Day 3: Equivariance and Stability to Deformations

Charilaos I. Kanatsoulis, Navid NaderiAlizadeh,

Alejandro Parada-Mayorga, Alejandro Ribeiro, and Luana Ruiz

gnn.seas.upenn.edu

2023 International Conference on Acoustics, Speech, and Signal Processing

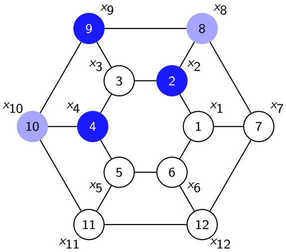
Rhodes, Greece – June 6, 2023

Permutation Equivariance, Stability, and Representation Power of Graph Neural Networks

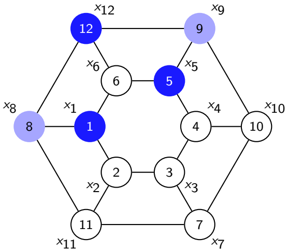
- We will start demystifying the success of Graph Neural Networks by studying their fundamental properties.
- We will show that Graph Neural Networks are equivariant to permutations, stable, and highly expressive.

- If $(\mathbf{S}, \mathbf{x})$ is a graph signal, $(\mathbf{P}^T\mathbf{S}\mathbf{P}, \mathbf{P}^T\mathbf{x})$ is a relabeling of $(\mathbf{S}, \mathbf{x})$. Same signal. Different names

Graph signal $\mathbf{x}$ Supported on $\mathbf{S}$

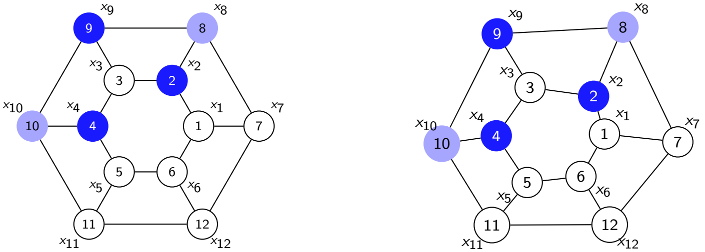


Graph signal $\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x}$ supported on $\hat{\mathbf{S}} = \mathbf{P}^T\mathbf{S}\mathbf{P}$



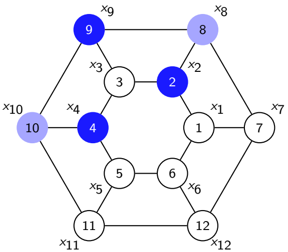
- Processing of isomorphic graphs and graph signals with Graph Neural Networks is label-independent.

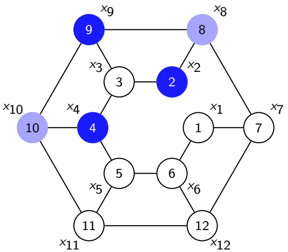
- Graphs are not isomorphic but close to isomorphic $\Rightarrow$ perturbed versions of each other



- We will show conditions for stability to deformations $\Rightarrow$ Approximate (close to) equivariance

- Graphs are not isomorphic





- We will show that a Graph Neural Network will produce non-isomorphic representations for the graphs.

Permutation Equivariance of Graph Neural Networks

- We will show that graph neural networks are equivariant to permutations

Definition (Permutation matrix)

A square matrix $\mathbf{P}$ is a permutation matrix if it has binary entries so that $\mathbf{P} \in \{0,1\}^{n \times n}$ and it further satisfies $\mathbf{P}\mathbf{1} = \mathbf{1}$ and $\mathbf{P}^T\mathbf{1} = \mathbf{1}$.

- The product $\mathbf{P}^T\mathbf{x}$ reorders the entries of the vector $\mathbf{x}$.
- The product $\mathbf{P}^T\mathbf{S}\mathbf{P}$ is a consistent reordering of the rows and columns of $\mathbf{S}$

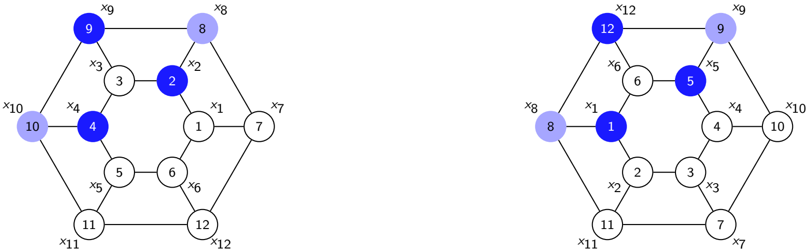
Definition (Permutation matrix)

A square matrix $\mathbf{P}$ is a permutation matrix if it has binary entries so that $\mathbf{P} \in \{0,1\}^{n \times n}$ and it further satisfies $\mathbf{P}\mathbf{1} = \mathbf{1}$ and $\mathbf{P}^T\mathbf{1} = \mathbf{1}$.

- Since $\mathbf{P}\mathbf{1} = \mathbf{P}^T\mathbf{1} = \mathbf{1}$ with binary entries $\Rightarrow$ Exactly one nonzero entry per row and column of $\mathbf{P}$
- Permutation matrices are unitary $\Rightarrow \mathbf{P}^T\mathbf{P} = \mathbf{I}$. Matrix $\mathbf{P}^T$ undoes the reordering of matrix $\mathbf{P}$

- If $(\mathbf{S}, \mathbf{x})$ is a graph signal, $(\mathbf{P}^T\mathbf{S}\mathbf{P}, \mathbf{P}^T\mathbf{x})$ is a relabeling of $(\mathbf{S}, \mathbf{x})$. Same signal. Different names

Graph signal $\mathbf{x}$ Supported on $\mathbf{S}$ | Graph signal $\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x}$ supported on $\hat{\mathbf{S}} = \mathbf{P}^T\mathbf{S}\mathbf{P}$



- Processing should be label-independent $\Rightarrow$ Permutation equivariance of graph filters and GNNs

- Graph filter $\mathbf{H}(\mathbf{S})$ is a polynomial on shift operator $\mathbf{S}$ with coefficients h_k. Outputs given by

$$\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$$

- We consider running the same filter on $(\mathbf{S}, \mathbf{x})$ and permuted (relabeled) $(\hat{\mathbf{S}}, \hat{\mathbf{x}}) = (\mathbf{P}^T\mathbf{S}\mathbf{P}, \mathbf{P}^T\mathbf{x})$

$$\mathbf{H}(\mathbf{S})\mathbf{x} \;=\; \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x} \qquad\qquad \mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} \;=\; \sum_{k=0}^{K-1} h_k \hat{\mathbf{S}}^k \hat{\mathbf{x}}$$

- Filter $\mathbf{H}(\mathbf{S})\mathbf{x} \Rightarrow$ Coefficients h_k. Input signal $\mathbf{x}$. Instantiated on shift $\mathbf{S}$
- Filter $\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} \Rightarrow$ Same Coefficients h_k. Permuted Input signal $\hat{\mathbf{x}}$. Instantiated on permuted shift $\hat{\mathbf{S}}$

- L layers recursively process outputs of previous layers. GNN Output parametrized by tensor $\mathcal{H}$

$$\mathbf{x}_\ell = \sigma\left[\sum_{k=0}^{K-1} h_{\ell k}\, \mathbf{S}^k\, \mathbf{x}_{\ell-1}\right] = \sigma\Big[\mathbf{H}_\ell(\mathbf{S})\, \mathbf{x}_{\ell-1}\Big] \qquad \Phi\Big(\mathbf{x};\ \mathbf{S},\ \mathcal{H}\Big) = \mathbf{x}_L$$

- We consider running the same GNN on $(\mathbf{S}, \mathbf{x})$ and permuted (relabeled) $(\hat{\mathbf{S}}, \hat{\mathbf{x}}) = (\mathbf{P}^T\mathbf{S}\mathbf{P}, \mathbf{P}^T\mathbf{x})$

$$\Phi\Big(\mathbf{x};\ \mathbf{S},\ \mathcal{H}\Big) \qquad\qquad \Phi\Big(\hat{\mathbf{x}};\ \hat{\mathbf{S}},\ \mathcal{H}\Big)$$

- GNN $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \Rightarrow$ Tensor $\mathcal{H}$. Input signal $\mathbf{x}$. Instantiated on shift $\mathbf{S}$
- GNN $\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) \Rightarrow$ Same Tensor $\mathcal{H}$. Permuted Input signal $\hat{\mathbf{x}}$. Instantiated on permuted shift $\hat{\mathbf{S}}$

Theorem (Permutation equivariance of graph neural networks)

Consider consistent permutations of the shift operator $\hat{\mathbf{S}} = \mathbf{P}^T\mathbf{S}\mathbf{P}$ and input signal $\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x}$. Then

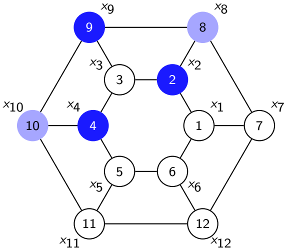
$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) \;=\; \mathbf{P}^T\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

- GNNs equivariant to permutations $\Rightarrow$ Permute input and shift $\equiv$ Permute output

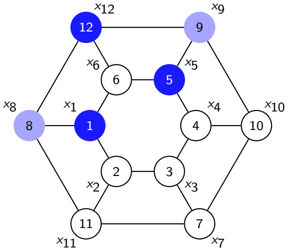
- We requested signal processing independent of labeling $\Rightarrow$ GNNs fulfill this request

$\Rightarrow$ Permute input and shift $\equiv$ Relabel input $\Rightarrow$ Permute output $\equiv$ Relabel output

Graph signal $\mathbf{x}$ Supported on $\mathbf{S}$



Graph signal $\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x}$ supported on $\hat{\mathbf{S}} = \mathbf{P}^T\mathbf{S}\mathbf{P}$



- We requested signal processing independent of labeling $\Rightarrow$ GNNs fulfill this request

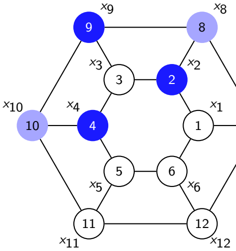
$\Rightarrow$ Permute input and shift $\equiv$ Relabel input $\Rightarrow$ Permute output $\equiv$ Relabel output

GNN output $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ supported on $\mathbf{S}$

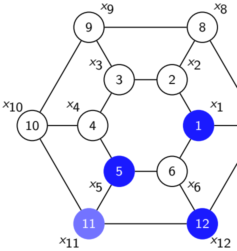
GNN $\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^T\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ on $\hat{\mathbf{S}} = \mathbf{P}^T\mathbf{S}\mathbf{P}$

- Equivariance to permutations allows GNNs to exploit symmetries of graphs and graph signals
- By symmetry we mean that the graph can be permuted onto itself $\Rightarrow \mathbf{S} = \mathbf{P}^T\mathbf{S}\mathbf{P}$
- Equivariance theorem implies $\Rightarrow \Phi\Big(\mathbf{P}^T\mathbf{x};\, \mathbf{S}, \mathcal{H}\Big) = \Phi\Big(\mathbf{P}^T\mathbf{x};\, \mathbf{P}^T\mathbf{S}\mathbf{P}, \mathcal{H}\Big) = \mathbf{P}^T\Phi\Big(\mathbf{x};\, \mathbf{S}, \mathcal{H}\Big)$

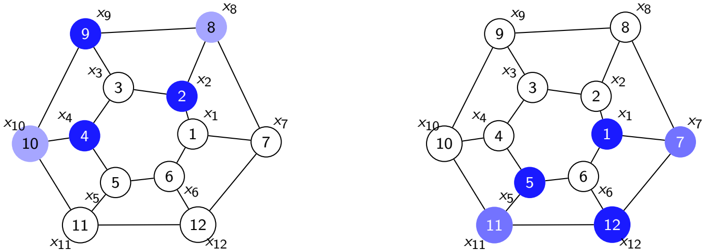
From observing $\mathbf{x}$ supported on $\mathbf{S}$



Learn to process $\mathbf{P}^T\mathbf{x}$ supported on $\mathbf{S} = \mathbf{P}^T\mathbf{S}\mathbf{P}$



- Graph not symmetric but close to symmetric $\Rightarrow$ perturbed version of a permutation of itself



- We will show conditions for stability to deformations $\Rightarrow$ Approximate (close to) equivariance

Definition (Operator Distance Modulo Permutation)

For operators Ψ and $\hat{\Psi}$, the operator distance modulo permutation is defined as

$$\left\|\Psi - \hat{\Psi}\right\|_{\mathcal{P}} = \min_{\mathbf{P}\in\mathcal{P}} \max_{\mathbf{x}:\|\mathbf{x}\|=1} \left\| \mathbf{P}^T\Psi(\mathbf{x}) - \hat{\Psi}(\mathbf{P}^T\mathbf{x}) \right\|$$

where $\mathcal{P}$ is the set of $n \times n$ permutation matrices and where $\|\cdot\|$ stands for the ℓ_2-norm.

- Equivariance to permutations of graph filters $\Rightarrow$ If $\left\|\hat{\mathbf{S}} - \mathbf{S}\right\|_{\mathcal{P}} = \mathbf{0}$. Then $\left\|\mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S})\right\|_{\mathcal{P}} = \mathbf{0}$
- Equivariance to permutations GNNs $\Rightarrow$ If $\left\|\hat{\mathbf{S}} - \mathbf{S}\right\|_{\mathcal{P}} = \mathbf{0}$. Then $\left\|\Phi(\cdot;\hat{\mathbf{S}},\mathcal{H}) - \Phi(\cdot;\mathbf{S},\mathcal{H})\right\|_{\mathcal{P}} = \mathbf{0}$
- When distance $\left\|\hat{\mathbf{S}} - \mathbf{S}\right\|_{\mathcal{P}}$ is small? (not zero) $\Rightarrow$ Stability properties of graph filters and GNNs

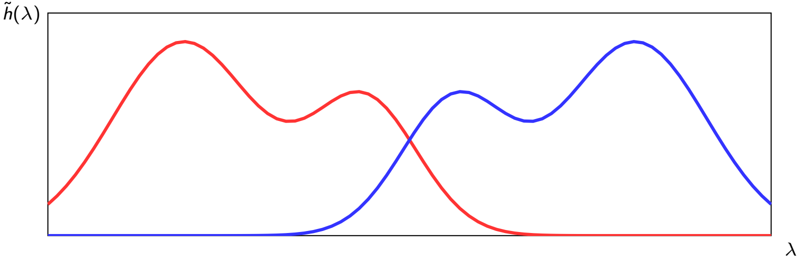
Lipschitz and Integral Lipschitz Filters

- Classes of filters to study discriminablity of GNNs $\Rightarrow$ Lipschitz and integral Lipschitz graph filters

- Graph filters are polynomials on shift operators $\mathbf{S}$ with given coefficients h_k $\Rightarrow \mathbf{H}(\mathbf{S}) = \sum_{k=0}^{\infty} h_k \mathbf{S}^k$

- Filter's frequency response is the same polynomial with scalar variable λ $\Rightarrow \tilde{h}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$

- Frequency response determined by filter coefficients h_k. Independent of particular given graph



Definition (Lipschitz Filter)

Given a graph filter with coefficients $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$, and graph frequency response

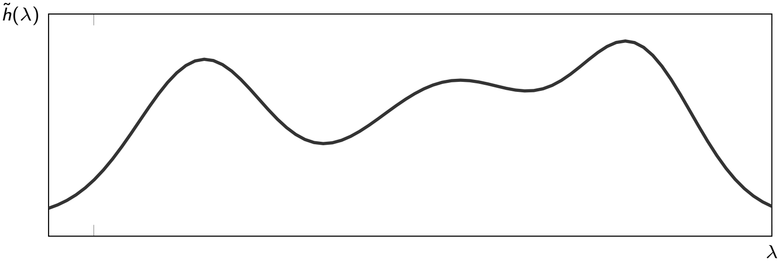
$$\tilde{h}(\lambda) \ = \ \sum_{k=0}^{\infty} h_k \lambda^k,$$

we say that the filter is Lipschitz if there exists a constant $C > 0$ such that for λ_1 and λ_2

$$\left| \tilde{h}(\lambda_2) - \tilde{h}(\lambda_1) \right| \ \leq \ C \left| \lambda_2 - \lambda_1 \right|.$$

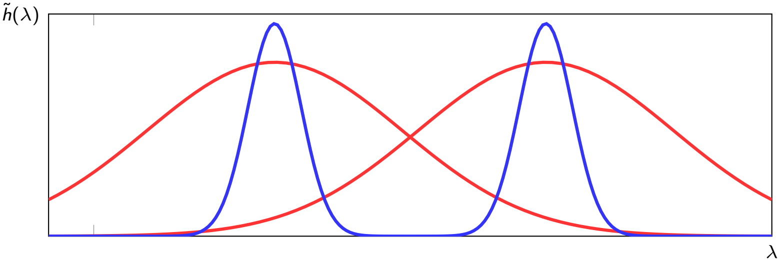
- Change in values of frequency response is at most linear with rate C $\Rightarrow$ Derivative $\tilde{h}'(\lambda) \leq C$

- Frequency response $\tilde{h}(\lambda)$ of Lipschitz filter is Lipschitz continuous $\Rightarrow$ Maximum slope is $\tilde{h}'(\lambda) \leq C$



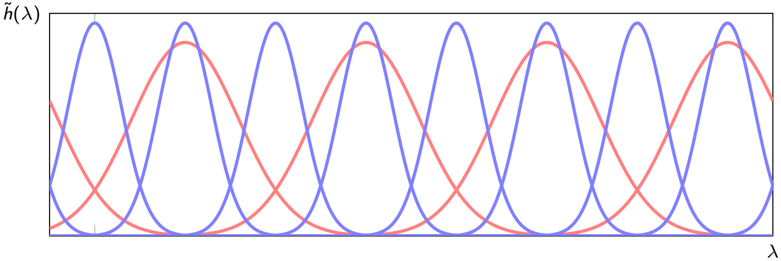
- Lipschitz constant determines discriminability $\Rightarrow$ Small / Large C $\equiv$ Low / High discriminability

- Frequency response $\tilde{h}(\lambda)$ of Lipschitz filter is Lipschitz continuous $\Rightarrow$ Maximum slope is $\tilde{h}'(\lambda) \leq C$



- Lipschitz constant determines discriminability $\Rightarrow$ Small / Large C $\equiv$ Low / High discriminability

- A Lipschitz frame with constant C is made up of Lipschitz filters with constant C
- Larger C allows for sharper filters, that can discriminate more signals. Tighter packing
- The discriminability of the frame is (or can be) the same at all frequencies.



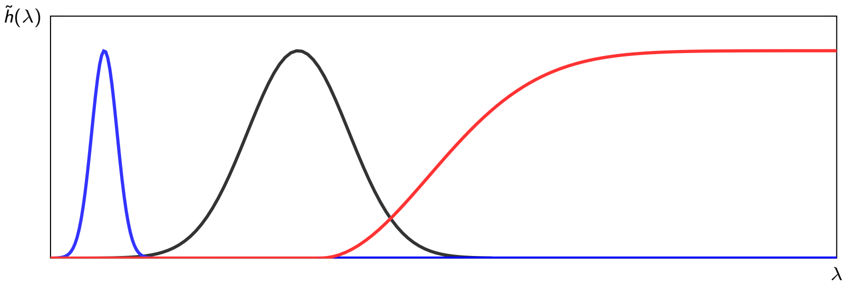
Definition (Integral Lipschitz Filter)

Consider graph filter with coefficients h_k and graph frequency response $\tilde{h}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$. The filter is said integral Lipschitz if there exists constant $C > 0$ such that for all λ_1 and λ_2,

$$\left| \tilde{h}(\lambda_2) - \tilde{h}(\lambda_1) \right| \leq C \frac{\left| \lambda_2 - \lambda_1 \right|}{\left| \lambda_1 + \lambda_2 \right| / 2}.$$

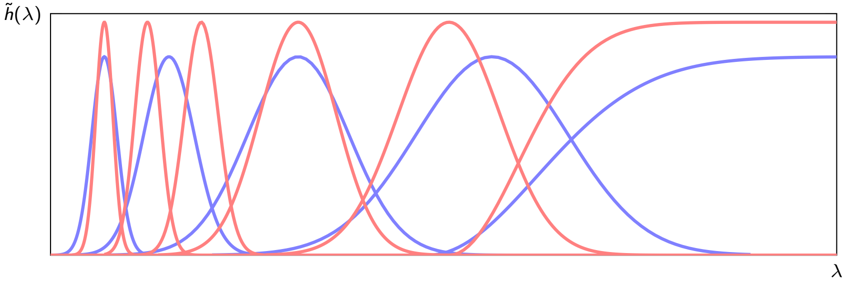
- Lipschitz with a constant that is inversely proportional to the interval's midpoint $\Rightarrow 2C/|\lambda_1 + \lambda_2|$.
- Letting $\lambda_2 \to \lambda_1$ we get that $\lambda \tilde{h}'(\lambda) \leq C$ $\Rightarrow$ The filter can't change for large λ.

- At medium frequencies, integral Lipschitz filters are akin to Lipschitz filters. Roughly speaking
- At low frequencies integral Lipschitz filters can be arbitrarily thin $\Rightarrow$ arbitrary discriminability
- At high frequencies integral Lipschitz filters have to be flat $\Rightarrow$ They lose discriminability



Integral Lipschitz Frames

- As Lipschitz frames, integral Lipschitz frames are more discriminative for larger C. Tighter packing
- Except that around $\lambda = 0$, filters can be thin no matter C $\Rightarrow$ High discriminability
- But for large λ filters have to be wide no matter C $\Rightarrow$ No discriminability



Additive Perturbations of Graph Filters

- We define additive perturbations of the graph support

A Graph Filter can be Perturbed in Three Ways but Only One is Interesting

- Graph filter $\mathbf{H}(\mathbf{S})$ is a polynomial on shift operator $\mathbf{S}$ with coefficients h_k. Outputs given by

$$\mathbf{H}(\mathbf{S})\,\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$$

- Perturbations of the input $\Rightarrow$ The filter is linear in $\mathbf{x}$. Scale error by filter's norm.

- Perturbations of the coefficients $\Rightarrow$ Filter is linear in h_k. Plus, h_k is a design parameter.

- Perturbations of the shift operator $\mathbf{S}$ $\Rightarrow$ It is not easy (nonlinear). And it is necessary.

  $\Rightarrow$ The graph is estimated (recommendation systems). The graph changes (distributed systems)

  $\Rightarrow$ Quasi-symmetries in graphs that are quasi-invariant to permutations

- Apply the same filter $\mathbf{h}$ to the same signal $\mathbf{x}$ on different graphs shift operators $\mathbf{S}$ and $\hat{\mathbf{S}}$

$$\mathbf{H}(\mathbf{S})\,\mathbf{x} \;=\; \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x} \qquad\qquad \mathbf{H}(\hat{\mathbf{S}})\,\mathbf{x} \;=\; \sum_{k=0}^{K-1} h_k \hat{\mathbf{S}}^k \mathbf{x}$$

- Filter $\mathbf{H}(\mathbf{S})\,\mathbf{x}$ $\Rightarrow$ Coefficients h_k. Input signal $\mathbf{x}$. Instantiated on shift $\mathbf{S}$
- Filter $\mathbf{H}(\hat{\mathbf{S}})\,\hat{\mathbf{x}}$ $\Rightarrow$ Same Coefficients h_k. Same Input signal $\mathbf{x}$. Instantiated on perturbed shift $\hat{\mathbf{S}}$

- We will investigate two commonly encountered graph perturbation models.

- Additive perturbation model $\Rightarrow \hat{\mathbf{S}} = \mathbf{S} + \mathbf{E}$ $\Rightarrow$ Allows us to study deformations that are independent of the graph structure.
- Error matrix $\mathbf{E} = \hat{\mathbf{S}} - \mathbf{S}$ exists for any pair $\mathbf{S}$, $\hat{\mathbf{S}}$. $\Rightarrow$ It's norm $\|\mathbf{E}\|$ quantifies their difference
- A flaw $\Rightarrow$ Graphs $\mathbf{S}$ and $\hat{\mathbf{S}} = \mathbf{P}^T\mathbf{S}\mathbf{P}$ are the same (relabeling). Yet we may not have $\|\mathbf{E}\| = 0$.
- We know better $\Rightarrow$ Operator distances modulo permutation $\big\| \hat{\mathbf{S}} - \mathbf{S} \big\|_{\mathcal{P}} = \min_{\mathcal{P}} \big\| \hat{\mathbf{S}}\mathbf{P}^T - \mathbf{P}^T\mathbf{S} \big\|$

- We need a concrete handle on the error matrix. Start from set of symmetric error matrices

$$\mathcal{E}(\mathbf{S}, \hat{\mathbf{S}}) = \left\{ \tilde{\mathbf{E}} : \mathbf{P}^T \hat{\mathbf{S}} \mathbf{P} = \mathbf{S} + \tilde{\mathbf{E}}, \quad \mathbf{P} \in \mathcal{P} \right\}$$

- For each permutation $\mathbf{P} \in \mathcal{P}$ we have a different error matrix $\tilde{\mathbf{E}} = \mathbf{P}^T \hat{\mathbf{S}} \mathbf{P} - \mathbf{S}$ in the set $\mathcal{E}(\mathbf{S}, \hat{\mathbf{S}})$

- Error matrix modulo permutation is the one with smallest norm $\Rightarrow \mathbf{E} = \underset{\tilde{\mathbf{E}} \in \mathcal{E}(\mathbf{S}, \hat{\mathbf{S}})}{\text{argmin}} \|\tilde{\mathbf{E}}\|$

- Rewrite the distance modulo permutation as $\Rightarrow d(\mathbf{S}, \hat{\mathbf{S}}) = \|\mathbf{E}\| = \min_{\tilde{\mathbf{E}} \in \mathcal{E}(\mathbf{S}, \hat{\mathbf{S}})} \|\tilde{\mathbf{E}}\|$

- Error norm $\|\mathbf{E}\| = d(\mathbf{S}, \hat{\mathbf{S}})$ measures how far $\mathbf{S}$ and $\hat{\mathbf{S}}$ are from being permutations of each other

Eigenvector Misalignment Constant

- Consider eigenvector decompositions of the shift $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^H$ and the error $\mathbf{E} = \mathbf{U}\mathbf{M}\mathbf{U}^H$

- Define the eigenvector misalignment between the shift operator **S** and the error matrix **E** as

$$\delta \;=\; \Big(\|\mathbf{U}-\mathbf{V}\|+1\Big)^2 - 1$$

- Since **U** and **V** are unitary matrices $\|\mathbf{U}\| = \|\mathbf{V}\| = 1 \;\Rightarrow \delta \leq 8 \;=\; [(2+1)^2 - 1]$

$\Rightarrow$ The eigenvector misalignment δ is never large. It can be small. Depending on the error model.

Stability of Lipschitz Filters to Additive Perturbations

- We show that Lipschitz filters are stable to additive perturbations of the graph support.

Theorem (Lipschitz Filters are Stable to Additive Perturbations)

Consider graph filter **h** along with shift operators $\mathbf{S}$ and $\hat{\mathbf{S}}$ having n nodes. If it holds that:

(H1) Shift operators $\mathbf{S}$ and $\hat{\mathbf{S}}$ are related by $\mathbf{P}^T\hat{\mathbf{S}}\mathbf{P} = \mathbf{S} + \mathbf{E}$ with $\mathbf{P}$ a permutation matrix

(H2) The error matrix $\mathbf{E}$ has norm $\|\mathbf{E}\| = \epsilon$ and eigenvector misalignement δ relative to $\mathbf{S}$

(H3) The filter **h** is Lipschitz with constant C

Then, the operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ is bounded by

$$\left\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \right\|_{\mathcal{P}} \leq C \left(1 + \delta\sqrt{n}\right) \epsilon + \mathcal{O}(\epsilon^2).$$

Theorem (Lipschitz Filters are Stable to Additive Perturbations)

The operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ satisfies

$$\left\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \right\|_{\mathcal{P}} \leq C \left(1 + \delta\sqrt{n}\right) \epsilon + \mathcal{O}(\epsilon^2).$$

- If shifts $\mathbf{S}$ and $\hat{\mathbf{S}}$ are ϵ-close the filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ are ϵ-close. Modulo permutation
- Proportional to the Lipschitz constant of the filter's frequency response. Not integral Lipschitz
- Proportional to $(1 + \delta\sqrt{n})$. Not great for large graphs. Unless misalignement decreases with n.
- Growth with n is at most $(1 + 8\sqrt{n}) \geq (1 + \delta\sqrt{n})$. Because $\delta \leq 8$. Not that bad

Theorem (Lipschitz Filters are Stable to Additive Perturbations)

The operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ satisfies

$$\left\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \right\|_{\mathcal{P}} \leq C \left(1 + \delta\sqrt{n} \right) \epsilon + \mathcal{O}(\epsilon^2).$$

- Filter perturbations are first order Lipschitz continuous with respect to the perturbation's size ϵ

$\Rightarrow$ With Lipschitz constant $\Rightarrow C \left(1 + \delta\sqrt{n} \right)$

- Stronger than plain continuity. Which would say "output changes are small if input changes are"

Theorem (Lipschitz Filters are Stable to Additive Perturbations)

The operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ is bounded by

$$\left\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \right\|_{\mathcal{P}} \leq C \left(1 + \delta\sqrt{n}\right) \epsilon + \mathcal{O}(\epsilon^2).$$

- Bound is universal for all graphs with a given number of nodes n. Bound depends on:
  - ⇒ A property of the filter's frequency response. The filter's Lipschitz constant C
  - ⇒ And properties of the perturbation $\mathbf{E}$. The eigenvector misalignement δ and the norm $\|\mathbf{E}\| = \epsilon$
- There is no constant in the bound that depends on the graph shift operator $\mathbf{S}$. Save for n.

Theorem (Lipschitz Filters are Stable to Additive Perturbations)

The operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ satisfies

$$\left\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \right\|_{\mathcal{P}} \leq C \left(1 + \delta\sqrt{n} \right) \epsilon + \mathcal{O}(\epsilon^2).$$

- The filter's Lipschitz constant C is a parameter that we can affect with judicious filter choice
- Discriminability / stability tradeoff. Larger C improves discriminability at the cost of stability

Theorem (Lipschitz Filters are Stable to Additive Perturbations)

The operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ is bounded by

$$\big\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \big\|_{\mathcal{P}} \leq C \left(1 + \delta\sqrt{n}\right) \epsilon + \mathcal{O}(\epsilon^2).$$

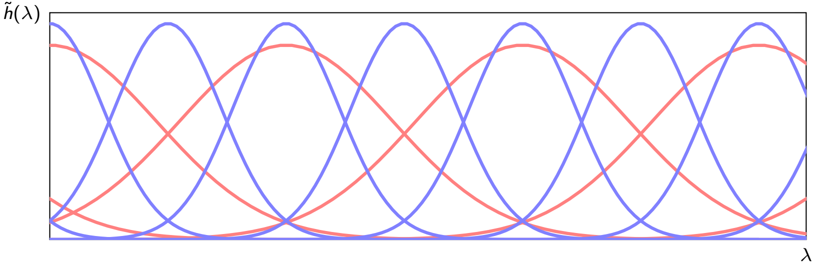
- Eigenvector misalignment δ is a property of the perturbation matrix. Independent of filter choice

  $\Rightarrow$ Not very relevant in studying stability / discriminability tradeoffs of different filters.

- Meaningless asymptotically on n. Don't know much about perturbations in the limit of large n

Lipschitz Filters are Good News

- Stability to additive perturbations requires Lipschitz filters. Not integral Lipschitz as with scalings
- Genuine stability / discriminability tradeoff $\Rightarrow$ Larger C tradeoffs stability for discriminability
- We can always discriminate, regardless of frequency, if we tolerate enough discriminability.



Relative Perturbations of Graph Filters

- Proved enticing stability properties with respect to additive perturbations. Alas, not ideal

- We switch focus to relative perturbations. Which tie perturbations to the graph structure

- Additive perturbations are not ideal

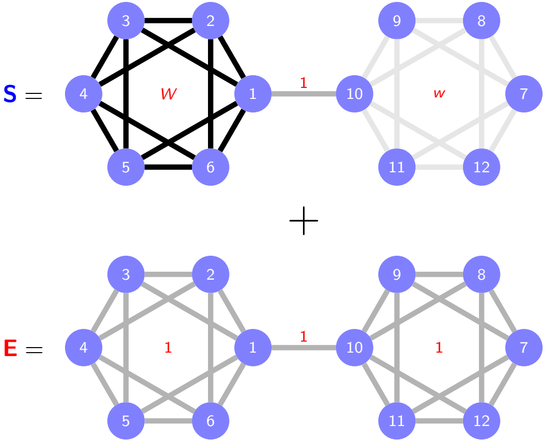
$$\mathbf{P}^T\hat{\mathbf{S}}\mathbf{P} = \mathbf{S} + \mathbf{E}$$

- With $w \ll 1 \ll W$.

  $\Rightarrow$ Is this perturbation small or large?

- Edges with small weights w can change a lot

  because other edges have large weights W



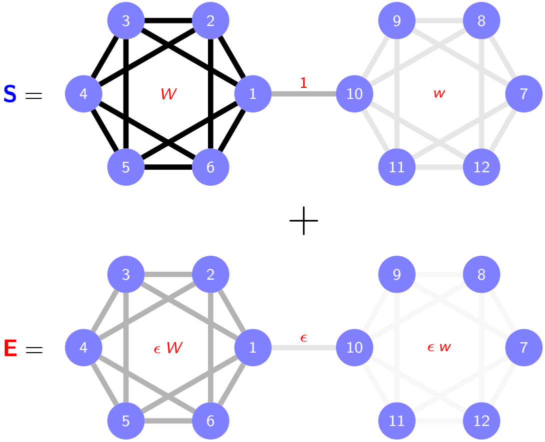
- Relative perturbations are more meaningful

$$\mathbf{P}^T\hat{\mathbf{S}}\mathbf{P} = \mathbf{S} + \mathbf{E} = \mathbf{S} + \epsilon\mathbf{I}\mathbf{S}$$

- With $w \ll 1 \ll W$ and $\epsilon \ll 1$

$\Rightarrow$ Is this perturbation small or large?

- It's small. Edges with small weights change little. Edges with large weights change more



- Relative perturbation model $\Rightarrow \hat{\mathbf{S}} = \mathbf{S} + \mathbf{ES} + \mathbf{SE}$. We must account for permutations (relabeling)

- Set of relative error matrices modulo permutation. Matrices $\tilde{\mathbf{E}}$ are symmetric, $\tilde{\mathbf{E}} = \tilde{\mathbf{E}}^T$

$$\mathcal{E}(\mathbf{S}, \hat{\mathbf{S}}) = \left\{ \tilde{\mathbf{E}} \; : \; \mathbf{P}^T \hat{\mathbf{S}} \mathbf{P} \; = \; \mathbf{S} \; + \; \tilde{\mathbf{E}}\mathbf{S} \; + \; \mathbf{S}\tilde{\mathbf{E}} \; , \;\; \mathbf{P} \in \mathcal{P} \right\}$$

- Relative error matrix modulo permutation is the one with smallest norm $\Rightarrow \mathbf{E} \; = \; \underset{\tilde{\mathbf{E}} \in \mathcal{E}(\mathbf{S}, \hat{\mathbf{S}})}{\operatorname{argmin}} \|\tilde{\mathbf{E}}\|$

- Define relative distance modulo permutation as $\Rightarrow d(\mathbf{S}, \hat{\mathbf{S}}) \; = \; \|\mathbf{E}\| \; = \; \min_{\tilde{\mathbf{E}} \in \mathcal{E}(\mathbf{S}, \hat{\mathbf{S}})} \|\tilde{\mathbf{E}}\|$

- Norm $\|\mathbf{E}\| = d(\mathbf{S}, \hat{\mathbf{S}})$ is a relative measure of how far $\hat{\mathbf{S}}$ is from being a permutation of $\mathbf{S}$

Relative Perturbations are Tied to the Local Structure of the Graph

- Relative perturbations tie changes in the edge weights to the local structure of the graph
- Compare edge weights in the given matrix $\mathbf{S}$ and the permuted version of the perturbations $\hat{\mathbf{S}}$

$$\left(\mathbf{P}^T\hat{\mathbf{S}}\mathbf{P}\right)_{ij} = S_{ij} + \left(\mathbf{ES}\right)_{ij} + \left(\mathbf{SE}\right)_{ij}$$
$$= S_{ij} + \sum_{k\in n(j)} E_{ik}S_{kj} + \sum_{k\in n(i)} S_{ik}E_{kj}$$

- Edge changes are proportional to the degree of the incident nodes. Scaled by entries of error matrix
- Parts of the graph with weaker connectivity see smaller changes than parts with stronger links
- In generic additive perturbations weights can change the same regardless of local connectivity

Stability of Integral Lipschitz Filters to Relative Perturbations

- We show that integral Lipschitz filters are stable to relative perturbations of the graph support.

Theorem (Integral Lipschitz Filters are Stable to Relative Perturbations)

Consider graph filter **h** along with shift operators $\mathbf{S}$ and $\hat{\mathbf{S}}$ having n nodes. If it holds that:

(H1) $\mathbf{S}$ and $\hat{\mathbf{S}}$ are related by $\mathbf{P}^T\hat{\mathbf{S}}\mathbf{P} = \mathbf{S} + \mathbf{ES} + \mathbf{SE}$ with $\mathbf{P}$ a permutation matrix

(H2) Error matrix has norm $\|\mathbf{E}\| = \epsilon$ and eigenvector misalignment constant δ relative to $\mathbf{S}$

(H3) The filter is integral Lipschitz with constant C

Then, the operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ is bounded by

$$\left\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \right\|_{\mathcal{P}} \leq 2C\left(1 + \delta\sqrt{n}\right)\epsilon + \mathcal{O}(\epsilon^2).$$

Theorem (Integral Lipschitz Filters are Stable to Relative Perturbations)

The operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ is bounded by

$$\left\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \right\|_{\mathcal{P}} \leq 2C\left(1 + \delta\sqrt{n}\right)\epsilon + \mathcal{O}(\epsilon^2).$$

- Save for the 2 factor, it is the same bound we have for the case of additive perturbations.
- The difference is in hypotheses (H1) and (H3). Hypothesis (H2) does not change

(H1) The perturbation is relative. $\Rightarrow \mathbf{P}^T\hat{\mathbf{S}}\mathbf{P} = \mathbf{S} + \mathbf{E}\mathbf{S} + \mathbf{S}\mathbf{E}$. Not additive.

(H3) The filter is integral Lipschitz with constant C. Not regular Lipschitz.

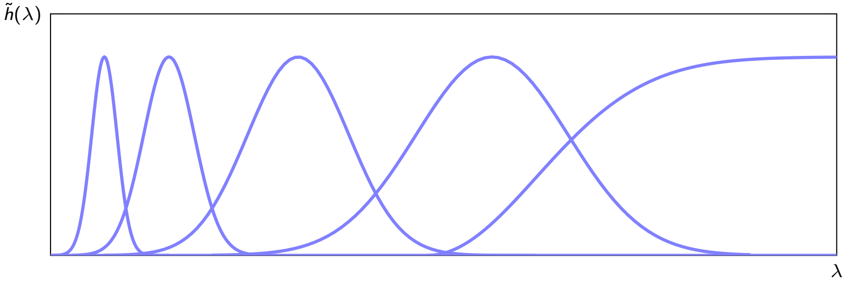
Theorem (Integral Lipschitz Filters are Stable to Relative Perturbations)

The operator distance modulo permutation between filters $\mathbf{H}(\mathbf{S})$ and $\mathbf{H}(\hat{\mathbf{S}})$ is bounded by

$$\left\| \mathbf{H}(\hat{\mathbf{S}}) - \mathbf{H}(\mathbf{S}) \right\|_{\mathcal{P}} \leq 2C \left(1 + \delta\sqrt{n}\right) \epsilon + \mathcal{O}(\epsilon^2).$$

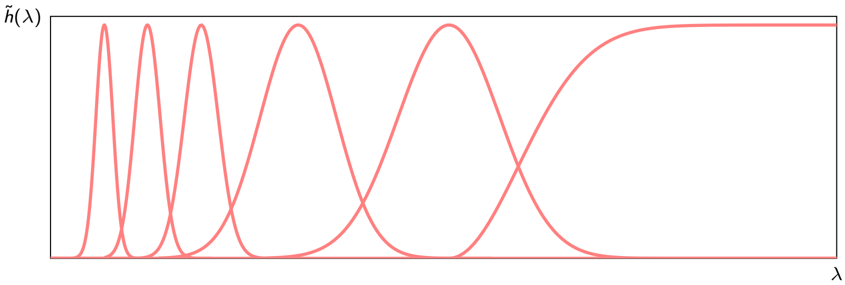
- Bound depends on integral Lipschitz constant C. Very different from Lipschitz constant
- Can decrease C to increase stability. But effect on Discriminability depends on the frequency.
  - $\Rightarrow$ Discriminative at low frequencies regardless of C
  - $\Rightarrow$ Non-discriminative at high frequencies regardless of C

- Integral Lipschitz filters are necessary for stability to deformations of the supporting graph
- This is not an artifact of the analysis. The result is tight. The term $\sum_{k=0}^{\infty} k\, h_k\, \lambda_i^k = \lambda_i\, h'(\lambda_i)$ appears.



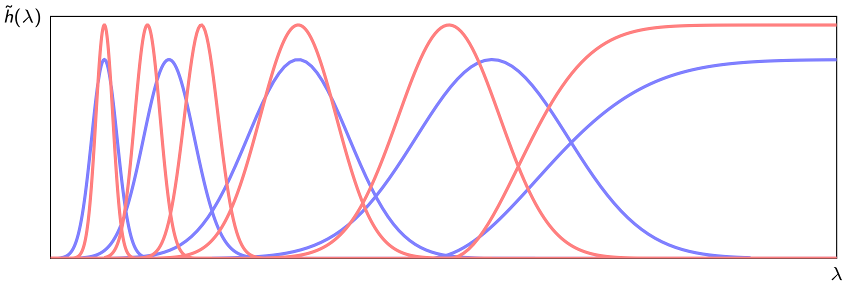
Integral Lipschitz Filters are Not Good News

- One would expect a stability vs discriminability tradeoff. But in a sense, we get a non-tradeoff.
- Integral Lipschitz filters have to be flat at high frequencies. $\Rightarrow$ They can't discriminate
- It is impossible to separate signals with high frequency features and be stable to deformations



Integral Lipschitz Filters are Not Good News

- One would expect a stability vs discriminability tradeoff. But in a sense, we get a non-tradeoff.
- Integral Lipschitz filters have to be flat at high frequencies. $\Rightarrow$ They can't discriminate
- It is impossible to separate signals with high frequency features and be stable to deformations



Stability Properties of Graph Neural Networks

- The stability properties we studied for graph filters are inherited by GNNs

GNNs Inherit Any Stability Properties that Filters May Have

- Lipschitz filters are stable to additive deformations of the shift operator

  ⇒ GNNs with Lipschitz layers are stable to additive deformations of the shift operator

- Integral Lipschitz filters are stable to relative deformations of the shift operator

  ⇒ GNNs with integral Lipschitz layers are stable to relative deformations of the shift operator

Normalizations

- At each layer of the GNN, the filters have unit operator norm $\Rightarrow \| \mathbf{H}_\ell(\mathbf{S}) \| = 1$

  $\Rightarrow$ Easy to achieve with scaling $\Rightarrow$ Equivalent to $\max_\lambda \tilde{h}_\ell(\lambda) = 1$

- The nonlinearity σ is Lipschitz and normalized so that $\Rightarrow \| \sigma(\mathbf{x}_2) - \sigma(\mathbf{x}_1) \| \leq \| \mathbf{x}_2 - \mathbf{x}_1 \|$

  $\Rightarrow$ Easy to achieve with scaling. True of ReLU, hyperbolic tangent, and absolute value

- Joining both assumptions $\Rightarrow$ If input energy is $\|\mathbf{x}\| \leq 1$, all layer outputs have energy $\|\mathbf{x}_\ell\| \leq 1$

Theorem (GNN Stability to Additive Perturbations)

Consider a GNN operator $\mathbf{\Phi}(\cdot; \mathbf{S}, \mathcal{H})$ along with shifts operators $\mathbf{S}$ and $\hat{\mathbf{S}}$ having n nodes. If:

- **(H1)** Shift operators are related by $\mathbf{P}^T \hat{\mathbf{S}} \mathbf{P} = \mathbf{S} + \mathbf{E}$. With $\mathbf{P}$ a permutation matrix
- **(H2)** The error matrix $\mathbf{E}$ has norm $\|\mathbf{E}\| = \epsilon$ and eigenvector misalignement δ relative to $\mathbf{S}$
- **(H3)** The GNN has L single-feature layers with Lipschitz filters with constant C
- **(H4)** Filters have unit operator norm and the nonlinearity is normalized Lipschitz

Then, the operator distance modulo permutation between $\mathbf{\Phi}(\cdot; \mathbf{S}, \mathcal{H})$ and $\mathbf{\Phi}(\cdot; \hat{\mathbf{S}}, \mathcal{H})$ is bounded by

$$\left\| \mathbf{\Phi}(\cdot; \hat{\mathbf{S}}, \mathcal{H}) - \mathbf{\Phi}(\cdot; \mathbf{S}, \mathcal{H}) \right\|_{\mathcal{P}} \leq C \left(1 + \delta\sqrt{n}\right) L\epsilon + \mathcal{O}(\epsilon^2).$$

Theorem (GNN Stability to Additive Perturbations)

The operator distance modulo permutation between $\mathbf{\Phi}(\cdot; \mathbf{S}, \mathcal{H})$ and $\mathbf{\Phi}(\cdot; \hat{\mathbf{S}}, \mathcal{H})$ is bounded by

$$\left\| \mathbf{\Phi}(\cdot; \hat{\mathbf{S}}, \mathbf{h}) - \mathbf{\Phi}(\cdot; \mathbf{S}, \mathbf{h}) \right\|_{\mathcal{P}} \leq C \left(1 + \delta\sqrt{n}\right) L \epsilon + \mathcal{O}(\epsilon^2).$$

- It is essentially the same bound we have for the case of Lipschitz filters. Propagated over L layers
- A GNN with Lipschitz layers inherits the stability of the Lipschitz filter class
- The nonlinearity is pointwise $\Rightarrow$ Graph deformations have no effect on its action

Theorem (GNN Stability to Relative Perturbations)

Consider a GNN operator $\mathbf{\Phi}(\cdot;\mathbf{S},\mathcal{H})$ along with shifts operators $\mathbf{S}$ and $\hat{\mathbf{S}}$ having n nodes. If:

- **(H1)** Shift operators are related by $\mathbf{P}^T\hat{\mathbf{S}}\mathbf{P} = \mathbf{S}+\mathbf{ES}+\mathbf{SE}$ with $\mathbf{P}$ a permutation matrix
- **(H2)** The error matrix $\mathbf{E}$ has norm $\|\mathbf{E}\| = \epsilon$ and eigenvector misalignement δ relative to $\mathbf{S}$
- **(H3)** The GNN has L single-feature layers with integral Lipschitz filters with constant C
- **(H4)** Filters have unit operator norm and the nonlinearity is normalized Lipschitz

Then, the operator distance modulo permutation between $\mathbf{\Phi}(\cdot;\mathbf{S},\mathcal{H})$ and $\mathbf{\Phi}(\cdot;\hat{\mathbf{S}},\mathcal{H})$ is bounded by

$$\left\| \mathbf{\Phi}(\cdot;\hat{\mathbf{S}},\mathcal{H}) - \mathbf{\Phi}(\cdot;\mathbf{S},\mathcal{H}) \right\|_{\mathcal{P}} \leq 2C\left(1+\delta\sqrt{n}\right)L\epsilon + \mathcal{O}(\epsilon^2).$$

Theorem (Single Feature GNN Stability to Relative Perturbations)

The operator distance modulo permutation between $\mathbf{\Phi}(\cdot; \mathbf{S}, \mathcal{H})$ and $\mathbf{\Phi}(\cdot; \hat{\mathbf{S}}, \mathcal{H})$ is bounded by

$$\left\| \mathbf{\Phi}(\cdot; \hat{\mathbf{S}}, \mathbf{h}) - \mathbf{\Phi}(\cdot; \mathbf{S}, \mathbf{h}) \right\|_{\mathcal{P}} \leq 2C \left(1 + \delta\sqrt{n}\right) L\epsilon + \mathcal{O}(\epsilon^2).$$

- It is essentially the same bound we have for integral Lipschitz filters. Propagated over L layers
- A GNN with integral Lipschitz layers inherits the stability of integral Lipschitz filters
- The nonlinearity is pointwise $\Rightarrow$ Graph deformations have no effect on its action

GNNs Inherit the Stability Properties of Graph Filters

- Provide a generic inheritance proof $\Rightarrow$ the steps apply to any stability claim on any filter class.

- Let's do the proof for relative perturbations and integral Lipschitz filters.

Theorem (GNN Stability to Relative Perturbations)

Consider a GNN operator $\mathbf{\Phi}(\cdot; \mathbf{S}, \mathcal{H})$ along with shifts operators $\mathbf{S}$ and $\hat{\mathbf{S}}$ having n nodes. If:

- **(H1)** Shift operators are related by $\mathbf{P}^T\hat{\mathbf{S}}\mathbf{P} = \mathbf{S} + \mathbf{E}\mathbf{S} + \mathbf{S}\mathbf{E}$ with $\mathbf{P}$ a permutation matrix
- **(H2)** The error matrix $\mathbf{E}$ has norm $\|\mathbf{E}\| = \epsilon$ and eigenvector misalignement δ relative to $\mathbf{S}$
- **(H3)** The GNN has L single-feature layers with integral Lipschitz filters with constant C
- **(H4)** Filters have unit operator norm and the nonlinearity is normalized Lipschitz

Then, the operator distance modulo permutation between $\mathbf{\Phi}(\cdot; \mathbf{S}, \mathcal{H})$ and $\mathbf{\Phi}(\cdot; \hat{\mathbf{S}}, \mathcal{H})$ is bounded by

$$\left\| \mathbf{\Phi}(\cdot; \hat{\mathbf{S}}, \mathcal{H}) - \mathbf{\Phi}(\cdot; \mathbf{S}, \mathcal{H}) \right\|_{\mathcal{P}} \leq 2C \left(1 + \delta\sqrt{n}\right) L\epsilon + \mathcal{O}(\epsilon^2).$$

Proof: Let $\mathbf{x}_\ell$ be the Layer ℓ output of GNN $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$. Input signal $\mathbf{x}$ with $\|\mathbf{x}\| = 1$

Let $\hat{\mathbf{x}}_\ell$ be the Layer ℓ output of GNN $\Phi(\mathbf{x}; \hat{\mathbf{S}}, \mathcal{H})$. Input signal $\mathbf{x}$ with $\|\mathbf{x}\| = 1$

- Layer ℓ is a perceptron with filter $\mathbf{H}_\ell \;\Rightarrow\; \Big\| \hat{\mathbf{x}}_\ell - \mathbf{x}_\ell \Big\| = \Bigg\| \sigma\Big[\mathbf{H}_\ell(\hat{\mathbf{S}})\hat{\mathbf{x}}_{\ell-1} \Big] - \sigma\Big[\mathbf{H}_\ell(\mathbf{S})\mathbf{x}_{\ell-1} \Big] \Bigg\|$
- Nonlinearity is normalized Lipschitz $\;\Rightarrow\; \Big\| \hat{\mathbf{x}}_\ell - \mathbf{x}_\ell \Big\| \;\leq\; \Big\| \mathbf{H}_\ell(\hat{\mathbf{S}})\hat{\mathbf{x}}_{\ell-1} - \mathbf{H}_\ell(\mathbf{S})\mathbf{x}_{\ell-1} \Big\|$
- This is the critical step of the proof. The rest of the proof is just algebra.

- In last bound, add and subtract $\mathbf{H}_\ell(\hat{\mathbf{S}})\mathbf{x}_{\ell-1}$. Triangle inequality. Submultiplicative property of norms

$$\left\| \hat{\mathbf{x}}_\ell - \mathbf{x}_\ell \right\| \leq \left\| \mathbf{H}_\ell(\hat{\mathbf{S}})\hat{\mathbf{x}}_{\ell-1} - \mathbf{H}_\ell(\mathbf{S})\mathbf{x}_{\ell-1} + \mathbf{H}_\ell(\hat{\mathbf{S}})\mathbf{x}_{\ell-1} - \mathbf{H}_\ell(\hat{\mathbf{S}})\mathbf{x}_{\ell-1} \right\|$$

$$\leq \left\| \mathbf{H}_\ell(\hat{\mathbf{S}}) - \mathbf{H}_\ell(\mathbf{S}) \right\| \times \left\| \mathbf{x}_{\ell-1} \right\| + \left\| \mathbf{H}_\ell(\hat{\mathbf{S}}) \right\| \times \left\| \hat{\mathbf{x}}_{\ell-1} - \mathbf{x}_{\ell-1} \right\|$$

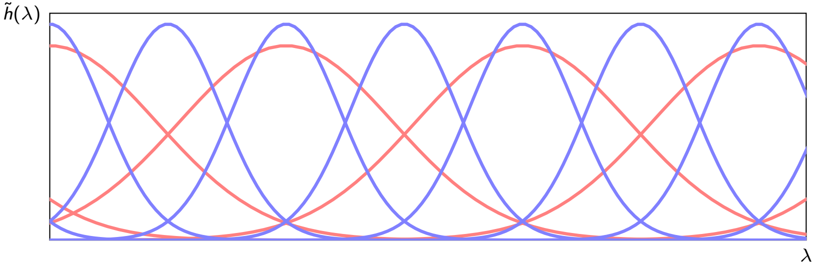
- Since filters are normalized $\Rightarrow$ Filter norm $\left\| \mathbf{H}_\ell(\hat{\mathbf{S}}) \right\| = 1$. Signal norm $\Rightarrow \left\| \mathbf{x}_{\ell-1} \right\| \leq 1$
- Relative perturbations and integral Lipschitz $\Rightarrow \left\| \mathbf{H}_\ell(\hat{\mathbf{S}}) - \mathbf{H}_\ell(\mathbf{S}) \right\| \leq 2C\big(1+\delta\sqrt{n}\big)\epsilon + \mathcal{O}(\epsilon^2)$
- Put all bounds together $\Rightarrow \left\| \hat{\mathbf{x}}_\ell - \mathbf{x}_\ell \right\| \leq 2C\big(1+\delta\sqrt{n}\big)\epsilon \times 1 + 1 \times \left\| \hat{\mathbf{x}}_{\ell-1} - \mathbf{x}_{\ell-1} \right\| + \mathcal{O}(\epsilon^2)$
- Apply recursively from Layer L back to Layer 1. The L factor appears ■

GNNs Inherit the Stability of Graph Filters

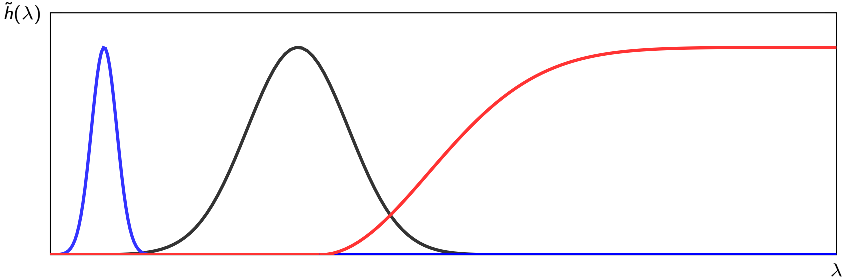
Since Stability is inherited from graph filters, mutatis mutandis, the same observations hold here.

- The stability bounds are universal for all graphs with a given number of nodes
- Bounds depend on filter's Lipschitz constant C and the number of layers L. Which we control.
- And the eigenvector misalignment constant. Which we don't control. Depends on the perturbation.

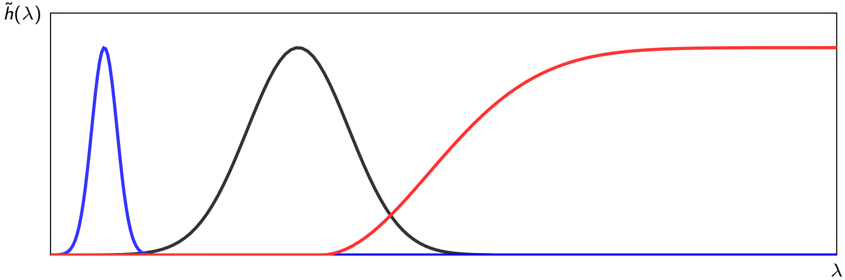
- GNNs whose layers are made up of Lipschitz graph filters are stable to additive deformations
- This is good news $\Rightarrow$ We have a genuine stability vs discriminability tradeoff
- Alas, a bit of a mirage $\Rightarrow$ Graph perturbations are more naturally measured in relative tems



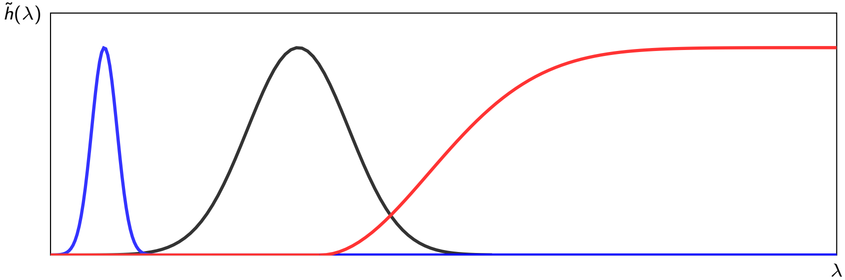
- Meaningful stability claims with respect to relative perturbations require integral Lipschitz filters.
- Integral Lipschitz filters have to be flat at high frequencies. $\Rightarrow$ They can't discriminate
- It is impossible to separate signals with high frequency features and be stable to deformations



- Meaningful stability claims with respect to relative perturbations require integral Lipschitz filters.
- On the flip side, integral Lipschitz filter can be very sharp at low frequencies
- We can be very discriminative at low frequencies. And at the same very stable to deformations



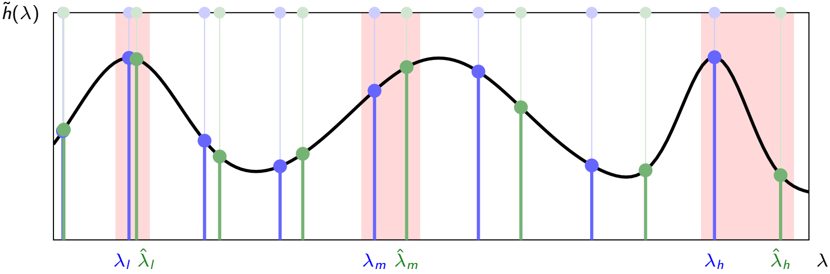
- GNNs use low-pass nonlinearities to demodulate high frequencies into low frequencies
- Where they can be discriminated sharply with a stable filter at the next layer
- Thus, they can be stable and discriminative. Something that linear graph filters can't be



Stability vs Discriminability: An illustrative Example

- The stability vs discriminability tradeoff depends on the frequency components of the signal

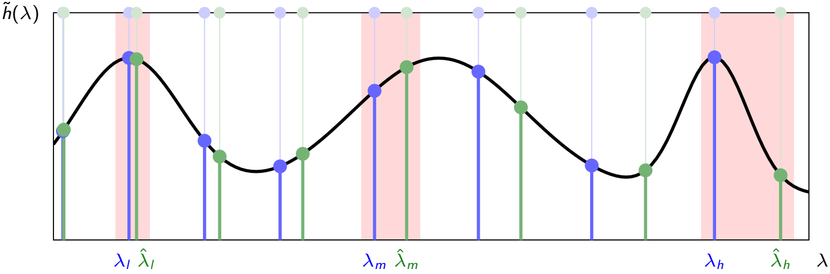
- Meaningful perturbations of a shift operator operator are relative $\Rightarrow \mathbf{P}^T\hat{\mathbf{S}}\mathbf{P} = \mathbf{S} + \mathbf{ES} + \mathbf{SE}$
- Conceptually, we learn all there is to be learnt from dilations $\Rightarrow \hat{\mathbf{S}} = \mathbf{S} + \epsilon\mathbf{S}$
- Eigenvalues are dilated $\lambda_i \to \hat{\lambda}_i = (1+\epsilon)\lambda_i$. Frequency response instantiated on dilated eigenvalues



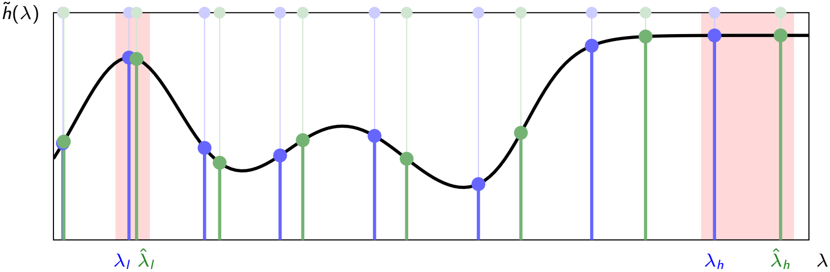
- Higher eigenvalues move more. Signals with high frequency components are more difficult to process

  ⇒ Even small perturbations yield large differences in the filter values that are instantiated

  ⇒ We think we instantiate $h\Big(\lambda_i\Big)$ ⇒ But in reality we instantiate $h\Big(\hat{\lambda}_i\Big) = h\Big((1+\epsilon)\,\lambda_i\Big)$

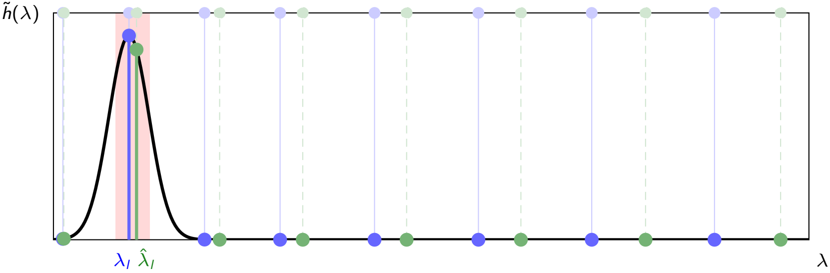


- To attain stable graph signal processing we need integral Lipschitz filters $\Rightarrow \left|\lambda\tilde{h}'(\lambda)\right| \leq C$
- Either the eigenvalue does not change because we are considering low frequencies
- Or the frequency response does not change when we are considering high frequencies



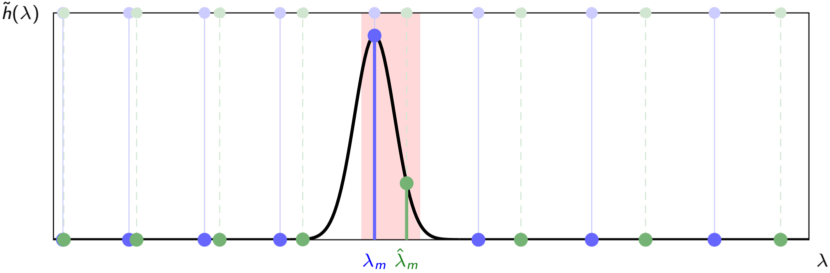
- At low frequencies a sharp highly discriminative filter is also highly stable

$\Rightarrow$ Ideal response $h\Big(\lambda_l\Big)$ is very close to perturbed response $h\Big(\hat{\lambda}_l\Big) = h\Big((1+\epsilon)\,\lambda_l\Big)$



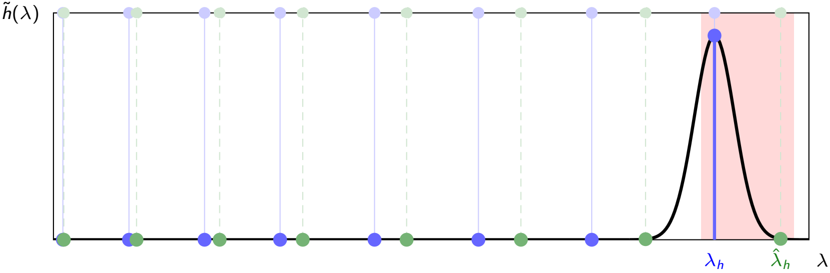
- At intermediate frequencies a sharp highly discriminative filter is somewhat stable

$\Rightarrow$ Ideal response $h\Big(\lambda_m\Big)$ is somewhat close to perturbed response $h\Big(\hat{\lambda}_m\Big) = h\Big((1+\epsilon)\,\lambda_m\Big)$

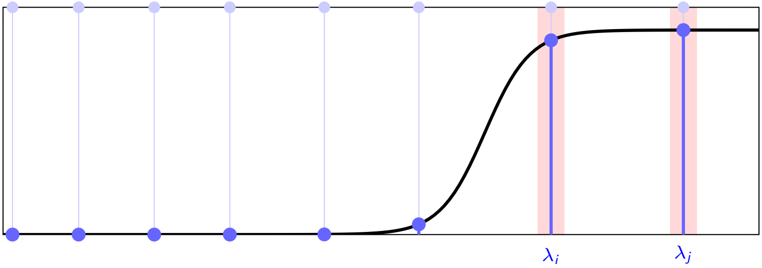


- At high frequencies a sharp highly discriminative filter is unstable. It becomes useless

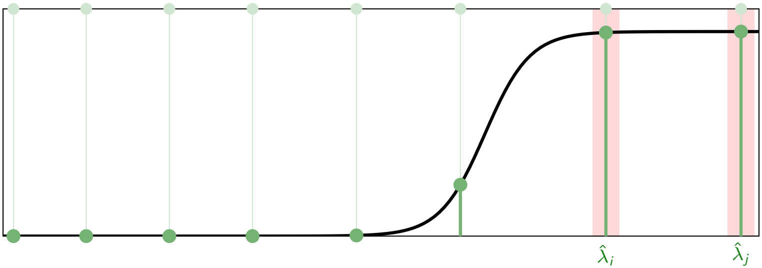
$\Rightarrow$ Ideal response $h\Big(\lambda_h\Big)$ is very different from perturbed response $h\Big(\hat{\lambda}_h\Big) = h\Big((1+\epsilon)\,\lambda_h\Big)$



- Separates them from the rest. But it doesn't discriminate between them



- It is, however, stable to deformations.



Fact: It is impossible to discriminate high frequency components with a stable filter

We can have a filter that is discriminative. Or a filter that is stable. But not one that is both.

Graph $\mathcal{G}$

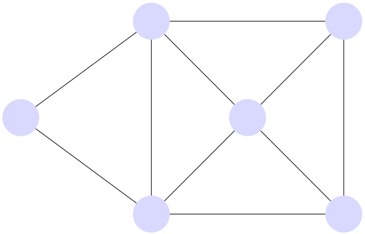


Table: Eigenvalues of $\mathcal{G}$ and Graph Fourier Transform of the graph signal.

λ_n	3.47	0.91	0	-2.00	-1.58	-0.80
$\tilde{\mathbf{x}}(\lambda_n)$	10	0	10	0	0	0

Graph $\mathcal{G}$

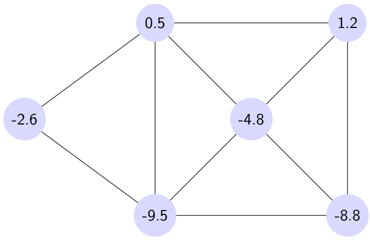


Table: Eigenvalues of $\mathcal{G}$ and Graph Fourier Transform of the graph signal.

λ_n	3.47	0.91	0	-2.00	-1.58	-0.80
$\tilde{\mathbf{x}}(\lambda_n)$	10	0	10	0	0	0

Graph $\mathcal{G}$

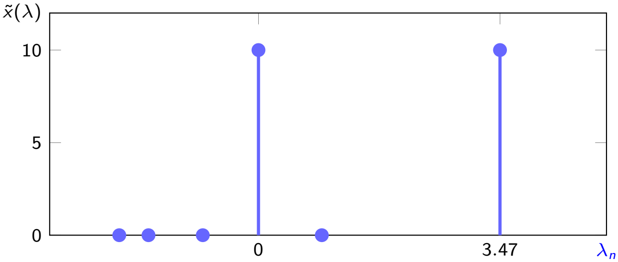
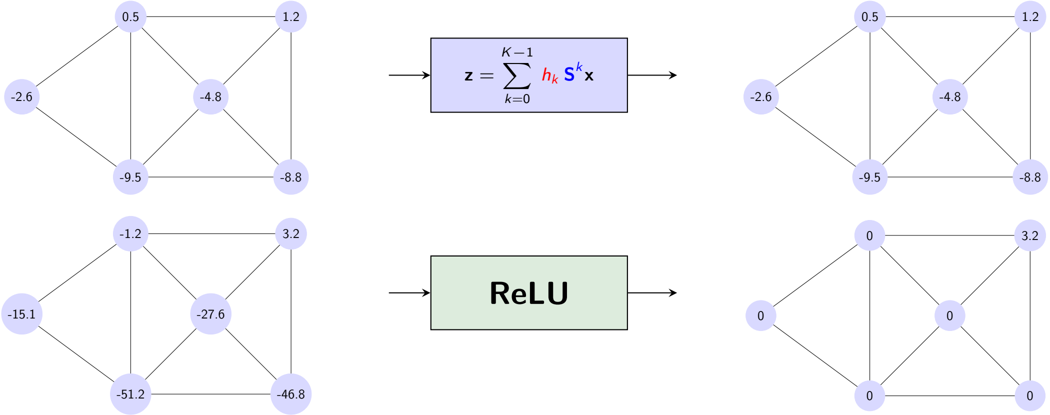
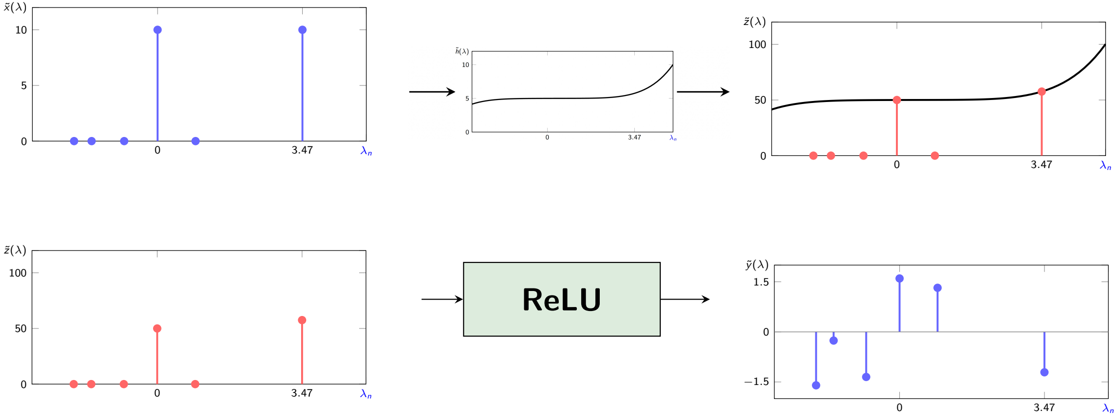
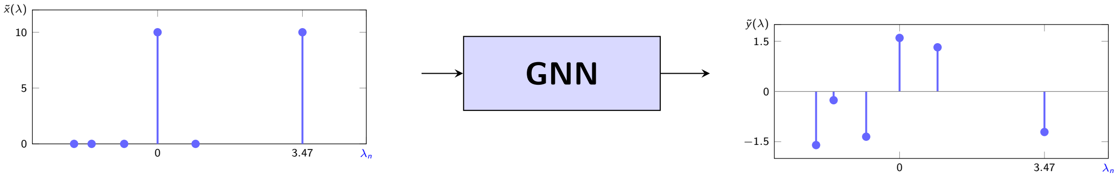


Table: Eigenvalues of $\mathcal{G}$ and Graph Fourier Transform of the graph signal.

λ_n	3.47	0.91	0	-2.00	-1.58	-0.80
$\tilde{\mathbf{x}}(\lambda_n)$	10	0	10	0	0	0

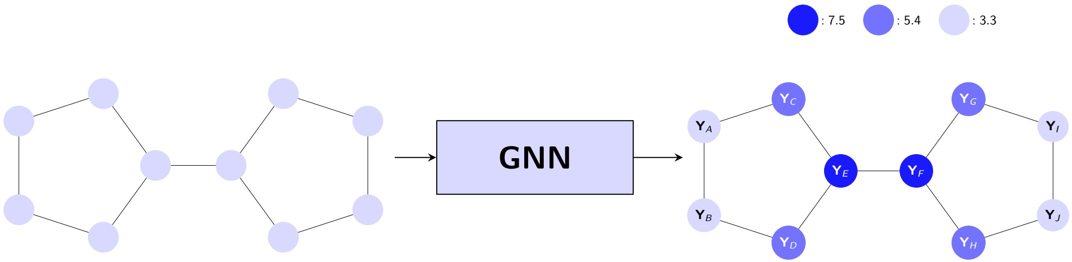

0.5
1.2
-2.6
-4.8
-9.5
-8.8
$z = \sum_{k=0}^{K-1} h_k S^k x$
0.5
1.2
-2.6
-4.8
-9.5
-8.8
-1.2
3.2
-15.1
-27.6
-51.2
-46.8
ReLU
0
3.2
0
0
0
0


$\tilde{x}(\lambda)$
10
5
0
0
3.47
λ_n
$\tilde{h}(\lambda)$
10
5
0
0
3.47
λ_n
$\tilde{z}(\lambda)$
100
50
0
0
3.47
λ_n
$\tilde{z}(\lambda)$
100
50
0
0
3.47
λ_n
ReLU
$\tilde{y}(\lambda)$
1.5
0
−1.5
0
3.47
λ_n

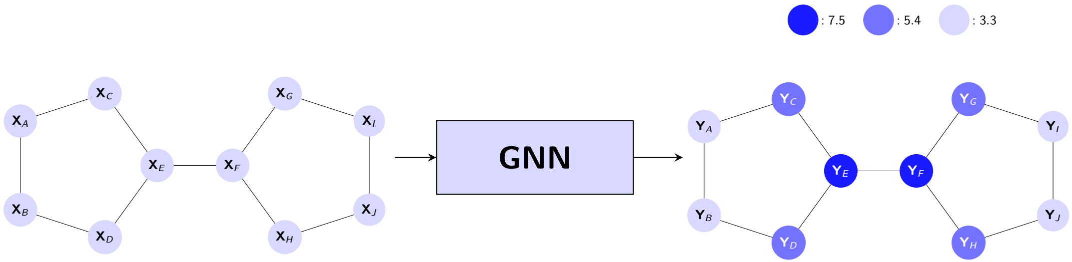


- GNNs use low-pass nonlinearities to demodulate high frequencies into low frequencies
- They can add information to zero frequency components
- Thus, they can be stable and discriminative/expressive.

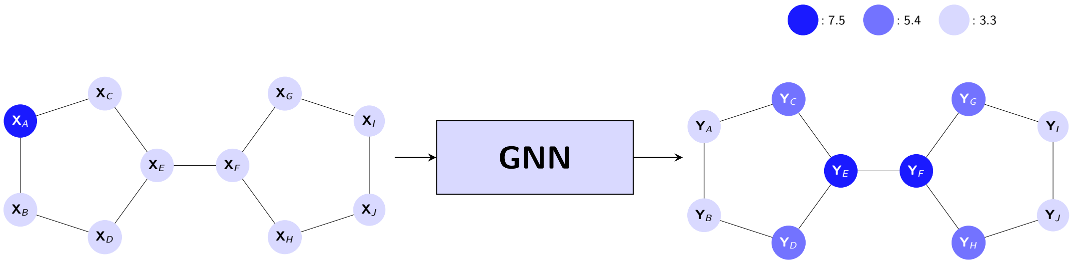
Representation Power of Graph Neural Networks



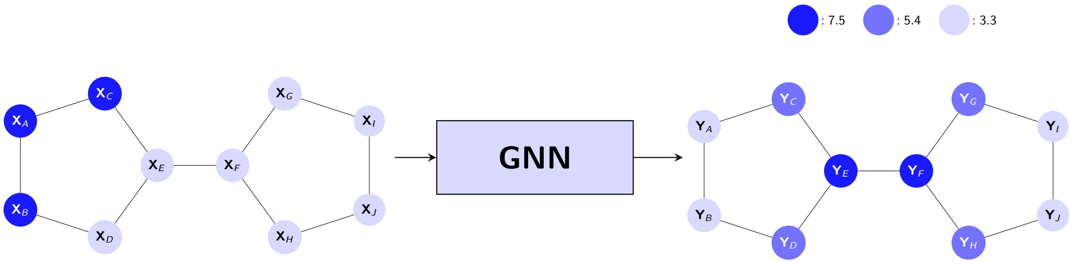
$$\text{GNN} : \mathbb{G} \rightarrow \mathbb{R}^N$$



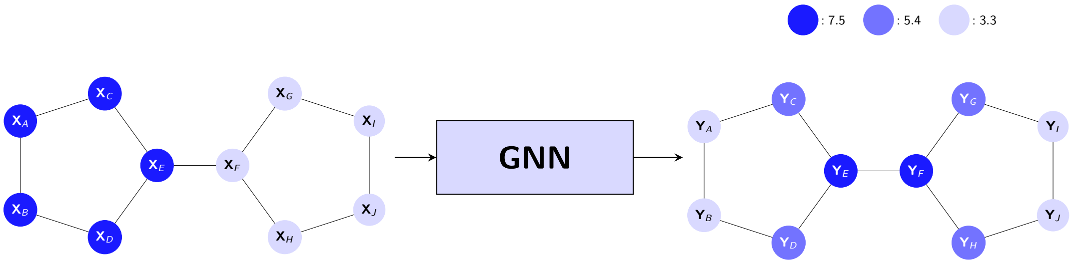
$$\text{GNN} : \mathbb{G} \rightarrow \mathbb{R}^N$$



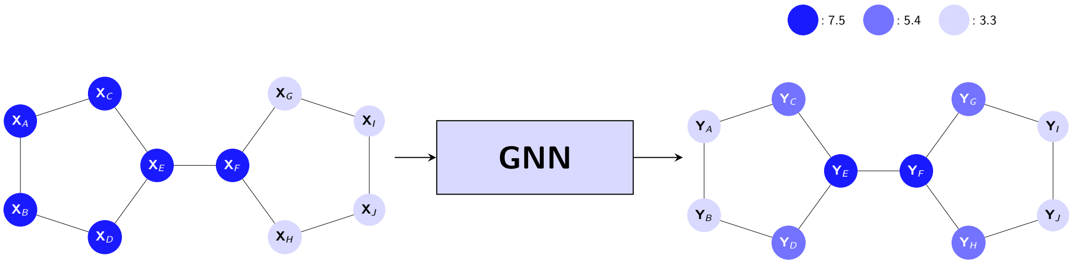
$$\text{GNN} : \mathbb{G} \rightarrow \mathbb{R}^N$$



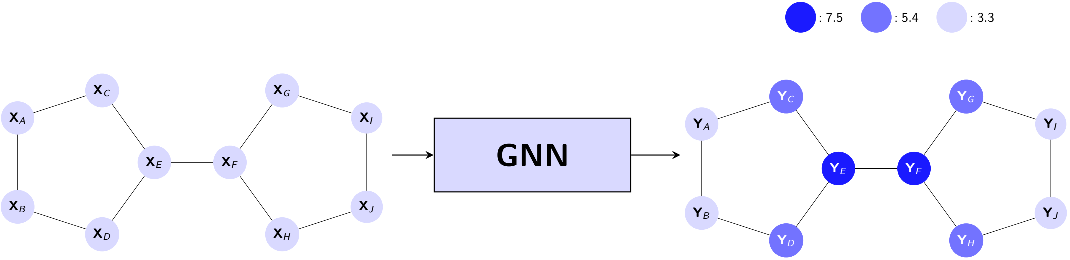
$$\text{GNN} : \mathbb{G} \to \mathbb{R}^N$$



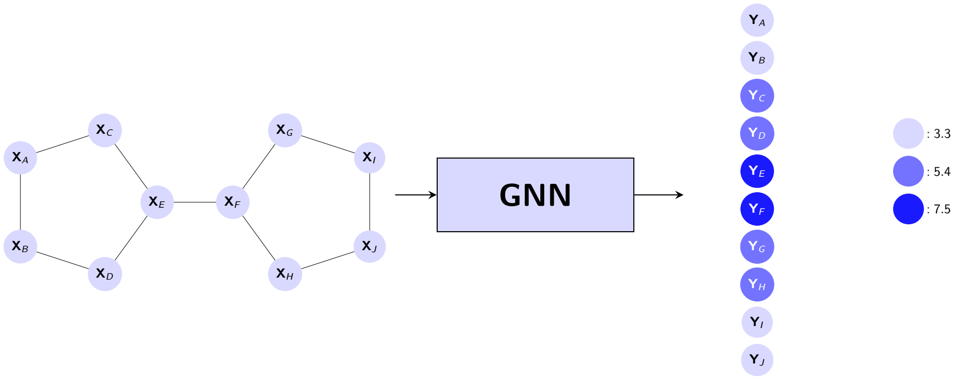
$$\text{GNN} : \mathbb{G} \rightarrow \mathbb{R}^N$$



$$\text{GNN} : \mathbb{G} \to \mathbb{R}^N$$



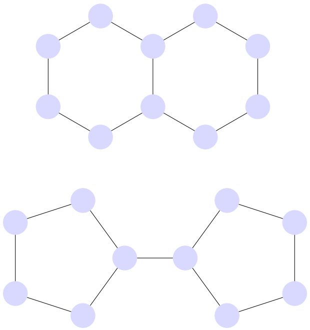
$$\text{GNN} : \mathbb{G} \to \mathbb{R}^N$$



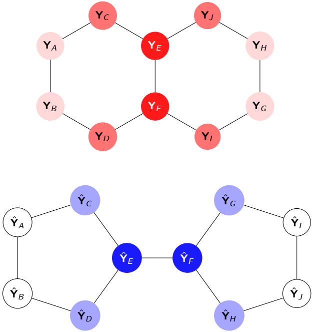
$$\text{GNN} : \mathbb{G} \rightarrow \mathbb{R}^N$$

Problem Definition

Given a pair of **different** (non-isomorphic) graphs $\mathcal{G}$, $\hat{\mathcal{G}}$ with adjacency matrices $\mathbf{S}$, $\hat{\mathbf{S}}$ and anonymous inputs $\mathbf{x}$, $\hat{\mathbf{x}}$, is there a GNN $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ such that $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \neq_{\Pi} \Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$?

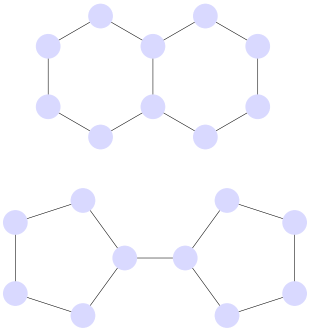


$$\Rightarrow \Phi\left(\mathbf{x}; \mathbf{S}, \mathcal{H}\right) \Rightarrow$$

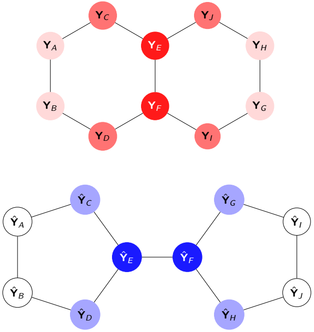


Problem Definition

Given a pair of **different** (non-isomorphic) graphs $\mathcal{G}$, $\hat{\mathcal{G}}$ with adjacency matrices $\mathbf{S}$, $\hat{\mathbf{S}}$ and anonymous inputs $\mathbf{x}$, $\hat{\mathbf{x}}$, is there a GNN $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ such that $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \neq_{\Pi} \Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$?

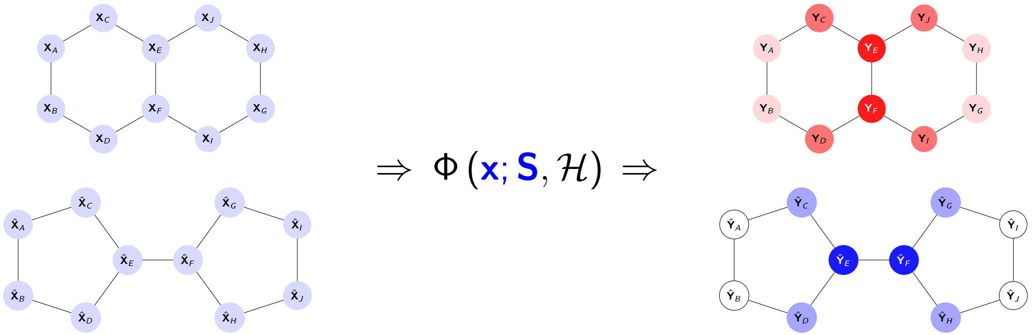


$$\Rightarrow \Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \Rightarrow$$



Problem Definition

Given a pair of **different** (non-isomorphic) graphs $\mathcal{G}$, $\hat{\mathcal{G}}$ with adjacency matrices $\mathbf{S}$, $\hat{\mathbf{S}}$ and anonymous inputs $\mathbf{x}$, $\hat{\mathbf{x}}$, is there a GNN $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ such that $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \neq_{\Pi} \Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$?



Problem Definition

Given a pair of **different** (non-isomorphic) graphs $\mathcal{G}$, $\hat{\mathcal{G}}$ with adjacency matrices $\mathbf{S}$, $\hat{\mathbf{S}}$ and anonymous inputs $\mathbf{x}$, $\hat{\mathbf{x}}$, is there a GNN $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ such that $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \neq_{\Pi} \Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$?

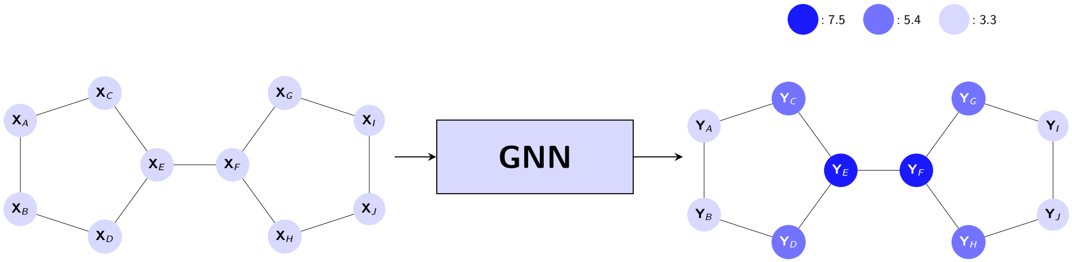
- The inputs are anonymous $\Rightarrow$ structure and identity agnostic, i.e., they carry **no** information about the graph and the nodes.
- Study the ability of a GNN to **generate** information about the graph.
- GNNs have high representation power, if they can produce discriminative representations from anonymous inputs, for a large class of graphs

Problem Definition

Given a pair of **different** (non-isomorphic) graphs $\mathcal{G}$, $\hat{\mathcal{G}}$ with adjacency matrices $\mathbf{S}$, $\hat{\mathbf{S}}$ and anonymous inputs $\mathbf{x}$, $\hat{\mathbf{x}}$, is there a GNN $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ such that $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \neq_{\Pi} \Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$?

- The expressive power of a GNN is related to the function approximation properties of GNNs.
- The above definition is related to the Graph Isomorphism problem $\Rightarrow$ It belongs to the class of NP problems. It is not known whether it is P or NP-complete.
- A number of algorithms exist that can separate a large set of nonisomorphic graphs e.g., the Weisfeiler-Lehman (WL) test.

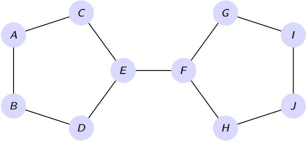
Vertex Domain Analysis

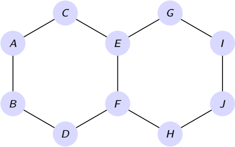


Results of Vertex Domain Analysis:

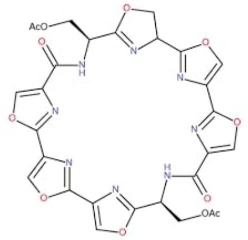
- GNNs are at most as powerful as the Weisfeiler-Lehman (WL) test [Weisfeiler, & Leman, 1968].
- GNNs cannot produce informative representations for a large class of real-world graphs.

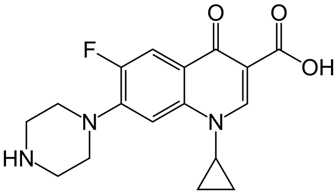
- WL indistinguishable non-isomorphic graphs.



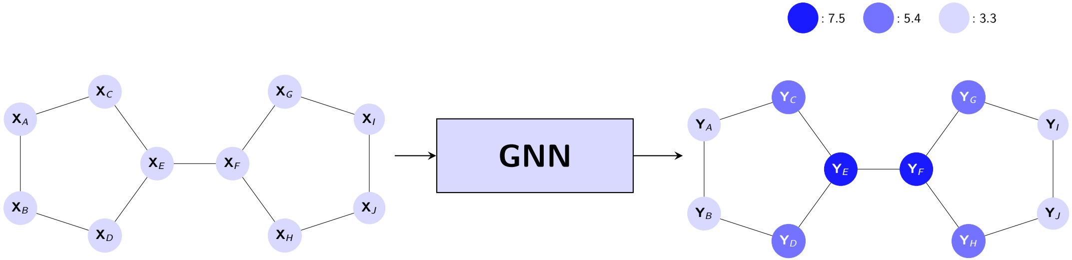


- Heterocyclic antibiotics.

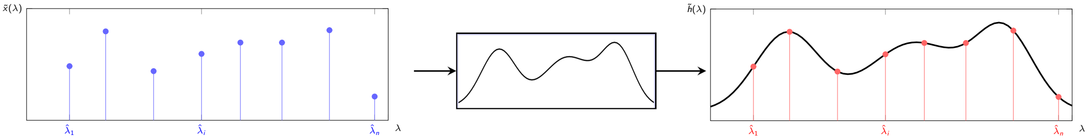




Vertex Domain Analysis



Frequency Domain Analysis



- GNNs are **more powerful** than the WL test!
- Can produce **informative** representations for almost all practical graphs.
- Stay tuned to see how this analysis can be used in practice to **efficiently train GNNs**.

- [Xu et al '19] study the representation power of $\Phi(\mathbf{1}; \mathbf{S}, \mathcal{H})$ and not of $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$.

- Studying a function $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ by only observing a subset of the domain of $\Phi(\mathbf{1}; \mathbf{S}, \mathcal{H})$, cannot yield concrete conclusions about the **representation power**.

- The **all-one vector** is associated with limitations involving the spectral decomposition of the graph.

Frequency Domain Analysis

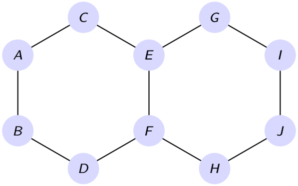
- **Spectral Decomposition: $\mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$**

- Recall the definition of a **Graph Convolution**: $\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{V}\boldsymbol{\Lambda}^k \mathbf{V}^T \mathbf{x}$.
- When $\mathbf{x} = \mathbf{1}$:

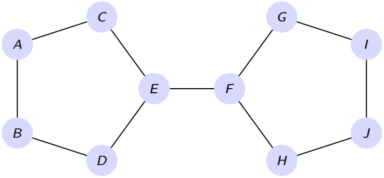
$$\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{1} = \sum_{k=0}^{K-1} h_k \mathbf{V}\boldsymbol{\Lambda}^k \mathbf{V}^T \mathbf{1} = \sum_{n=1}^{N} \sum_{k=0}^{K-1} h_k \lambda_n^k \left(\mathbf{v}_n^T \mathbf{1} \right) \mathbf{v}_n$$

- The information associated with the spectral components that are orthogonal to one has been **lost!**

Graph $\mathcal{G}$



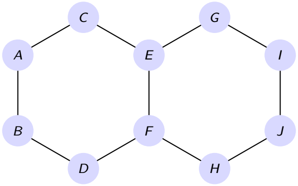
Graph $\hat{\mathcal{G}}$



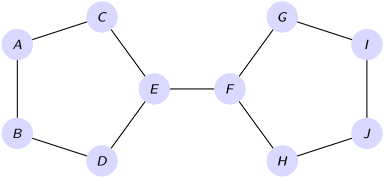
$\mathcal{G}$	λ_n	2.303	1.618	1.303	1	0.618	-2.303	-1.618	-0.618	-1	-1.303
$\hat{\mathcal{G}}$	$\hat{\lambda}_n$	2.303	1.861	1	0.618	0.618	0.254	-1.303	-1.618	-1.618	-2.115

Spectral limitations of GNNs with all-one inputs

Graph $\mathcal{G}$

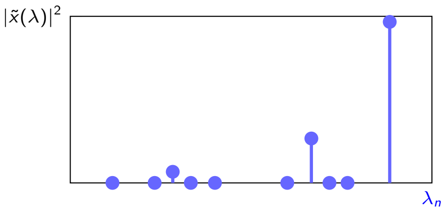


Graph $\hat{\mathcal{G}}$

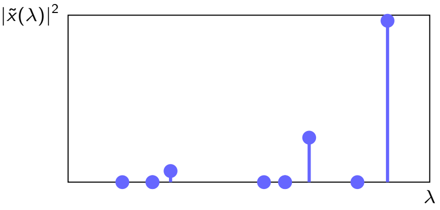


$\mathcal{G}$	λ_n	2.303	1.618	1.303	1	0.618	-2.303	-1.618	-0.618	-1	-1.303
	$\langle \boldsymbol{v}_n, \mathbf{1} \rangle$	3.048	0	0	-0.816	0	0	0	0	0	-0.210
$\hat{\mathcal{G}}$	$\hat{\lambda}_n$	2.303	1.861	1	0.618	0.618	0.254	-1.303	-1.618	-1.618	-2.115
	$\langle \hat{\boldsymbol{v}}_n, \mathbf{1} \rangle$	3.048	0	-0.816	0	0	0	-0.210	0	0	0

Graph $\mathcal{G}$



Graph $\hat{\mathcal{G}}$



$\mathcal{G}$	λ_n	2.303	1.618	1.303	1	0.618	-2.303	-1.618	-0.618	-1	-1.303
	$\langle \boldsymbol{v}_n, \mathbf{1} \rangle$	3.048	0	0	-0.816	0	0	0	0	0	-0.210
$\hat{\mathcal{G}}$	$\hat{\lambda}_n$	2.303	1.861	1	0.618	0.618	0.254	-1.303	-1.618	-1.618	-2.115
	$\langle \hat{\boldsymbol{v}}_n, \mathbf{1} \rangle$	3.048	0	-0.816	0	0	0	-0.210	0	0	0

Problem Definition

Given a pair of **different** (non-isomorphic) graphs $\mathcal{G}$, $\hat{\mathcal{G}}$ with adjacency matrices $\mathbf{S}$, $\hat{\mathbf{S}}$ and anonymous inputs $\mathbf{x}$, $\hat{\mathbf{x}}$, is there a GNN $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ such that $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \neq \Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$?

- Revisit the above problem by studying GNNs with linear algebraic tools and white inputs.
- GNNs produce different representations for **all graphs** with different eigenvalues.
- The novel **constructive analysis** enables the design of simple and expressive GNNs.

White input properties

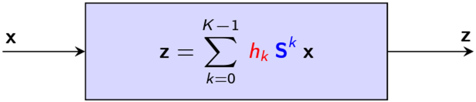
- $\mathbf{x}$ is a white random vector: $\mathbb{E}[\mathbf{x}] = 0, \;\; \mathbb{E}[\mathbf{x}\mathbf{x}^T] = \sigma^2 \mathbf{I}$

- White vectors are anonymous:

  $\Rightarrow$ structure and identity agnostic.

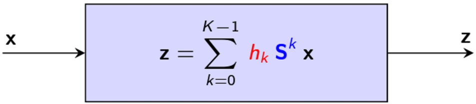
- White inputs allow us to study the domain of $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H}) \;\; \Rightarrow \mathbf{x} \in \mathbb{R}^N$.

- White inputs do not admit the spectral limitations of $\mathbf{x} = \mathbf{1}$.



- $\mathbf{x}$ is a random vector with $\mathbb{E}\left[\mathbf{x}\right] = 0 \;\Rightarrow$ $\mathbf{z}$ is a random vector with $\mathbb{E}\left[\mathbf{z}\right] = 0$.

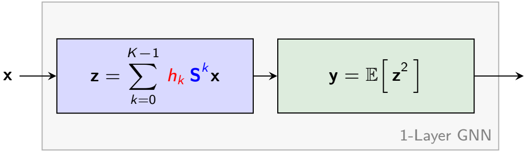
- $\mathbf{x}$ is a random vector with $\mathbb{E}\left[\mathbf{x}^2\right] = \text{diag}\left(\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]\right) = \sigma^2\mathbf{1} \;\Rightarrow$ $\mathbf{z}$ is a random vector with:



- $\mathbf{x}$ is a random vector with $\mathbb{E}[\mathbf{x}] = 0 \;\Rightarrow\; \mathbf{z}$ is a random vector with $\mathbb{E}[\mathbf{z}] = 0$.

- $\mathbf{x}$ is a random vector with $\mathbb{E}[\mathbf{x}^2] = \text{diag}\left(\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]\right) = \sigma^2 \mathbf{1} \;\Rightarrow\; \mathbf{z}$ is a random vector with:

$$\text{var}[\mathbf{z}] = \mathbb{E}[\mathbf{z}^2] = \text{diag}\left(\mathbb{E}\left[\mathbf{z}\mathbf{z}^T\right]\right) = \text{diag}\left(\sum_{k=0}^{2K-2} h'_k \mathbf{S}^k\right) = \sum_{k=0}^{2K-2} h'_k \text{diag}\left(\mathbf{S}^k\right)$$



- A white input is passed through a convolutional Graph Filter.
- The filter output is being processed by the square nonlinearity and the expectation operator to measure the variance.
- The output is a linear combination of the adjacency powers diagonals: $\mathbb{E}\left[\mathbf{z}^2\right] = \sum_{k=0}^{2K-2} h'_k \text{diag}\left(\mathbf{S}^k\right)$

$\Rightarrow$ **permutation equivariant.**

Theorem [Kanatsoulis et al '22]

Given **non-isomorphic** graphs $\mathcal{G}$, $\hat{\mathcal{G}}$ and a GNN with **anonymous** input $\mathbf{\Phi}(\mathbf{x}; \mathbf{S}, \mathcal{H}) : \mathbb{G} \to \mathbb{R}^N$. If $\mathcal{G}$ and $\hat{\mathcal{G}}$ have different eigenvalues, $\mathbf{\Phi}(\mathbf{x}; \mathbf{S}, \mathcal{H})$ maps $\mathcal{G}$ and $\hat{\mathcal{G}}$ to different representations.

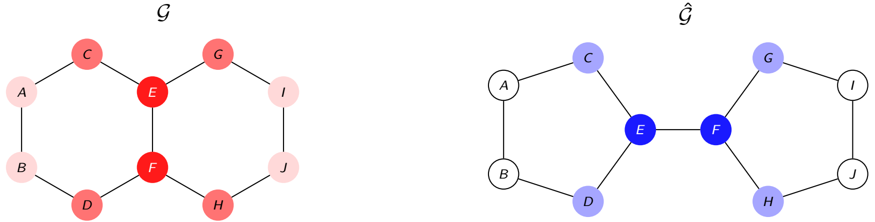
- GNNs produce different representations for **all graphs** with different eigenvalues.
- GNNs are more powerful than the **WL test**!

Theorem [Kanatsoulis et al '22]

Given **non-isomorphic** graphs $\mathcal{G}$, $\hat{\mathcal{G}}$ and a GNN with **anonymous** input $\boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}, \mathcal{H}) : \mathbb{G} \to \mathbb{R}^N$. If $\mathcal{G}$ and $\hat{\mathcal{G}}$ have different eigenvalues, $\boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}, \mathcal{H})$ maps $\mathcal{G}$ and $\hat{\mathcal{G}}$ to different representations.

- The majority of real graphs have different eigenvalues.
- Cospectral graphs involve certain tree structures.

- GNNs can differentiate between **WL indistinguishable** non-isomorphic graphs.



- $\mathbf{y} = \mathbb{E}\left[\mathbf{z}^2\right] = \sum_{k=0}^{5} h_k \text{diag}\left(\mathbf{S}^k\right)$, for $(h_0, h_1, h_2, h_3, h_4, h_5) = (10, 1, -\frac{1}{2}, \frac{1}{3}, -\frac{1}{4}, \frac{1}{5})$.

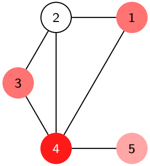
Graph \ Node	A	B	C	D	E	F	G	H	I	J
$\mathcal{G}$	7.5	7.5	7.25	7.25	5.25	5.25	7.25	7.25	7.5	7.5
$\hat{\mathcal{G}}$	7.9	7.9	7.65	7.65	5.65	5.65	7.65	7.65	7.9	7.9

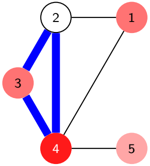
Proposition [Kanatsoulis et al '22]

A GNN with **anonymous** input can **compute** $\mathbf{c}^{(k)} = \text{diag}\left(\mathbf{S}^{k}\right) \in \mathbb{N}_0^N$ that count the number of closed paths of length k from each node to itself.

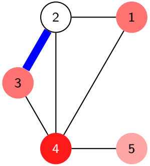
- For $k = 0 \Rightarrow \mathbf{c}^{(k)} = \mathbf{1}$
- For $k = 1 \Rightarrow \mathbf{c}^{(k)} = \mathbf{0}$
- For $k = 2 \Rightarrow \mathbf{c}^{(k)} = \mathbf{S1}$, counts the 1-hop neighbors of each node.

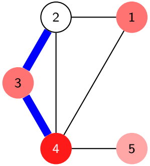
- For $k = 3 \Rightarrow \mathbf{c}^{(k)}$ counts the number of triangles each node is involved in.

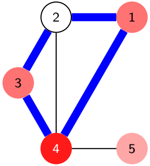




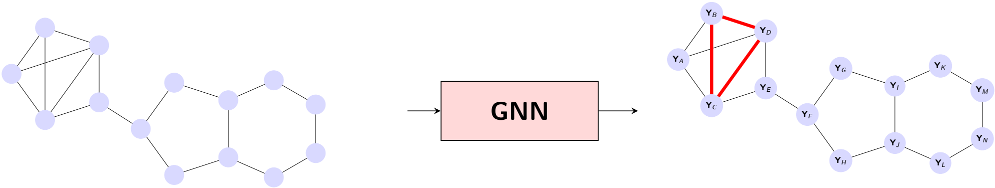
- For $k = 4 \Rightarrow \mathbf{c}^{(k)}$ counts the 1-hop neighbors, the 2-hop neighbors and the number of tetragons each node is involved in.



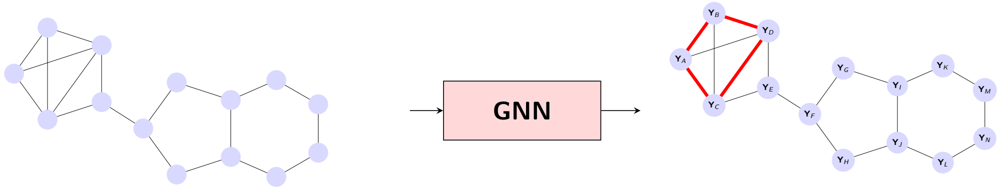




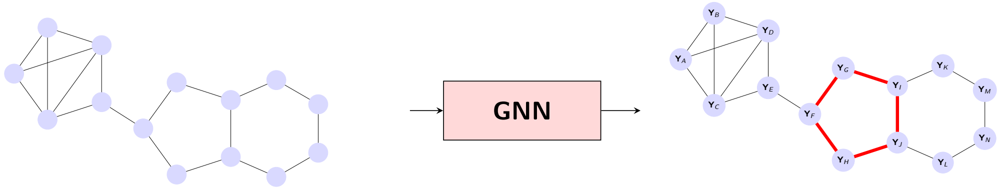
- Investigate the **function approximation properties of GNNs** and relate them to graph theory.



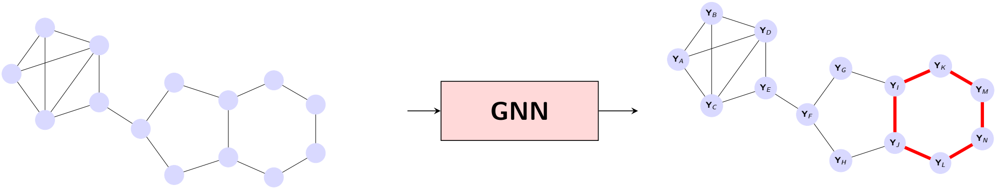
- Investigate the **function approximation properties of GNNs** and relate them to graph theory.



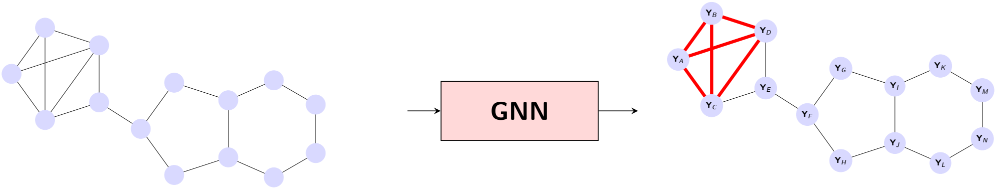
- Investigate the **function approximation properties of GNNs** and relate them to graph theory.



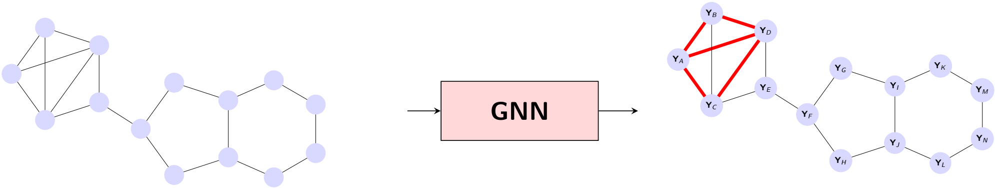
- Investigate the **function approximation properties of GNNs** and relate them to graph theory.



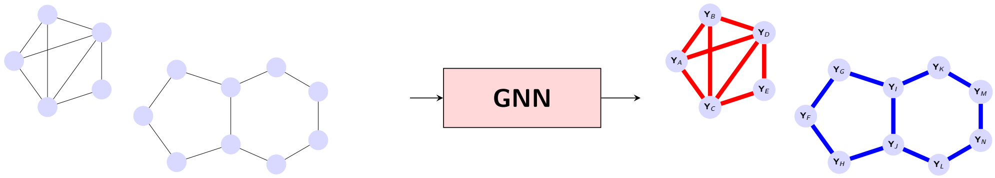
- Investigate the **function approximation properties of GNNs** and relate them to graph theory.

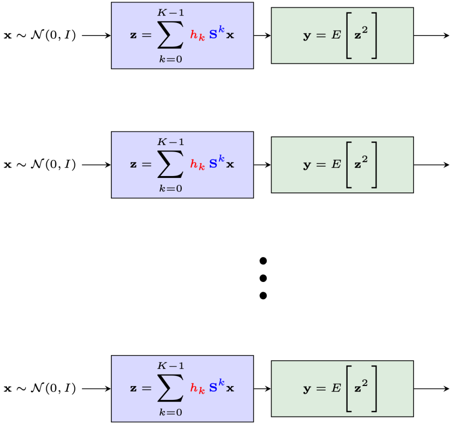


- Investigate the **function approximation properties of GNNs** and relate them to graph theory.

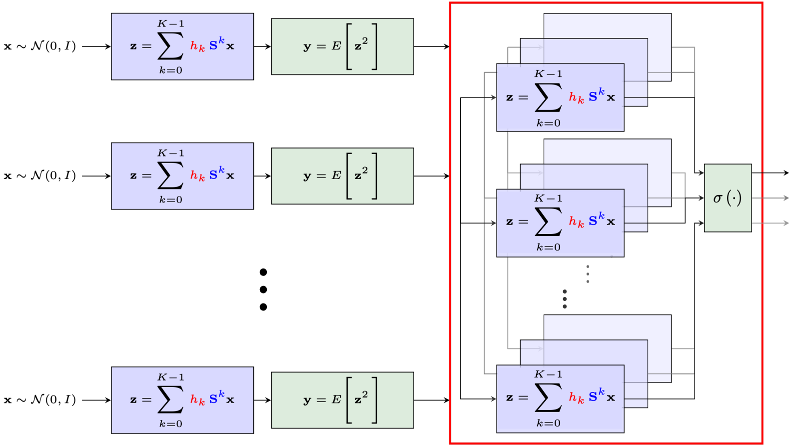


- Investigate the **function approximation properties of GNNs** and relate them to graph theory.

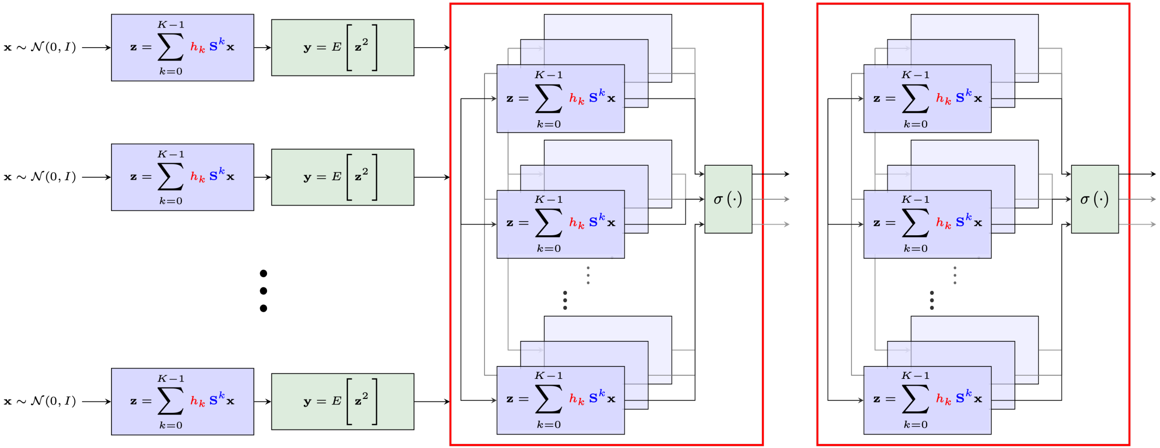



$\mathbf{x} \sim \mathcal{N}(0, I)$
$\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$
$\mathbf{y} = E\left[\mathbf{z}^2\right]$
$\mathbf{x} \sim \mathcal{N}(0, I)$
$\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$
$\mathbf{y} = E\left[\mathbf{z}^2\right]$
$\mathbf{x} \sim \mathcal{N}(0, I)$
$\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$
$\mathbf{y} = E\left[\mathbf{z}^2\right]$

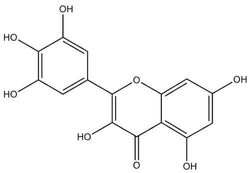
Powerful GNN architectures



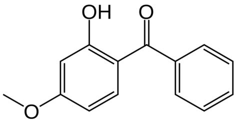
Powerful GNN architectures



Given a **collection of graphs** that belong to different classes, we aim to predict the class of each graph.



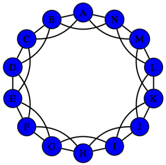
$\Longrightarrow$ Positive in **Lung Cancer** Prevention



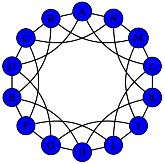
$\Longrightarrow$ Negative in **Lung Cancer** Prevention

- The Circular Skip Link (CSL) dataset is the **golden standard** when it comes to benchmarking GNNs for isomorphism and classification.

- CSL contains 150 4-regular graphs, that belong to one of 10 classes, each consisting of 41 nodes and 164 edges.



(a) $\mathcal{G}$



(b) $\hat{\mathcal{G}}$

- The all-one vector is always an eigenvector in regular graphs, therefore GNNs with $\mathbf{x} = \mathbf{1}$ cannot classify these graphs.

- The variance of a graph filter with filter parameters:

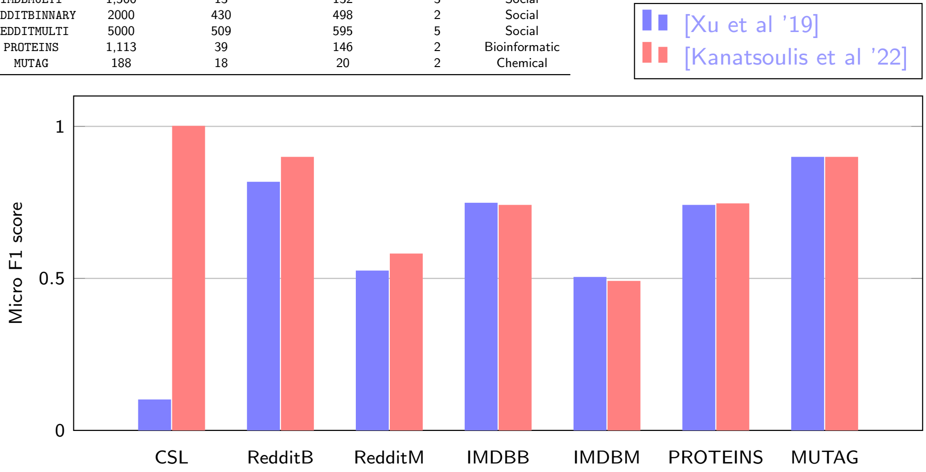
$$(h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9) = \left(0, 1, -\frac{1}{2}, \frac{1}{3}, -\frac{1}{4}, \frac{1}{5}, -\frac{1}{6}, \frac{1}{7}, -\frac{1}{8}, \frac{1}{9}\right)$$

can perfectly classify these graphs.

Table: GNN output $\mathbf{y}$ for every class of the CSL graphs.

Class									
0	1	2	3	4	5	6	7	8	9
73616	-45968	1059	-30593	-25345	-26001	-17555	-28543	16065	-21163

Dataset	# Graphs	Average # Vertices	Average # Edges	# Classes	Network Type
CSL	150	41	164	10	Circulant
IMDBBINARY	1,000	20	193	2	Social
IMDBMULTI	1,500	13	132	3	Social
REDDITBINNARY	2000	430	498	2	Social
REDDITMULTI	5000	509	595	5	Social
PROTEINS	1,113	39	146	2	Bioinformatic
MUTAG	188	18	20	2	Chemical



- Graph Neural Networks are more powerful than we think:

  $\Rightarrow$ **Differentiate** between almost all real graphs.

  $\Rightarrow$ **Count** substructures of the graph.

- Our framework yields **improved performance** of GNNs in graph classification.