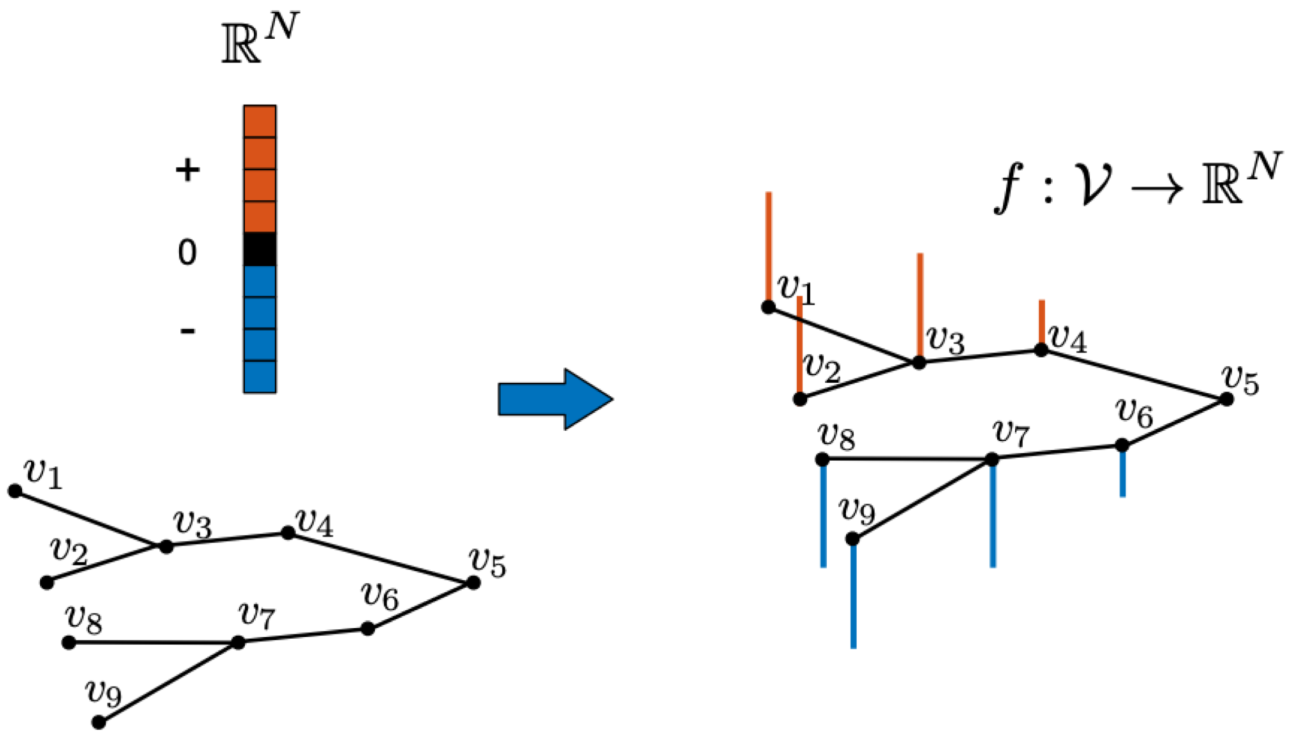
[4] Levie, R., Monti, F., Bresson, X., & Bronstein, M. M. (2018). Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Transactions on Signal Processing, 67(1), 97-109.

[5] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903

[6] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks?. arXiv preprint arXiv:1810.00826.

Review Graph Signals

- A function defined on the vertices of a graph



Node features:

$$X \in \mathbb{R}^{n \times f_0}$$

Each column (feature) is a signal:

$$X_i \in \mathbb{R}^n$$

Review Fourier Transformation

- Classical Fourier transformation
  - From time/space domain to frequency domain
  - Formula:

$$\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{2\pi i \xi t} dt$$

- Graph Fourier transformation
  - From vertex domain to graph spectral domain
  - Formula:

$$\hat{f}(\lambda_l) = \langle f, u_l \rangle = \sum_{i=1}^{n} f(i) u_l^*(i) \longrightarrow \hat{f} = U^T f$$

  - Inverse formula:

$$f(i) = \sum_{l=1}^{n} \hat{f}(\lambda_l) u_l(i) \longrightarrow f = U \hat{f}$$

λ_l : l-th eigenvalue of L

u_l : l-th eigenvector of L

Review Convolution and Graph Fourier Transformation

- Generalize graph convolution using graph Fourier transformation
  - Elementwise notation

$$(f * h)(i) = \sum_{l=1}^{n} \hat{f}(\lambda_l)\hat{h}(\lambda_l)u_l(i)$$

  - Matrix notation

$$f * h = U(\hat{f} \odot \hat{h}) = U((U^T f) \odot (U^T h))$$

  - Further equals to

$$f * h = U diag(\hat{h}(\boldsymbol{\lambda}))U^T f$$

Review Spatial GNNs

- General formula

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

$H_{:v}^{(l)}$: v-th row,
all features of node v

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

$C^{(s)} \in \mathbb{R}^{n \times n}$ is the s-th convolution support

- GCN $H^{(l+1)} = \sigma\Big(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\Big)$
- GIN $H^{(l+1)} = \sigma((A + (1+\epsilon)I)H^{(l)}W^{(l)})$
- GAT

$\Big(C^{(l,s)}\Big)_{v,u} = \alpha_{v,u}$ is the attention score between node v and u

Note: GAT also falls into this framework, but its convolution support is different from layer to layer

Review Spectral GNNs

- General spectral formula

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

  - Non-parametric model

$$F^{(l,j)} \in \mathbb{R}^{n \times f_l}$$

  - Model with base functions

$$F_i^{(l,j)} = B\left[W_{i,j}^{(l,1)}, \dots, W_{i,j}^{(l,s_e)}\right]^\top \qquad B_{k,s} = \Phi_s(\lambda_k)$$

- To the general formula of spatial and spectral

$$H^{(l+1)} = \sigma\left(\sum_s C^{(s)} H^{(l)} W^{(l,s)}\right) \qquad C^{(s)} = U\text{diag}(\Phi_s(\boldsymbol{\lambda}))U^\top$$

$W^{(l,s)} \in \mathbb{R}^{f_l \times f_{l+1}}$

s_e : total # filters

$k = 1, \dots, n$

$s = 1, \dots, s_e$

Summary

- Spectral-designed GNNs: start with the definition of convolution and parametrize the filter function using a function basis
- Spatial-designed GNNs: collect information from neighbors
- General formula for both cases

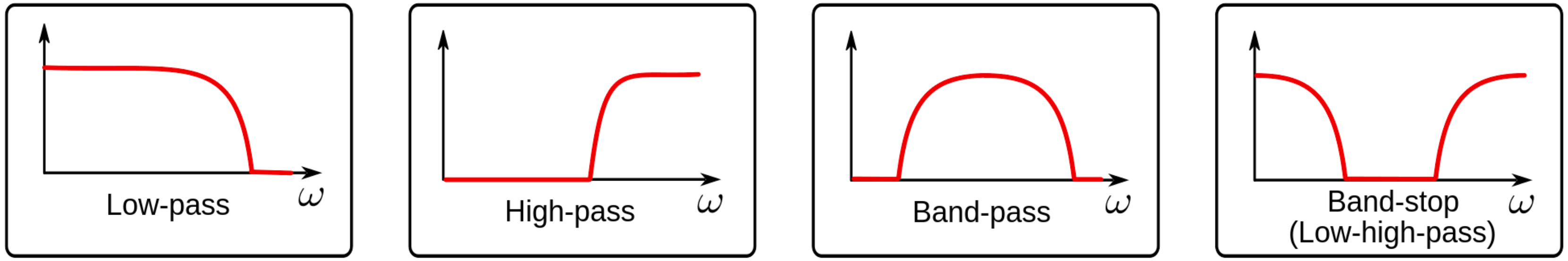
$$H^{(l+1)} = \sigma\Big(\sum_s C^{(s)} H^{(l)} W^{(l,s)}\Big)$$

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 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.*

Frequency Response

- A measure of magnitude and phase as a function of frequency

- Filters



Frequency Profile

Corollary 1.1. ***The frequency profile*** *of any given graph convolution support* $C^{(s)}$ *can be defined in spectral domain by*

$$\Phi_s(\boldsymbol{\lambda}) = diag^{-1}(U^\top C^{(s)} U). \tag{7}$$

where $diag^{-1}(.)$ *returns the vector made of the diagonal elements from the given matrix.*

Frequency Profile

Corollary 1.1. ***The frequency profile*** *of any given graph convolution support* $C^{(s)}$ *can be defined in spectral domain by*

$$\Phi_s(\boldsymbol{\lambda}) = diag^{-1}(U^\top C^{(s)} U). \tag{7}$$

where $diag^{-1}(.)$ *returns the vector made of the diagonal elements from the given matrix.*

- A measure of magnitude as a function of eigenvalues

Frequency Profile

Corollary 1.1. ***The frequency profile*** *of any given graph convolution support $C^{(s)}$ can be defined in spectral domain by*

$$\Phi_s(\boldsymbol{\lambda}) = diag^{-1}(U^\top C^{(s)} U). \tag{7}$$

where $diag^{-1}(.)$ returns the vector made of the diagonal elements from the given matrix.

- A measure of magnitude as a function of eigenvalues
- For spectral-designed GNNs, the frequency profile is the frequency response.

Frequency Profile

Corollary 1.1. ***The frequency profile*** *of any given graph convolution support $C^{(s)}$ can be defined in spectral domain by*

$$\Phi_s(\boldsymbol{\lambda}) = diag^{-1}(U^\top C^{(s)} U). \tag{7}$$

where $diag^{-1}(.)$ returns the vector made of the diagonal elements from the given matrix.

- A measure of magnitude as a function of eigenvalues
- For spectral-designed GNNs, the frequency profile is the frequency response.
- For spatial-designed GNNs, $U^\top C^{(s)} U$ is not diagonal, we further define the **full frequency profile** as $\Phi_s = U^\top C^{(s)} U$

Analyze Frequency Profile of ChebNets

- Chebyshev polynomial is recursively defined on [-1, 1]

$$T_0(x) = 1 \qquad T_1(x) = x$$

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

Analyze Frequency Profile of ChebNets

- Chebyshev polynomial is recursively defined on [-1, 1]

$$T_0(x) = 1 \qquad T_1(x) = x$$

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

- Frequency Profile

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

$$\Phi_k(\boldsymbol{\lambda}) = 2\Phi_2(\boldsymbol{\lambda})\Phi_{k-1}(\boldsymbol{\lambda}) - \Phi_{k-2}(\boldsymbol{\lambda})$$

Analyze Frequency Profile of ChebNets

- Chebyshev polynomial is recursively defined on [-1, 1]

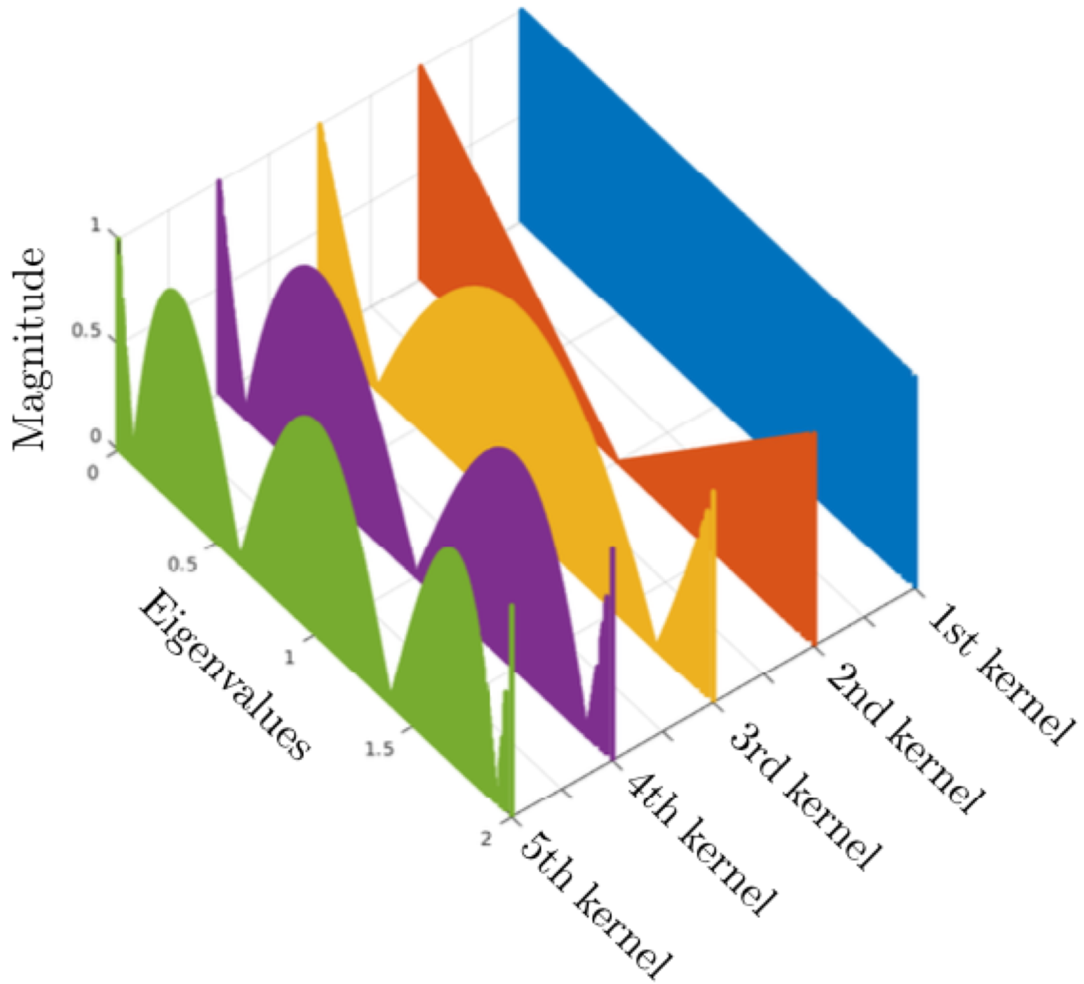
$$T_0(x) = 1 \qquad T_1(x) = x$$

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

- Frequency Profile

$$\boldsymbol{\Phi_1(\lambda) = 1},\ \ \boldsymbol{\Phi_2(\lambda) = \frac{2\lambda}{\lambda_{\max}} - 1}$$

$$\Phi_k(\boldsymbol{\lambda}) = 2\Phi_2(\boldsymbol{\lambda})\Phi_{k-1}(\boldsymbol{\lambda}) - \Phi_{k-2}(\boldsymbol{\lambda})$$



(a) First 5 ChebNet supports

Analyze Frequency Profile of CayleyNets

- CayleyNets are able to detect narrow frequency bands of importance and have greater flexibility.

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

Analyze Frequency Profile of CayleyNets

- CayleyNets are able to detect narrow frequency bands of importance and have greater flexibility.

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

- Frequency Profile

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

$$\theta(x) = atan2(-1, x) - atan2(1, x)$$

Analyze Frequency Profile of CayleyNets

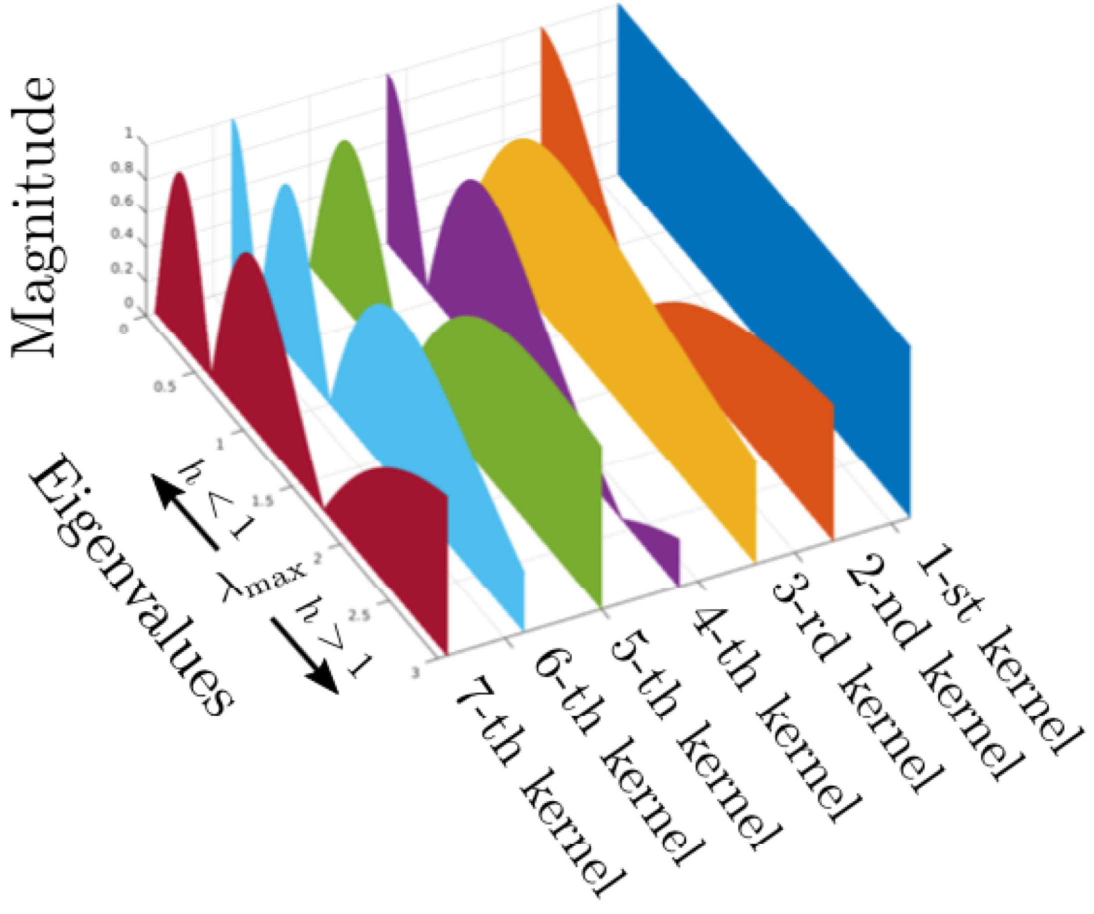
- CayleyNets are able to detect narrow frequency bands of importance and have greater flexibility.

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

- Frequency Profile

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

$$\theta(x) = atan2(-1, x) - atan2(1, x)$$



(b) First 7 CayleyNet support

Analyze Frequency Profile of GCN

- GCN on regular graph

Proposition 2. $C_{GCN} = (D+I)^{-1/2}(A+I)(D+I)^{-1/2}$ *frequency response is* $\Phi_{GCN}(\boldsymbol{\lambda}) = \mathbf{1} - \frac{p}{p+1}\boldsymbol{\lambda}$ *for regular graphs whose node degrees are* p*.*

Goal: $\Phi_{GCN}(\boldsymbol{\lambda}) = \mathbf{1} - \frac{p}{p+1}\boldsymbol{\lambda}$

$$D = pI \qquad A = pI - pL \qquad C_{GCN} = (D+I)^{-1/2}(A+I)(D+I)^{-1/2}$$

derivation

Goal: $\Phi_{GCN}(\boldsymbol{\lambda}) = \mathbf{1} - \frac{p}{p+1}\boldsymbol{\lambda}$

$D = pI \qquad A = pI - pL \qquad C_{GCN} = (D+I)^{-1/2}(A+I)(D+I)^{-1/2}$

$$C_{GCN} = \frac{pI - pL + I}{p+1} = I - \frac{p}{p+1}L$$

derivation

Goal: $\Phi_{GCN}(\boldsymbol{\lambda}) = \mathbf{1} - \frac{p}{p+1}\boldsymbol{\lambda}$

$D = pI \qquad A = pI - pL \qquad C_{GCN} = (D+I)^{-1/2}(A+I)(D+I)^{-1/2}$

$$C_{GCN} = \frac{pI - pL + I}{p+1} = I - \frac{p}{p+1}L$$

$$= U\text{diag}(\mathbf{1} - \frac{p}{p+1}\boldsymbol{\lambda})U^\top$$

derivation

Analyze Frequency Profile of GCN

- GCN on regular graph

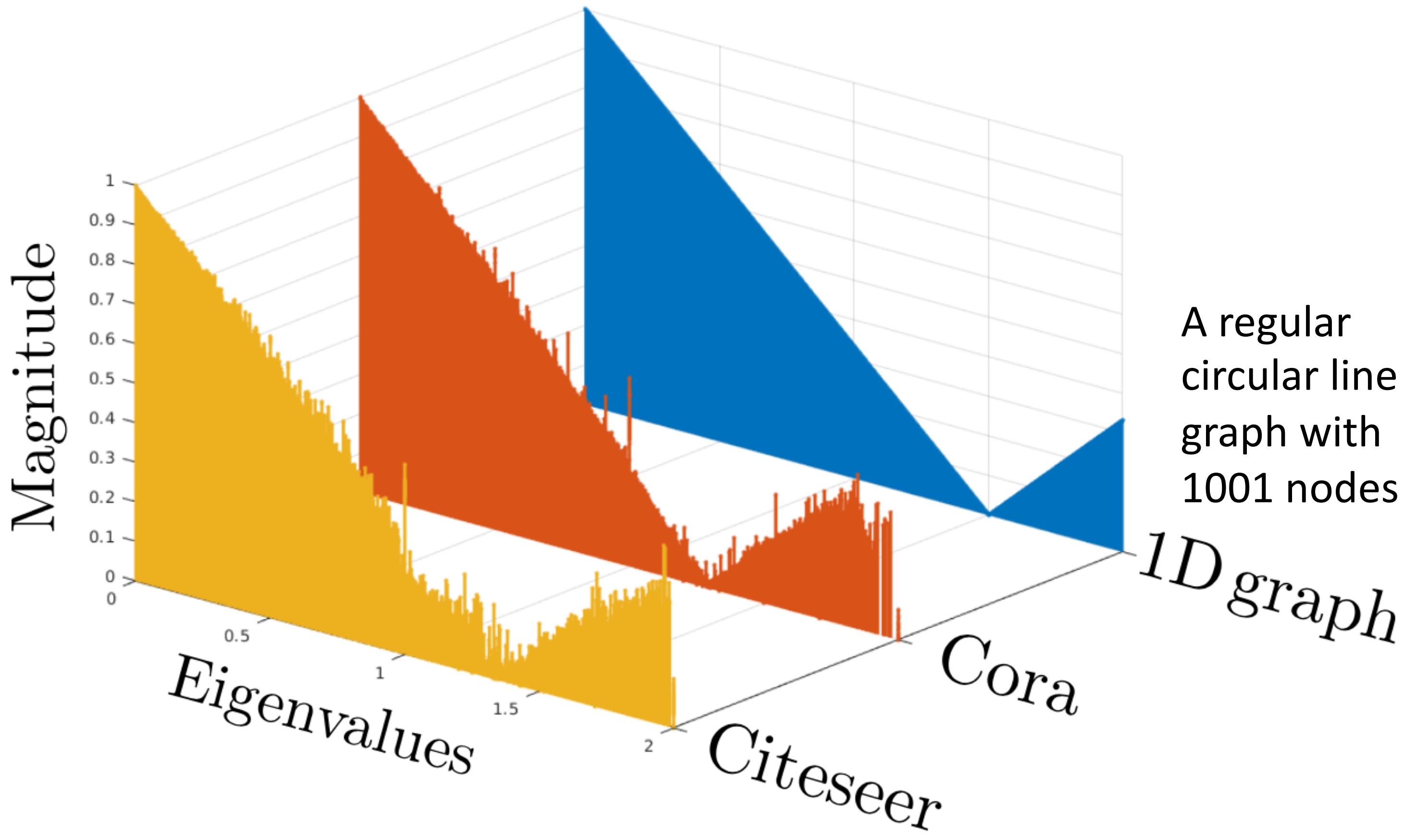
Proposition 2. $C_{GCN} = (D+I)^{-1/2}(A+I)(D+I)^{-1/2}$ *frequency response is* $\Phi_{GCN}(\boldsymbol{\lambda}) = \mathbf{1} - \frac{p}{p+1}\boldsymbol{\lambda}$ *for regular graphs whose node degrees are* p.

- GCN on general graph

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

Analyze Frequency Profile of GCN

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



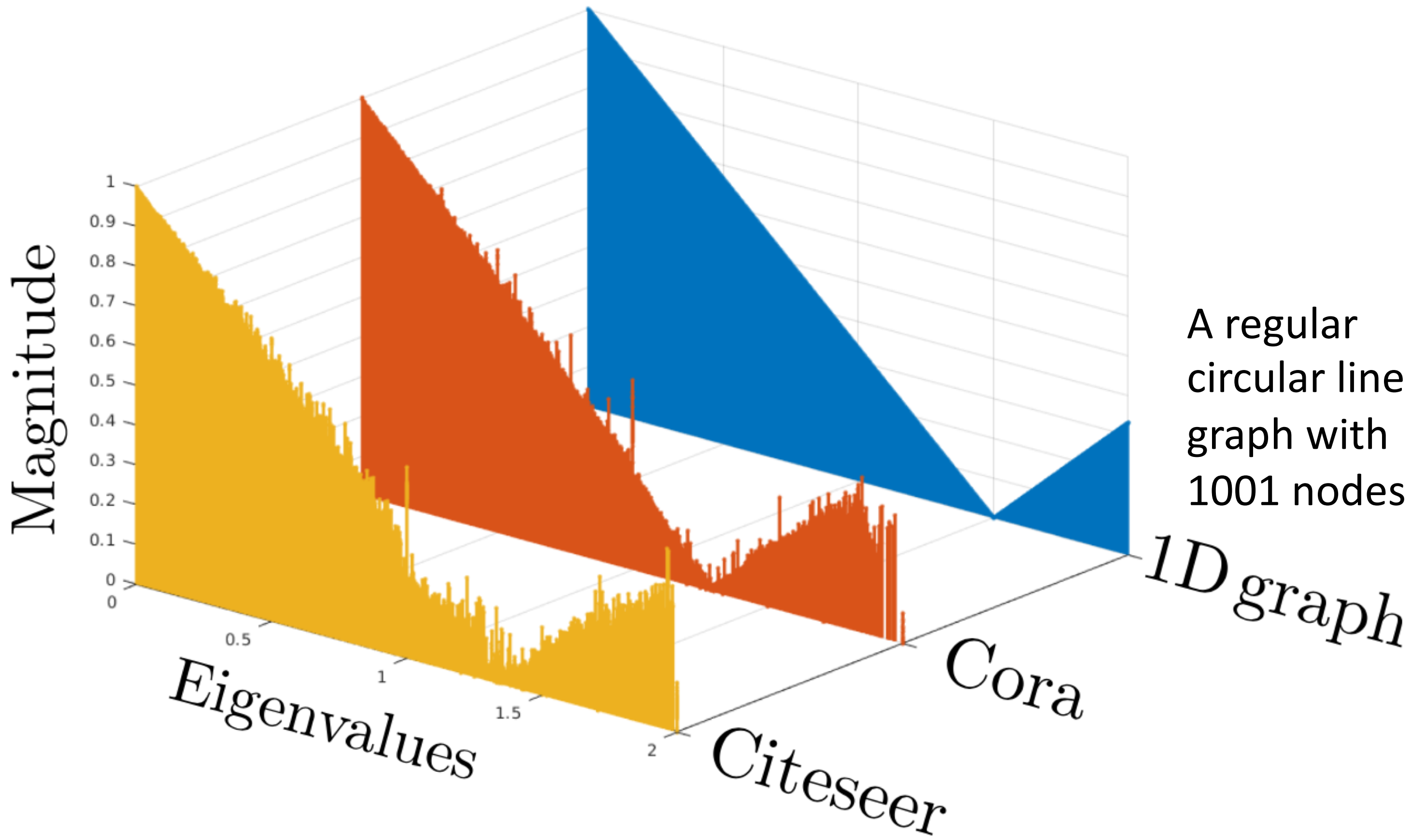
A regular circular line graph with 1001 nodes

(a) GCN frequency profiles

Analyze Frequency Profile of GCN

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

- GCN works as low-pass filter and does not cover the whole spectrum.
- GCN is not able to learn relations that are represented by high-pass or band-pass filtering



A regular circular line graph with 1001 nodes

(a) GCN frequency profiles

Full Frequency Profile of GCN

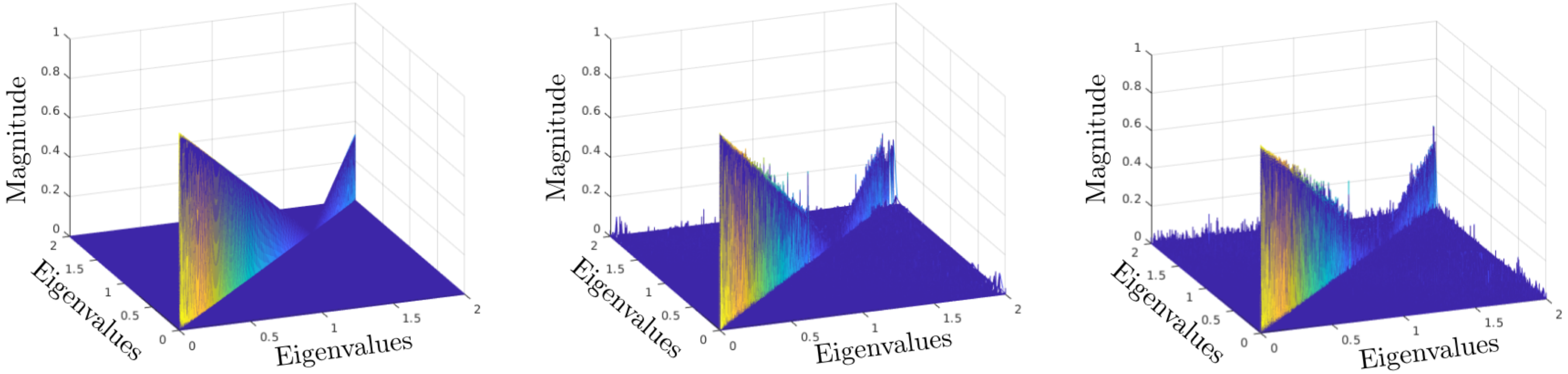


Figure 5: Full frequency response of GCN on 1D, Cora and CiteSeer graphs

Analyze Frequency Profile of GIN

- GIN on regular graphs

Proposition 3. *For* $C_{GIN} = A + (1+\epsilon)I$*, the frequency response is* $\Phi_{GIN}(\boldsymbol{\lambda}) = p\left(\frac{\mathbf{1}+\epsilon}{p} + 1 - \boldsymbol{\lambda}\right)$ *for regular graphs, where* p *is the node degrees.*

Goal: $\Phi_{GIN}(\boldsymbol{\lambda}) = p\left(\frac{\mathbf{1}+\epsilon}{p} + 1 - \boldsymbol{\lambda}\right)$

$D = pI \qquad A = pI - pL \qquad C_{GIN} = A + (1+\epsilon)I$

derivation

Goal: $\Phi_{GIN}(\boldsymbol{\lambda}) = p\left(\frac{\mathbf{1}+\epsilon}{p} + 1 - \boldsymbol{\lambda}\right)$

$D = pI \qquad A = pI - pL \qquad C_{GIN} = A + (1+\epsilon)I$

$$C_{GIN} = (p + 1 + \epsilon)I - pL$$

derivation

Goal: $\Phi_{GIN}(\boldsymbol{\lambda}) = p\left(\frac{\mathbf{1}+\epsilon}{p} + 1 - \boldsymbol{\lambda}\right)$

$D = pI \qquad A = pI - pL \qquad C_{GIN} = A + (1+\epsilon)I$

$$C_{GIN} = (p+1+\epsilon)I - pL = (p+1+\epsilon)UIU^{\top} - pU\mathrm{diag}(\boldsymbol{\lambda})U^{\top}$$

derivation

Goal: $\Phi_{GIN}(\boldsymbol{\lambda}) = p\left(\frac{\mathbf{1}+\epsilon}{p} + 1 - \boldsymbol{\lambda}\right)$

$D = pI \qquad A = pI - pL \qquad C_{GIN} = A + (1+\epsilon)I$

$$C_{GIN} = (p+1+\epsilon)I - pL = (p+1+\epsilon)UIU^\top - pU\mathbf{diag}(\boldsymbol{\lambda})U^\top$$

$$= U\mathbf{diag}(p+\epsilon+\mathbf{1}-p\boldsymbol{\lambda})U^\top$$

derivation

Analyze Frequency Profile of GIN

- GIN on regular graphs

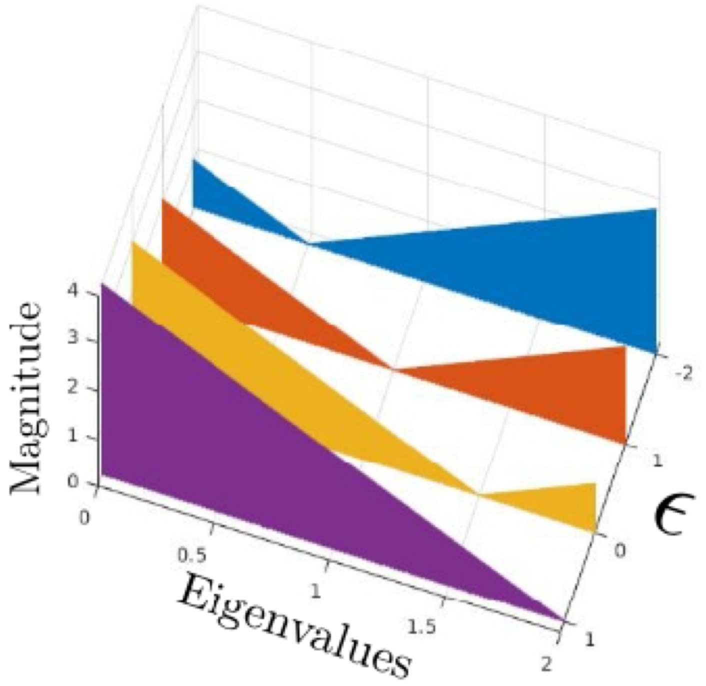
Proposition 3. *For* $C_{GIN} = A + (1+\epsilon)I$*, the frequency response is* $\Phi_{GIN}(\boldsymbol{\lambda}) = p\left(\frac{\mathbf{1}+\epsilon}{p} + 1 - \boldsymbol{\lambda}\right)$ *for regular graphs, where* p *is the node degrees.*

- GIN on general graphs

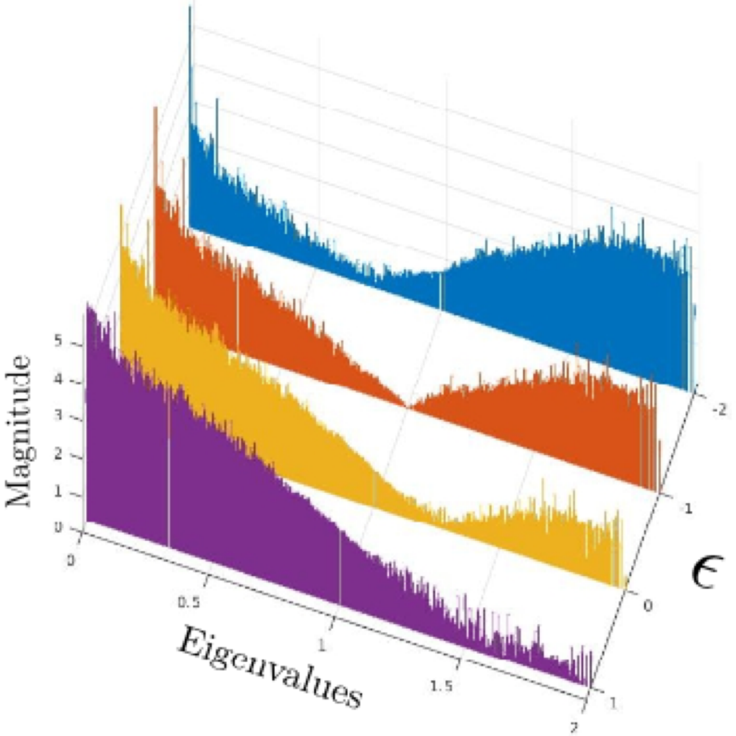
$$\Phi_{GIN}(\boldsymbol{\lambda}) \approx \overline{p}\left(\frac{1+\epsilon}{\overline{p}} + \mathbf{1} - \boldsymbol{\lambda}\right)$$

Analyze Frequency Profile of GIN

$$\Phi_{GIN}(\boldsymbol{\lambda}) = p\left(\frac{\mathbf{1}+\epsilon}{p} + 1 - \boldsymbol{\lambda}\right)$$



(b) GIN on 1D

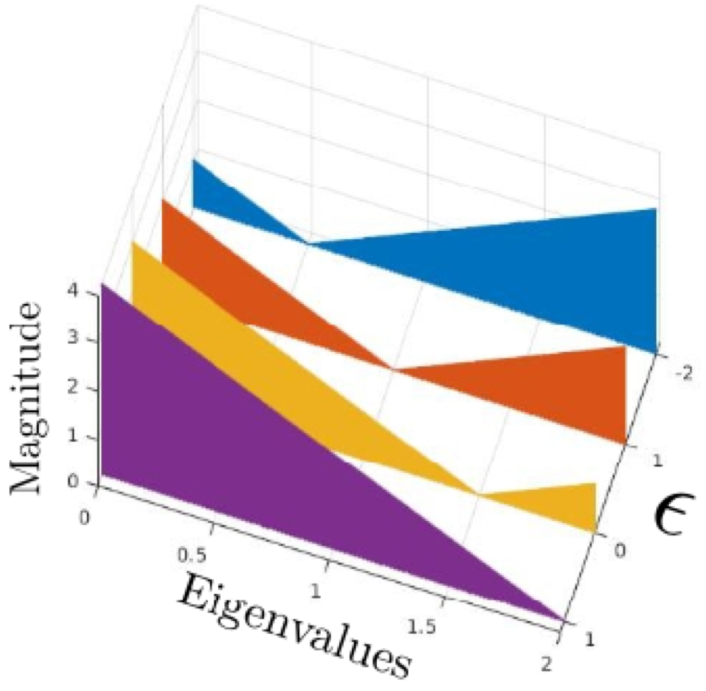


(c) GIN on CiteSeer

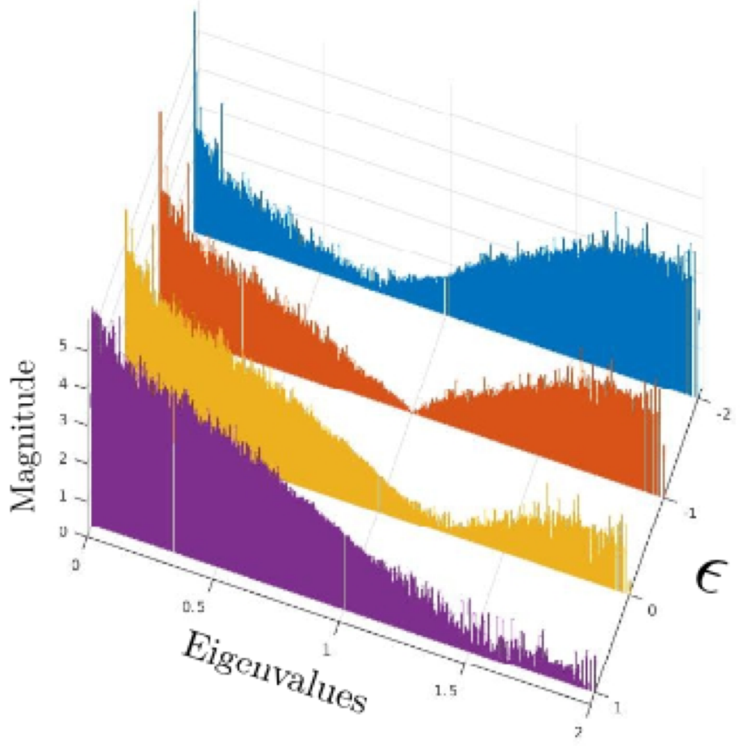
Analyze Frequency Profile of GIN

$$\Phi_{GIN}(\boldsymbol{\lambda}) = p\left(\frac{\mathbf{1}+\epsilon}{p} + 1 - \boldsymbol{\lambda}\right)$$

- GIN works as a filter covers a specific frequency corresponding to ϵ
- GIN is more expressive than GCN, but it still doesn't cover the whole spectrum



(b) GIN on 1D



(c) GIN on CiteSeer

Analyze Frequency Profile of GAT

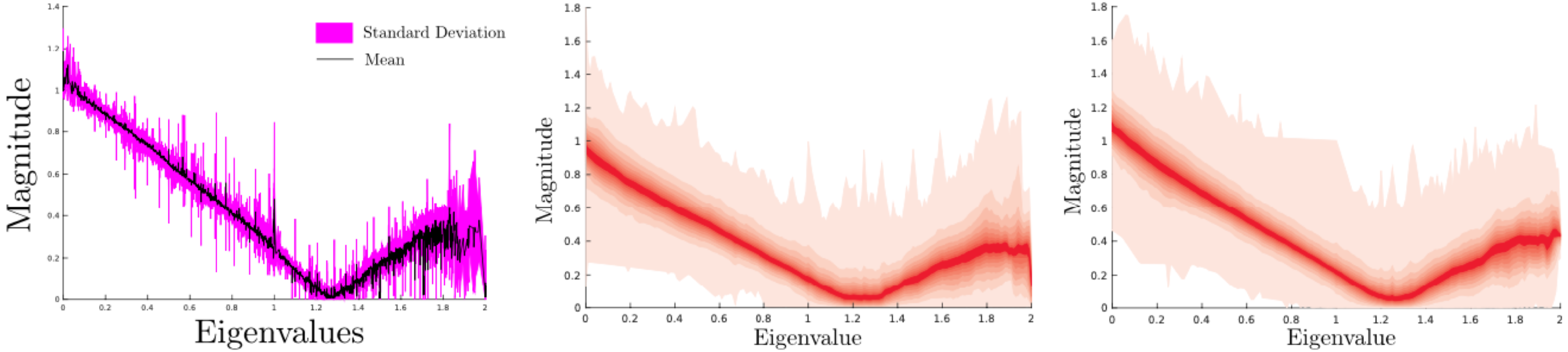
- The convolution support of GAT depends on node features, which makes it hard to derive a closed form frequency profile formula, but we can still check the empirical result.

Analyze Frequency Profile of GAT

- The convolution support of GAT depends on node features, which makes it hard to derive a closed form frequency profile formula, but we can still check the empirical result.

Randomly simulate attention weights

Train weights of all attention heads



(a) Expected frequency response from Simulation on Cora

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

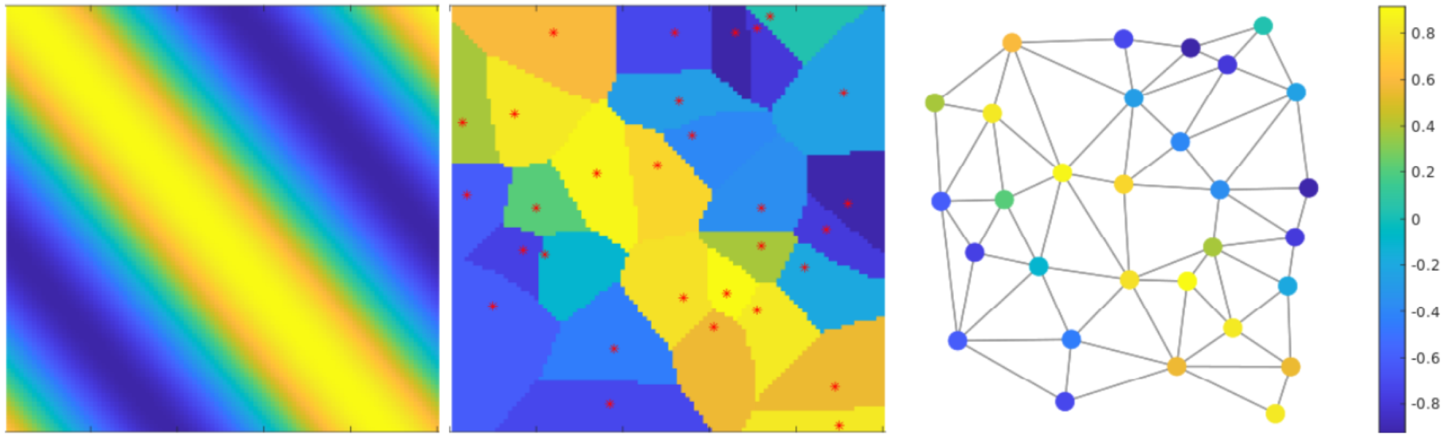
(c) Heat density map of learned frequency response on PROTEINS

Why Do GCN, GIN, and GAT Work Well?

- GCN, GIN and GAT obtain SOTA performance on reference node classification datasets such as Cora, CiteSeer and Pubmed. These good results are induced by the nature of the graphs to be processed. Indeed, citation network problems, which are heavily assortative, are inherently low-pass filtering problems.

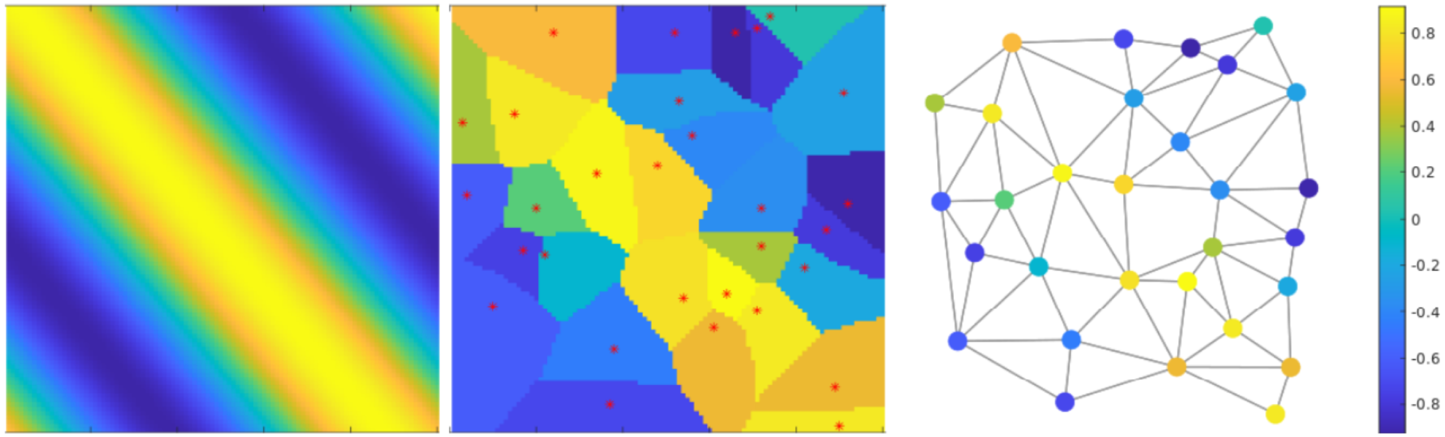
In What Case Will GCN, GIN, and GAT Fail?

- Pattern classification
  - Set up: generate images of random frequency patterns by a sinusoidal function with frequency in [1-5]. 0: frequency in [2-2.5] or [4-4.5]. 1: otherwise.



In What Case Will GCN, GIN, and GAT Fail?

- Pattern classification
  - Set up: generate images of random frequency patterns by a sinusoidal function with frequency in [1-5]. 0: frequency in [2-2.5] or [4-4.5]. 1: otherwise.



  - Result

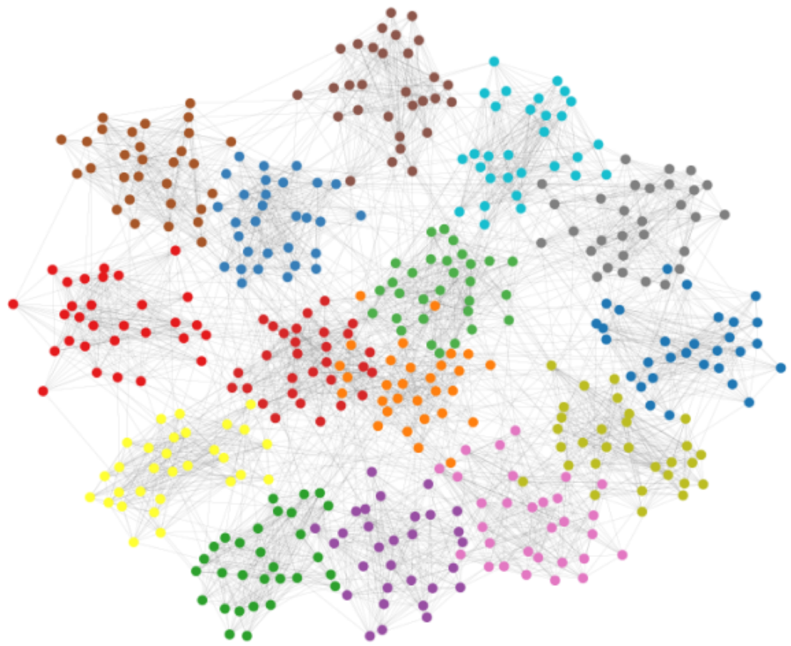
	MLP	GCN	GIN	GAT	ChebNet
Accuracy	50	77.90	87.60	85.30	98.2
Loss	0.69	0.454	0.273	0.324	0.062

Limitation of ChebNets?

- It is worth noting that, if we use enough convolution kernels, the frequency response of ChebNet kernels covers nearly all frequency profiles. However, these frequency responses are not specific to special bands of frequency.

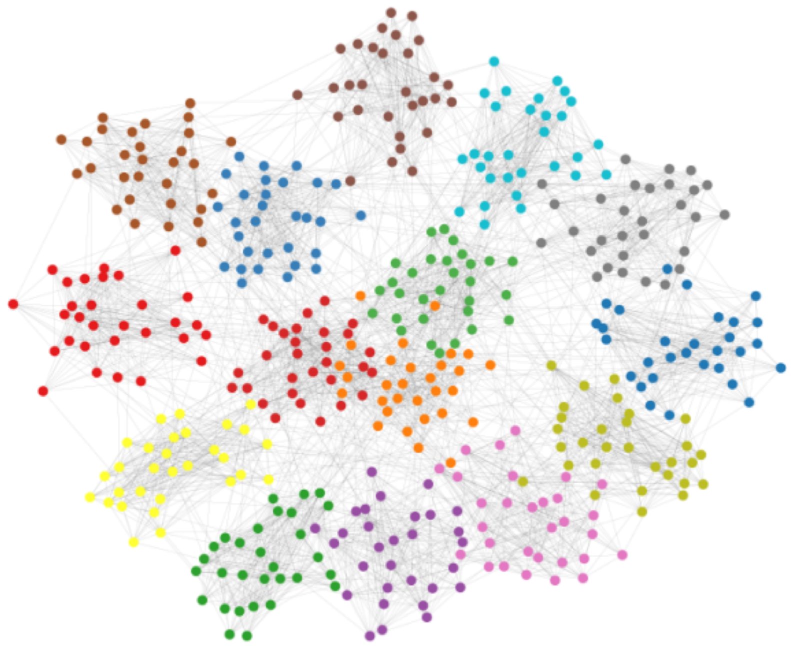
ChebNets v.s. CayleyNets

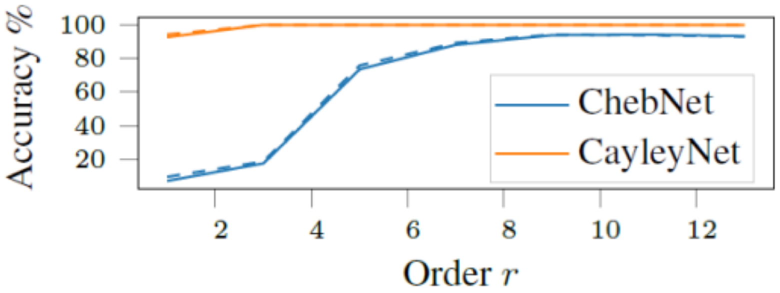
- Community detection of on a synthetic graph with 15 communities



ChebNets v.s. CayleyNets

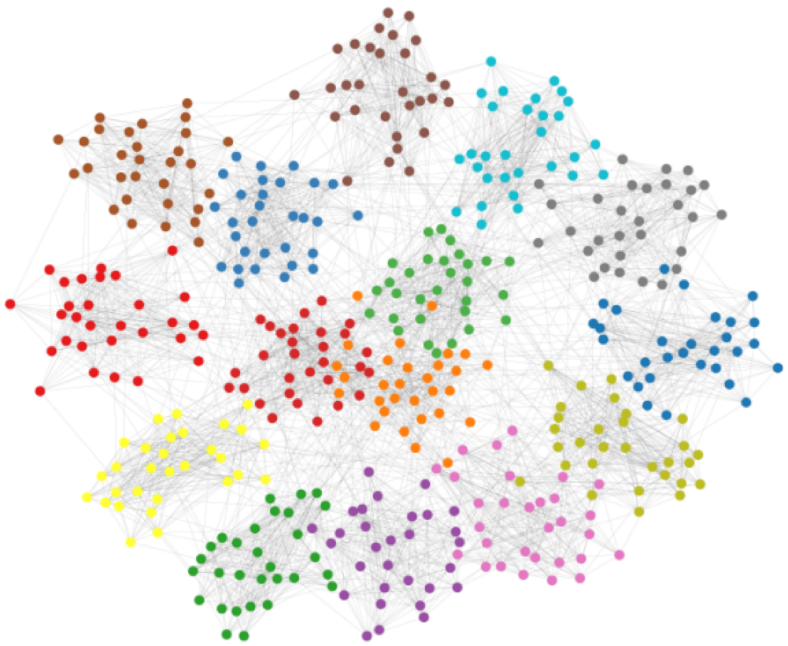
- Community detection of on a synthetic graph with 15 communities

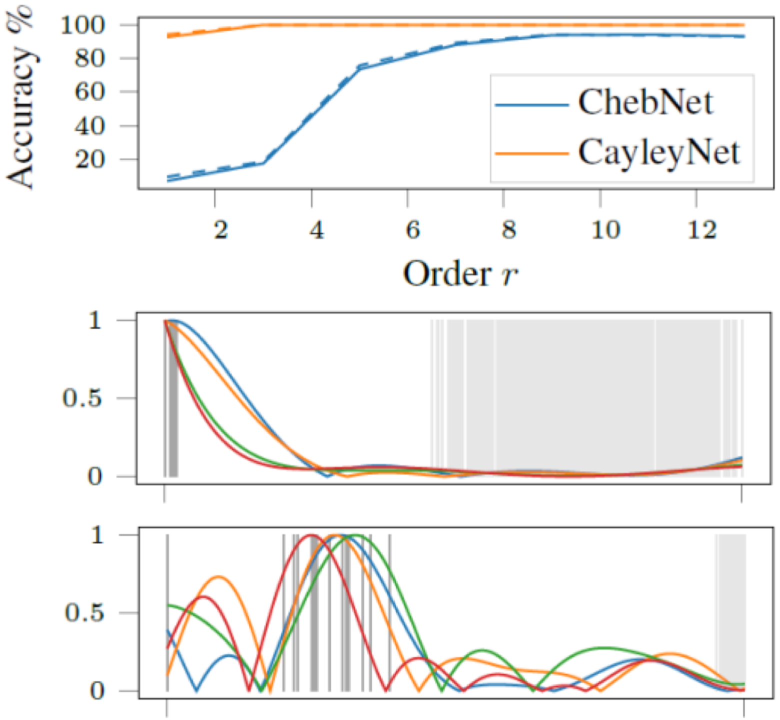




ChebNets v.s. CayleyNets

- Community detection of on a synthetic graph with 15 communities



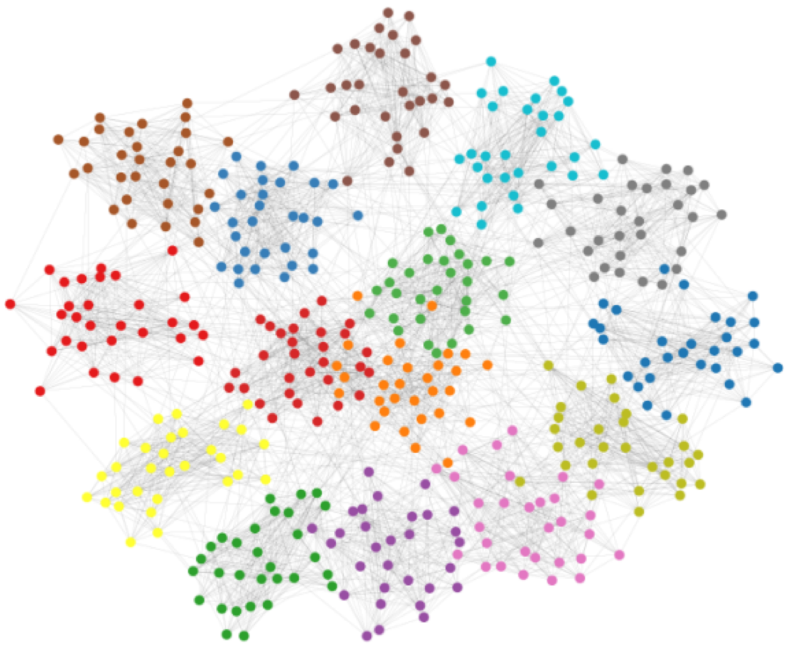


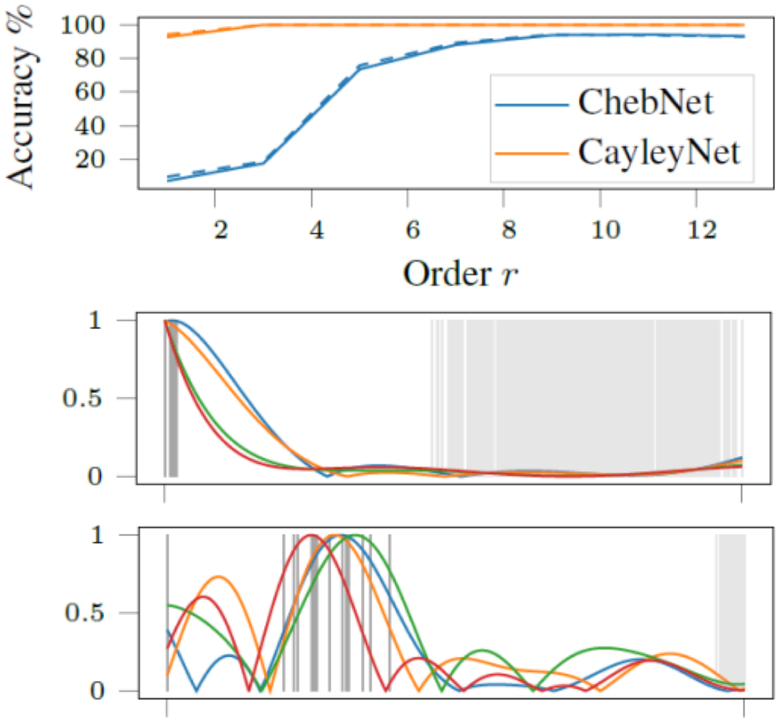
ChebNet $\tilde{\boldsymbol{\lambda}} = \frac{2\boldsymbol{\lambda}}{\lambda_{\max}} - \mathbf{1}$

CayleyNet $\tilde{\boldsymbol{\lambda}} = \frac{h\lambda - \mathbf{i}}{h\lambda + \mathbf{i}}$

ChebNets v.s. CayleyNets

- Community detection of on a synthetic graph with 15 communities
- CayleyNets are able to detect narrow frequency bands of importance, and thus have greater flexibility





ChebNet $\tilde{\boldsymbol{\lambda}} = \frac{2\boldsymbol{\lambda}}{\lambda_{\max}} - \mathbf{1}$

CayleyNet $\tilde{\boldsymbol{\lambda}} = \frac{h\lambda - \mathbf{i}}{h\lambda + \mathbf{i}}$

Conclusion

- From a spectral perspective, current GNNs are limited
- To achieve better performance
  - Use the most suitable model for a specific problem
  - Develop more expressive model architecture

Reference

- Patricia Xiao, Reading group 12.04.2018 http://web.cs.ucla.edu/~patricia.xiao/files/Reading_Group_20181204.pdf

- [1] Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.
- [2] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, *29*, 3844-3852.
- [3] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- [4] Levie, R., Monti, F., Bresson, X., & Bronstein, M. M. (2018). Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Transactions on Signal Processing, 67(1), 97-109.
- [5] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903
- [6] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks?. arXiv preprint arXiv:1810.00826.
- Analyzing the Expressive Power of Graph Neural Networks in a Spectral Perspective