CIKM 24' Tutorial

Unifying Spectral and Spatial Graph Neural Networks

Chen, Zhiqian
Assistant Professor
Mississippi State University

Zhang, Lei
Assistant Professor
Northern Illinois University

Zhao, Liang
Associate Professor
Emory University

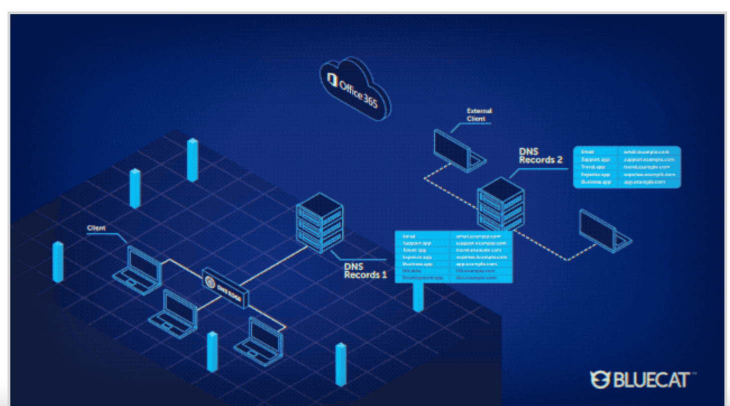
Agenda

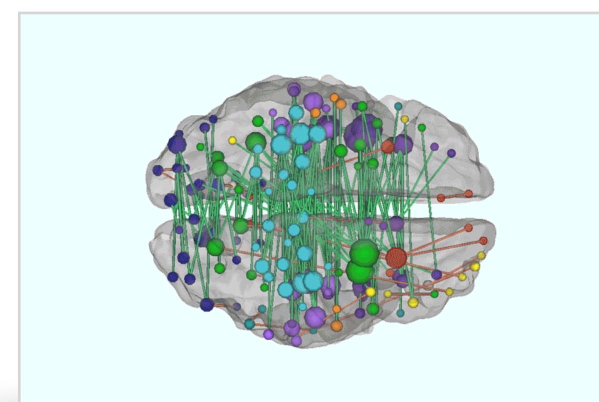
- **First Half (1 hour 30 min)**
  - *Background*: **unified frameworks for GNN** (45 min)
  - *Preliminary*: **graph convolutions** (45 min)
  - BREAK (30 min)
- **Second Half (1 hour)**
  - *Introduction*: **a new unified framework** (40 min)
  - *Future directions* (20min)
  - Q&A (15min)

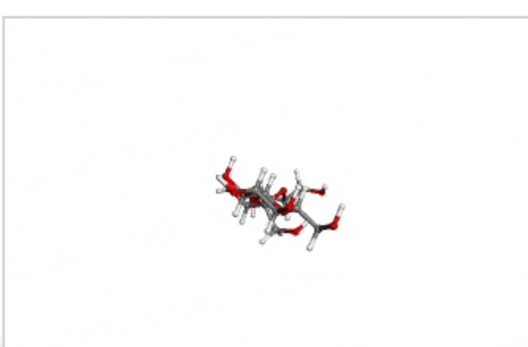
Agenda

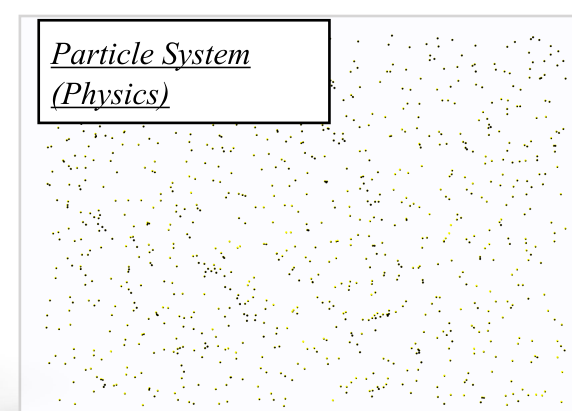
- **First Half (1 hour 30 min)**
  - *Background*: **unified frameworks for GNN** (45 min)
  - *Preliminary*: **graph convolutions** (45 min)
- **BREAK (30 min)**
- **Second Half (1 hour)**
  - *Introduction*: **a new unified framework** (40 min)
  - *Future directions* (20min)
- **Q&A (15min)**

Graph is Pervasive

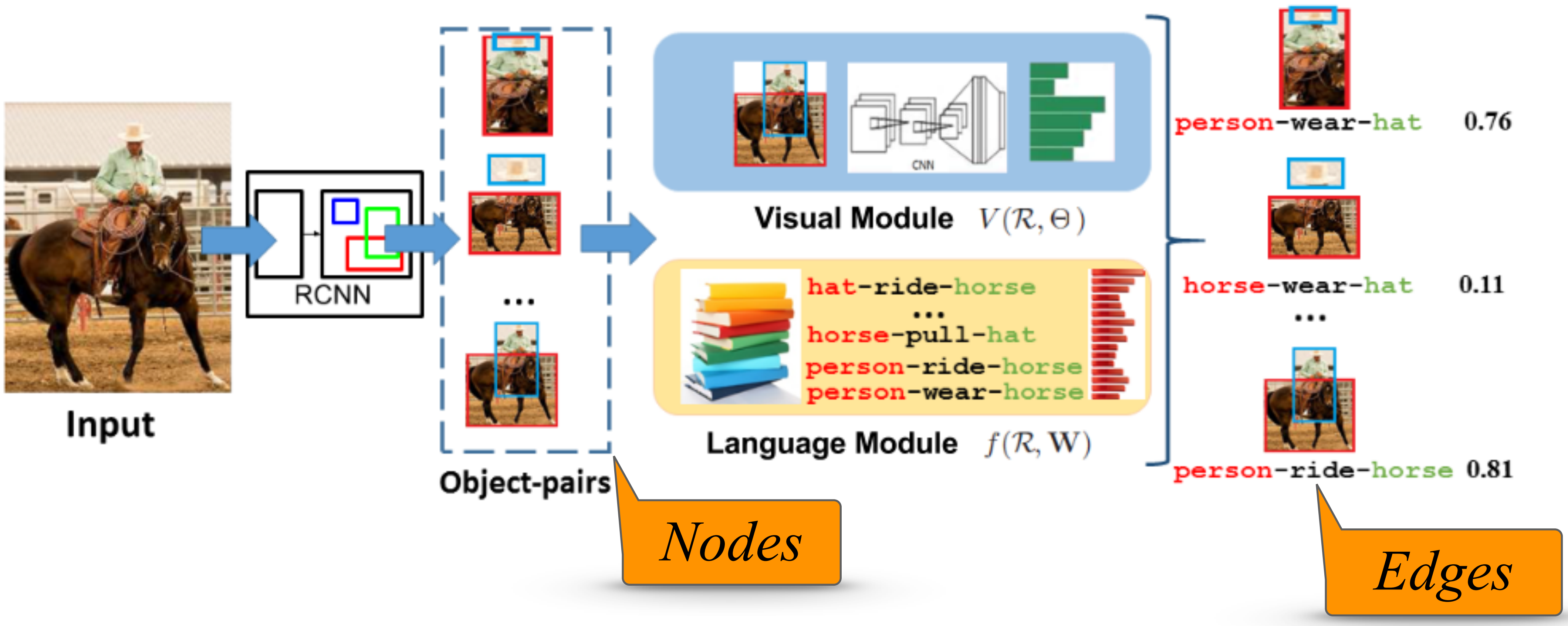








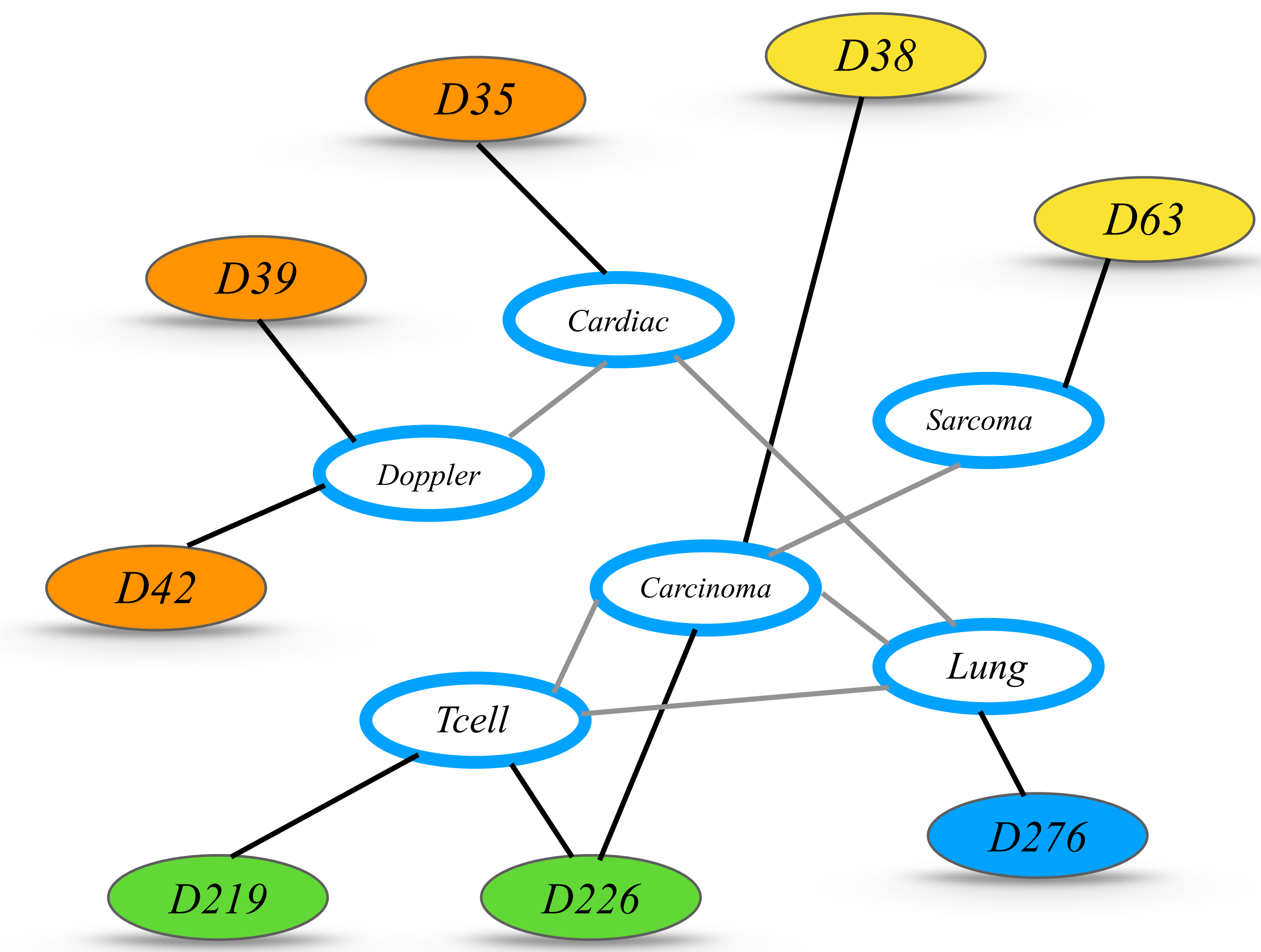
Graphs in CV



Graphs in Image Data:
Scene Graph

Lu, Cewu, et al. "**Visual relationship detection with language priors.**" ECCV, 2016.

Graphs in NLP



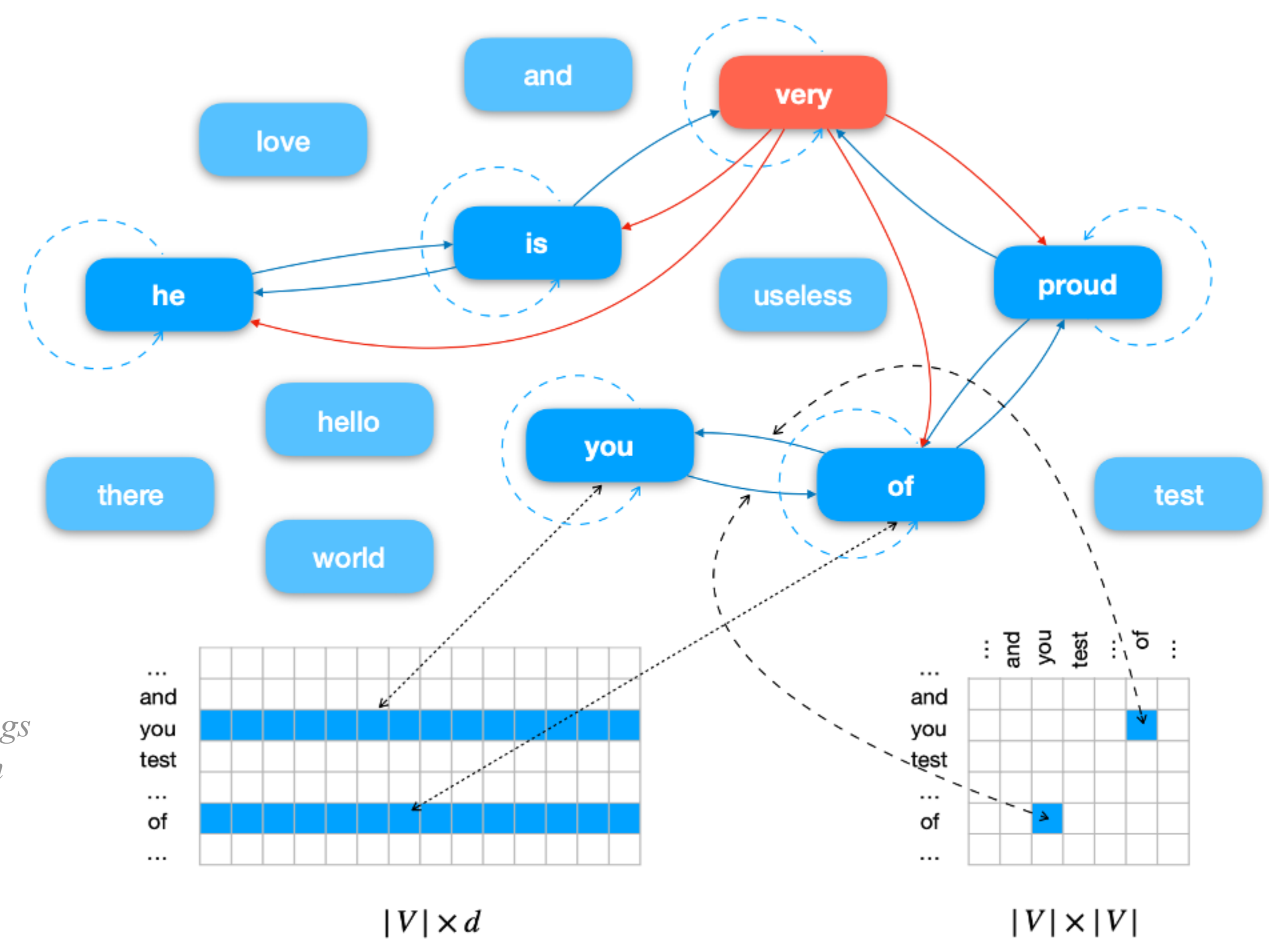
Inductive setting:
Document Graph

Huang, Lianzhe, et al. "Text Level Graph Neural Network for Text Classification." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

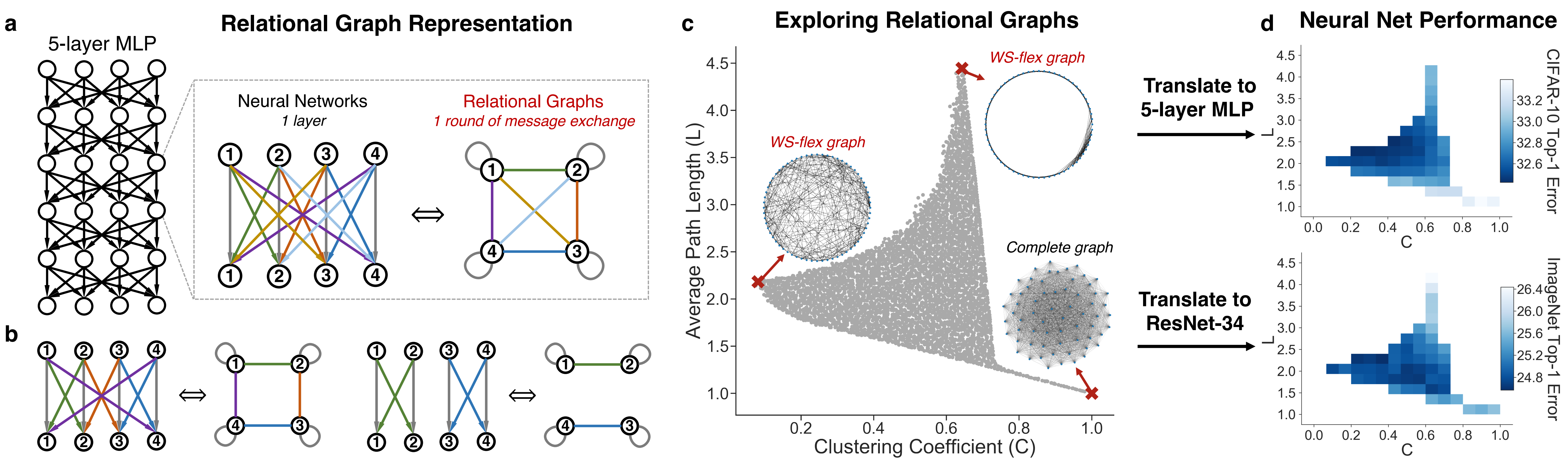
Transduction setting:
Word-Document Graph

- Word-word edges: PMI
- Word-Document edges: TF-IDF

Yao, Liang, Chengsheng Mao, and Yuan Luo. "Graph convolutional networks for text classification." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.



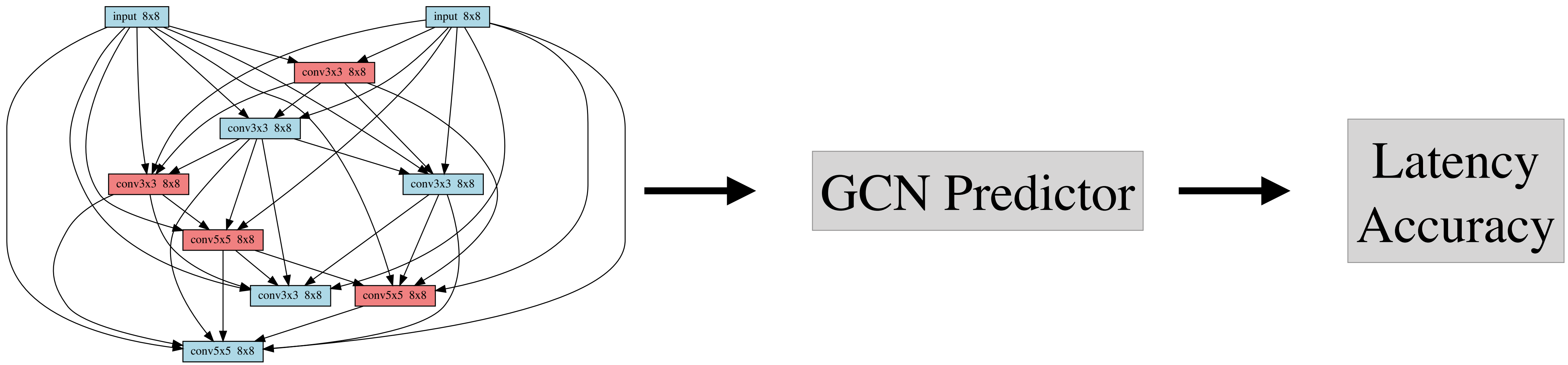
Graphs in Neural Networks Architecture



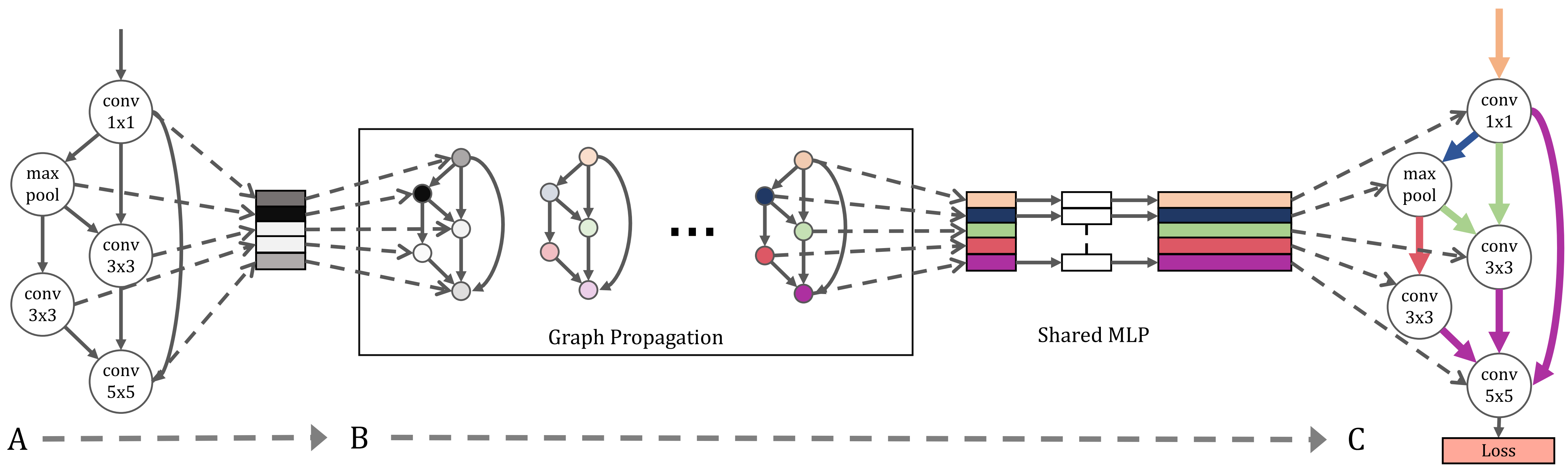
Graph topology and neural network architectures

You, Jiaxuan, et al. "**Graph structure of neural networks.**" International Conference on Machine Learning. PMLR, 2020.

Graphs in Neural Networks Architectures

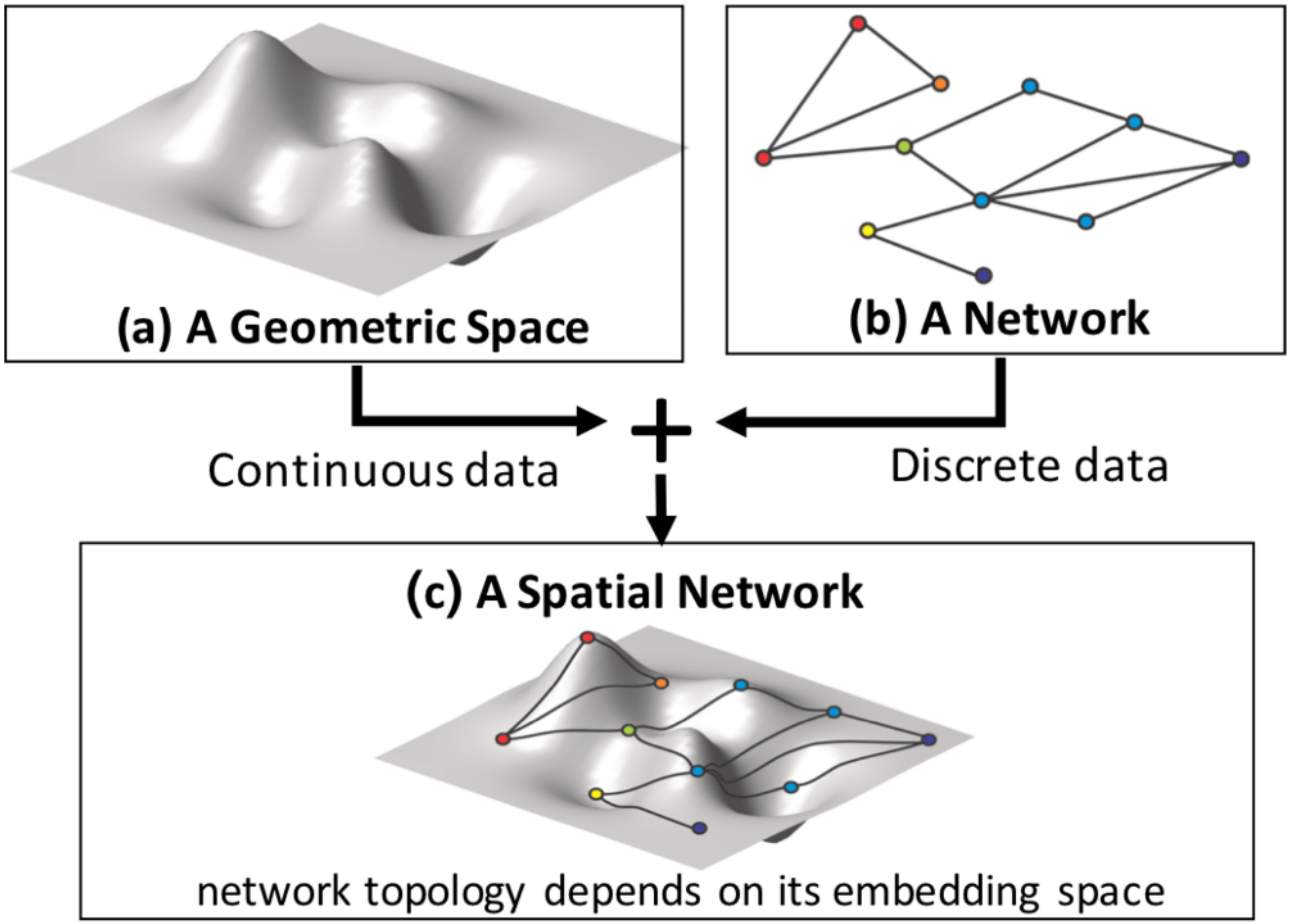


Dudziak, Lukasz, et al. **"Brp-nas: Prediction-based nas using gcns."** *Advances in Neural Information Processing Systems 33 (2020): 10480-10490.*



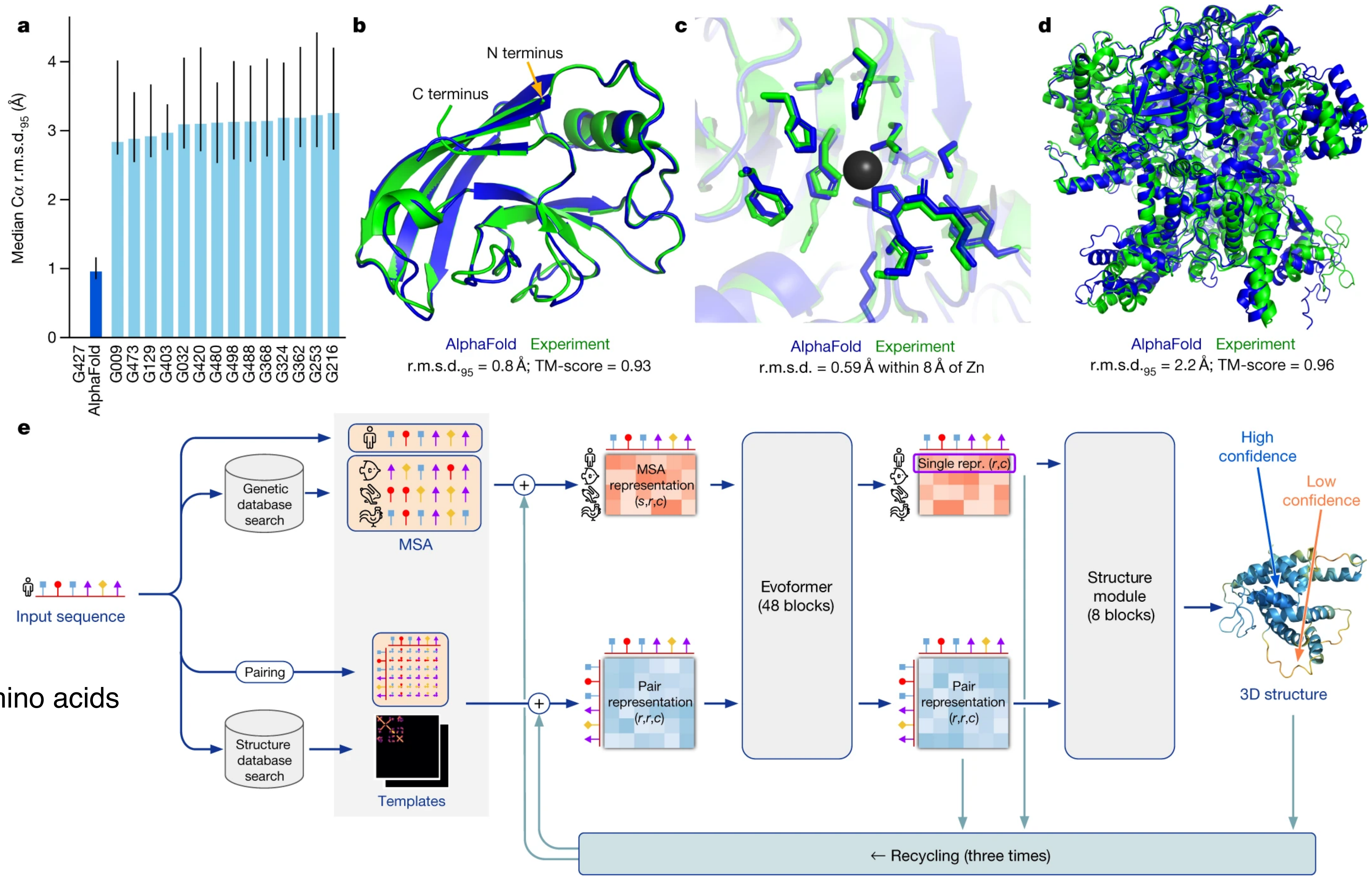
Zhang, C., et al,. ***Graph hypernetworks for neural architecture search.*** *ICLR 2019*

Graphs in Spatial Data



Zhang, Zheng, and Liang Zhao. "Representation learning on spatial networks." Advances in Neural Information Processing Systems 34 (2021): 2303-2318.

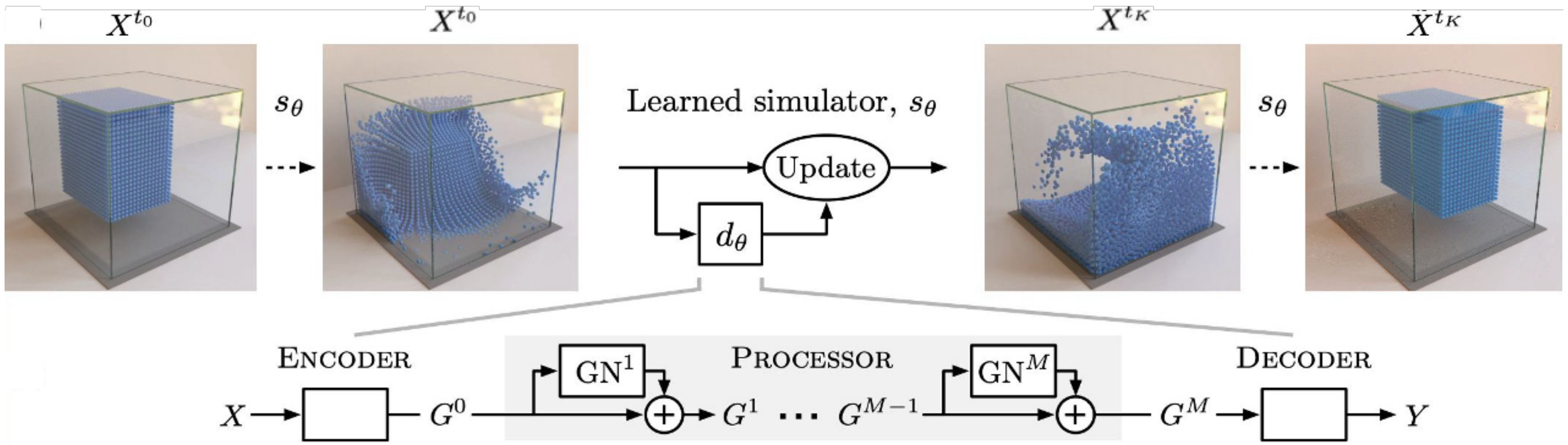
Graphs in Protein Folding

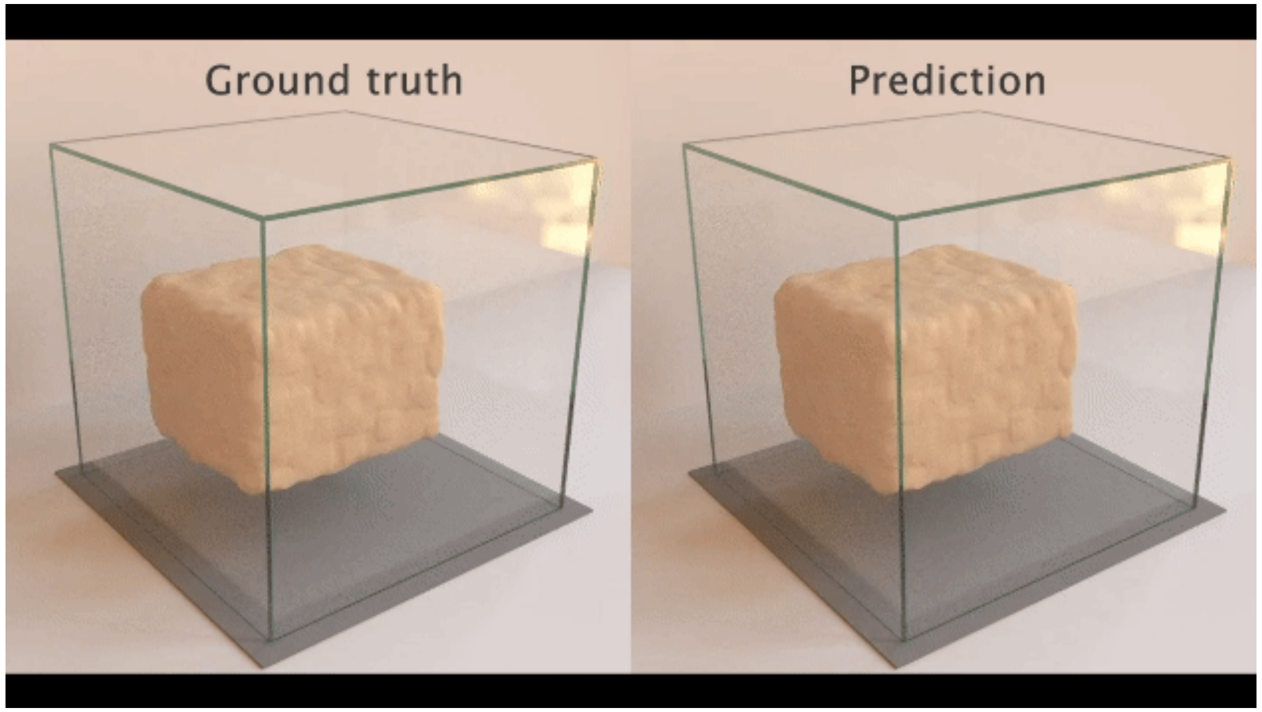


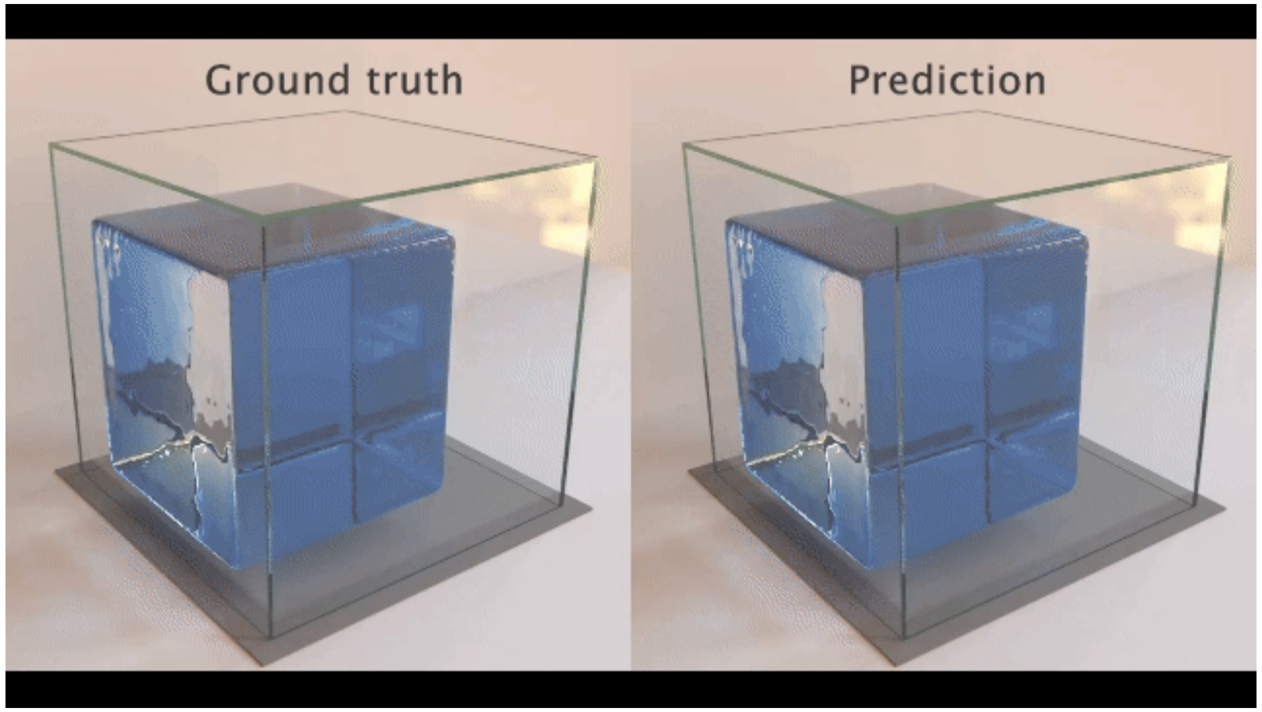
Amino acids

Jumper, John, et al. "Highly accurate protein structure prediction with AlphaFold." nature 596.7873 (2021): 583-589.

Graphs in Physics







Sanchez-Gonzalez, Alvaro, et al. "Learning to simulate complex physics with graph networks." International conference on machine learning. PMLR, 2020.

Motivation: A Unified View

A large number of graph neural networks, with different mechanisms

S Spatial

S Spectral

Spectral Graph S

Message Passing S

Heat Diffusion S

Page Rank S

Random Walk S

Graph Wavelet S

Attention Model S

Low-pass Filter S

DSF/BSF S

ARMA filter S

Weisfeiler-Lehman S

Laplacian Smooth S

...

Motivation: A Unified View

A large number of graph neural networks, with different mechanisms

S **Spatial**

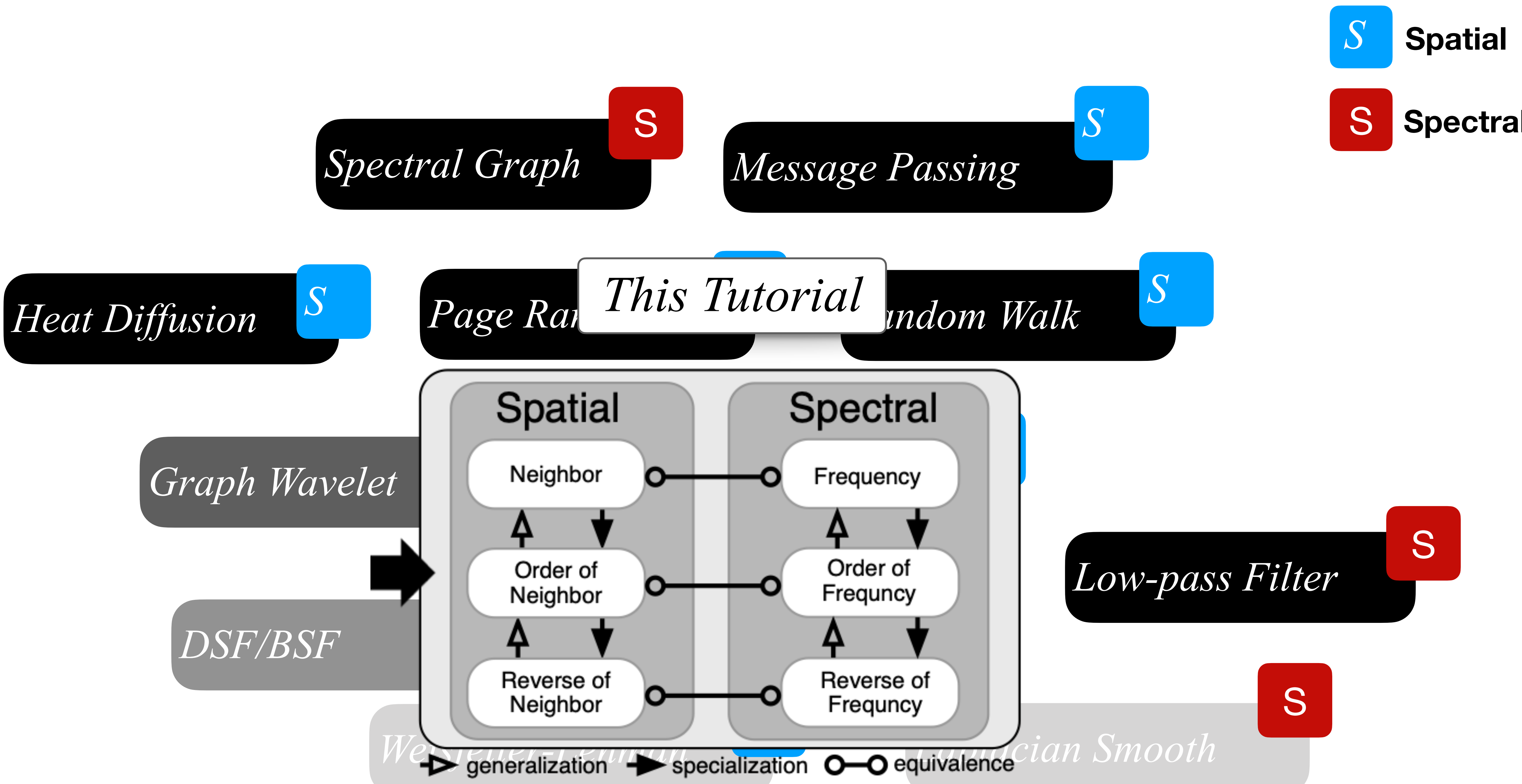
S **Spectral**

- *Spectral Graph* S
- *Message Passing* S
- *Heat Diffusion* S
- *Page Rank* S
- *Random Walk* S
- *Graph Wavelet* S
- *Attention Model* S
- *Low-pass Filter* S
- *DSF/BSF* S
- *ARMA filter* S
- *Weisfeiler-Lehman* S
- *Laplacian Smooth* S

…

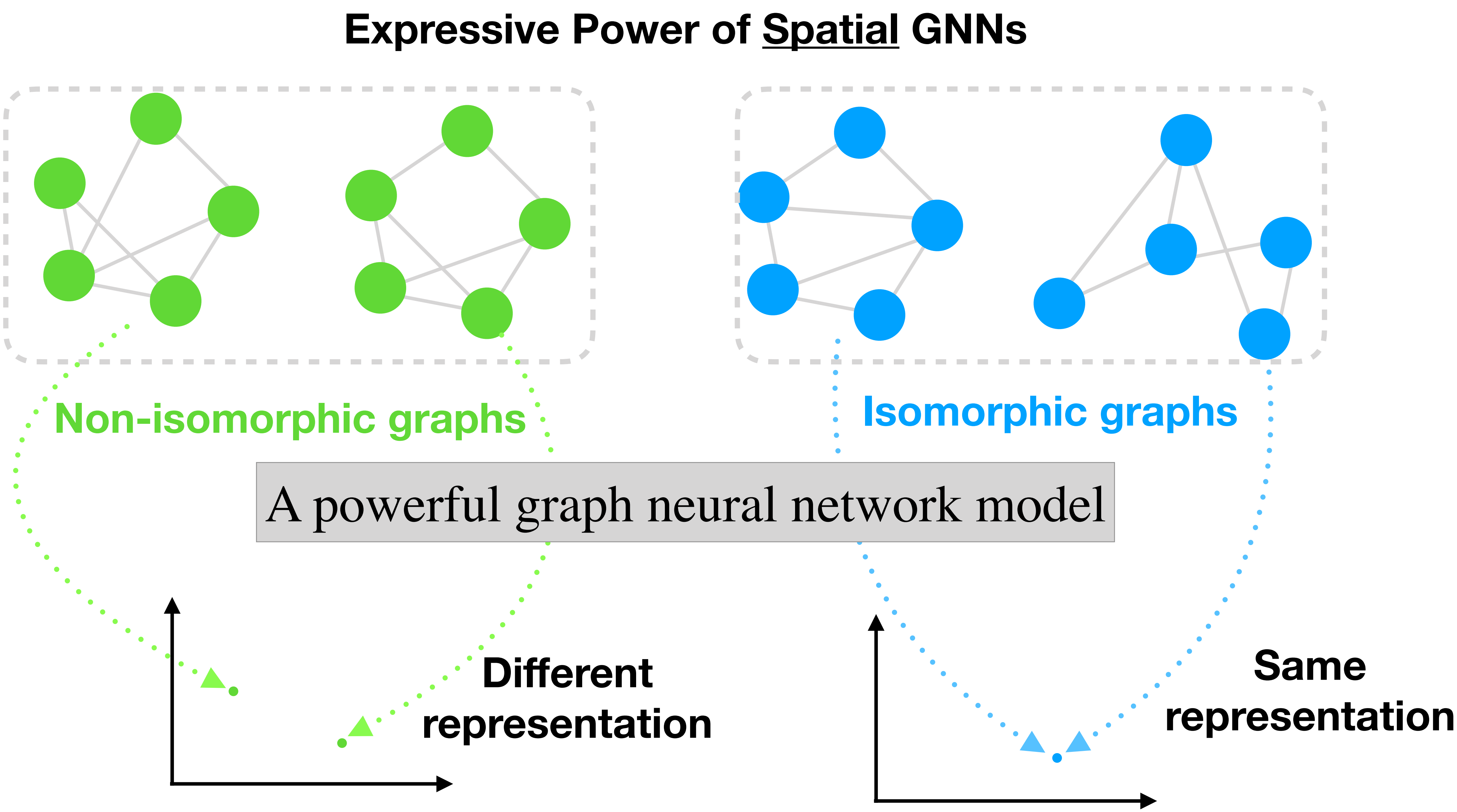
Motivation: A Unified View

A large number of graph neural networks, with different mechanisms



...

Attempts to Unify GNNs



Compare GNNs with respect to their ability to distinguish non-isomorphic graph structures

Attempts to Unify GNNs

Expressive Power of Spectral GNNs

Universal Approximation Theorem?

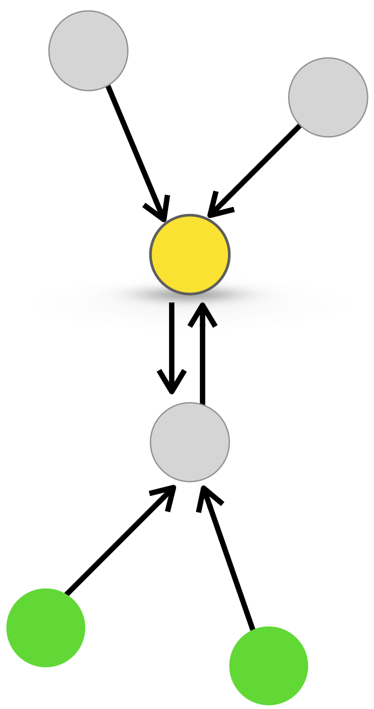
For MLP: A feedforward neural network (such as an MLP) with at least one hidden layer and a sufficient number of neurons can approximate any continuous function on a compact subset of R^n, to any desired degree of accuracy, provided the activation function is non-linear (e.g., sigmoid, ReLU).

Approximating a target function f^* with a parameterized function f_θ

$$f_\theta \to f^*$$

How about in graphs?

Expressive Power of Spatial GNNs



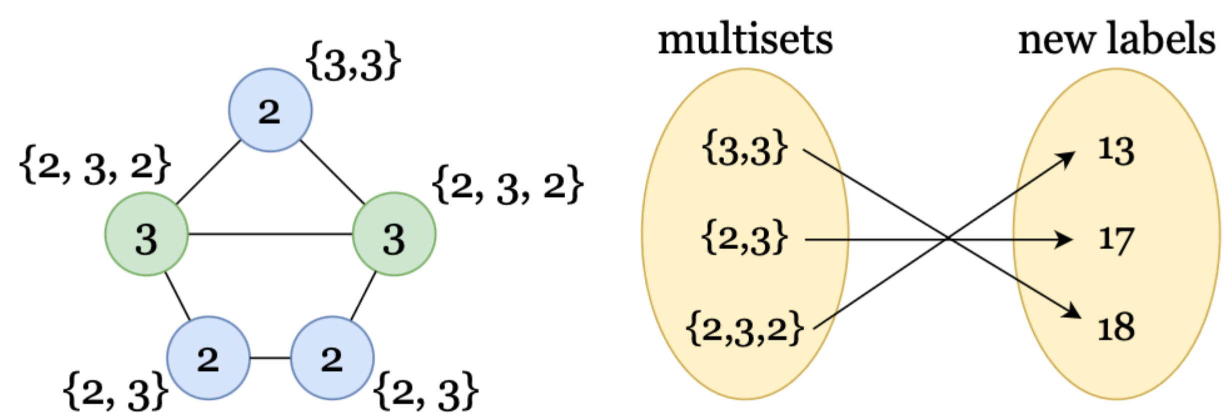
GNNs are defined as a composition of

- AGGREGATE functions and
- READOUT functions

$$h_u^{(k)} = \boxed{\text{AGGREGATE}}^{(k)}\left(\left\{\left(h_v^{(k-1)}, h_u^{(k-1)}\right)\right\} \mid v \in \mathcal{N}(u)\right)$$

$$h_G = \boxed{\text{READOUT}}\left(\left\{h_u^{(K)}\right\} \mid u \in V\right)$$

- *GNNs are at most as powerful as a **Weisfeiler-Lehman** graph isomorphism test.*
- *This upper bound is achieved if **AGGREGATE** and **READOUT** are **Injective Multiset Functions***



Every possible output has at most one associated input

Xu, K. et al., ***How powerful are graph neural networks?*** *In International Conference on Learning Representations, 2019.*

Expressive Power of Spectral GNNs

GNNs are defined as a simplified polynomial Spectral GNN:

$$Z = g(\hat{L})XW$$

Such GNNs can produce any-one-dimensional predictions if :

- ***No multiple eigenvalue***
  - *Each frequency component of $\hat{L}$ has a different frequency*
- ***No missing Component***
  - *No frequency component is missing from X*

Wang, Xiyuan, and Muhan Zhang. "**How powerful are spectral graph neural networks.**" International conference on machine learning. PMLR, 2022.

Expressive Power

Spatial

Spectral

Definition: the ability to distinguish different **graph structures**

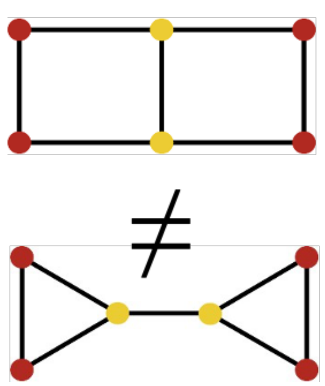
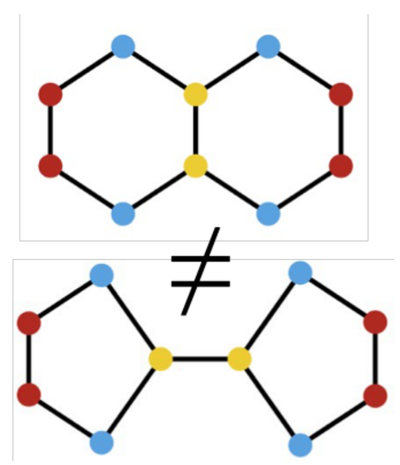
Definition: the universality on **node features**

Scope: MP-GNNs

Scope: Simplified Spectral GNNs

Expressive Power

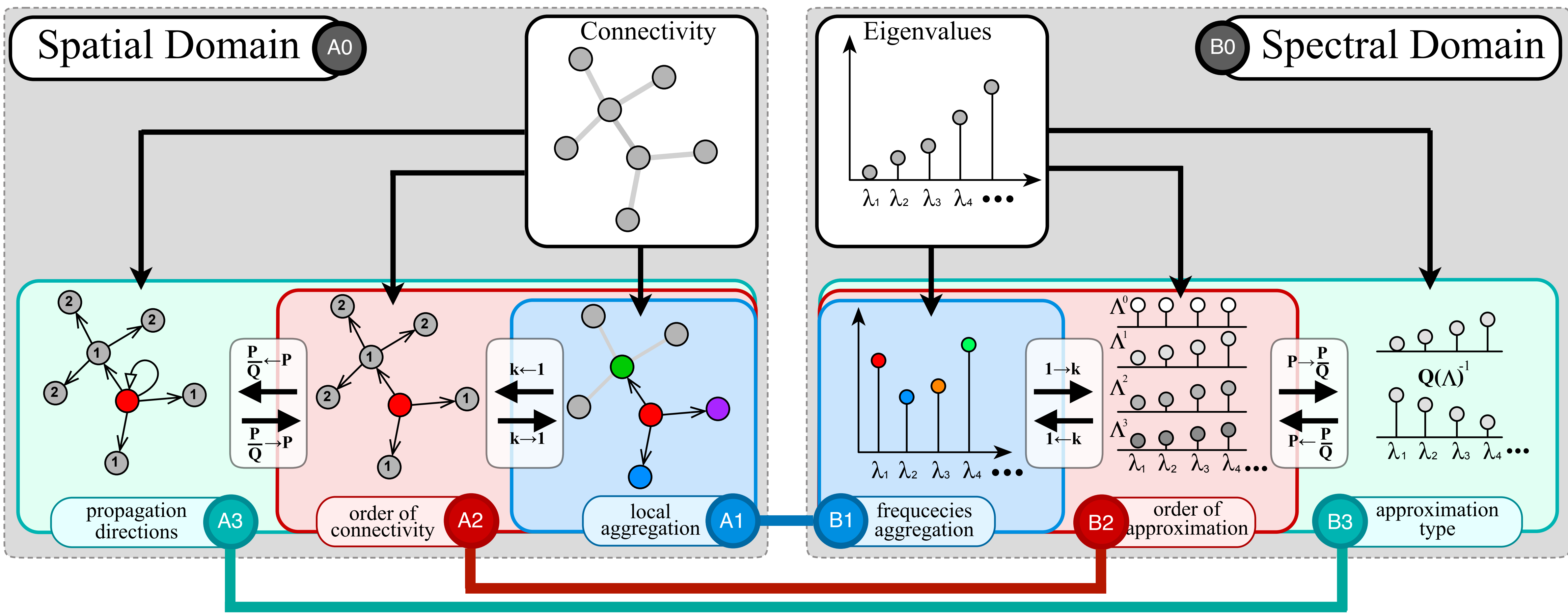
Spatial		*Spectral*
• **Upper bound: 1-WL test**	Progressive	• **Universal approximator**

• **What 1-WL cannot distinguish:**  	Compatible	• **No multiple eigenvalue**
		  • A Graph has low symmetry
		  • Order of graph's automorphism < 3

• **Does not consider node features**	Difference	• **No missing component** • Nodes have rich features

Comparison

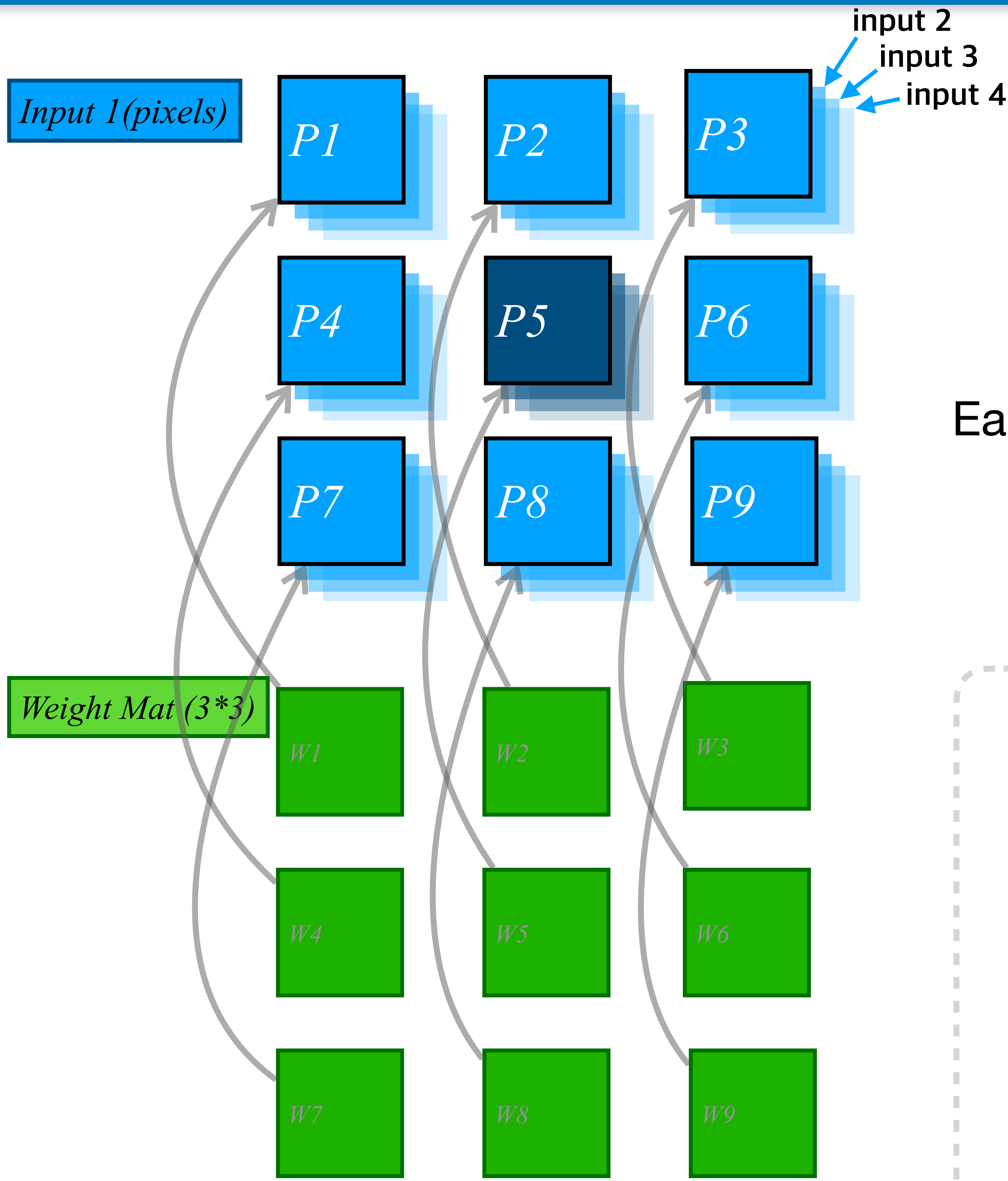
- Existing surveys and theoretical analyses focus on either the spatial or the spectral GNNs, not all of them.
- Here is the framework we will introduce later



Agenda

- **First Half (1 hour 30 min)**
  - *Background*: **unified frameworks for GNN** (45 min)
  - *Preliminary*: **graph convolutions** (45 min)
  - BREAK (30 min)
- **Second Half (1 hour)**
  - *Introduction*: **a new unified framework** (40 min)
  - *Future directions* (20 min)
  - Q&A (15 min)

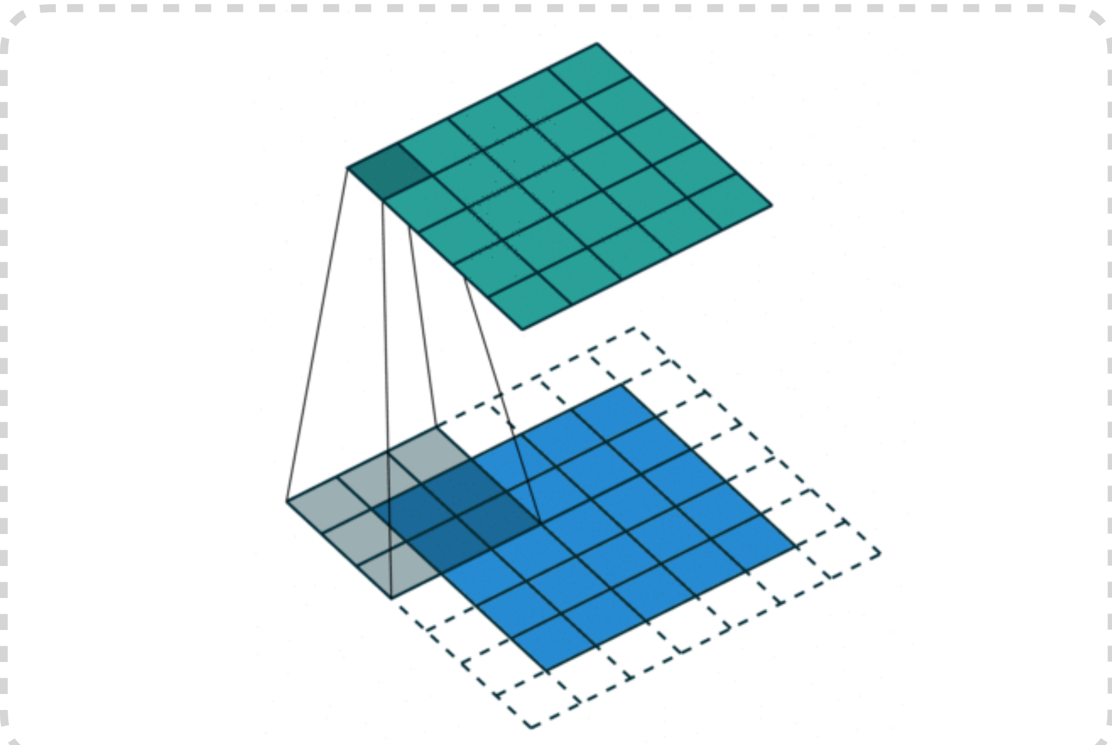
ConvNet



$$\hat{P}_5 = \sum_{i=1}^{9} P_i \cdot W_i$$

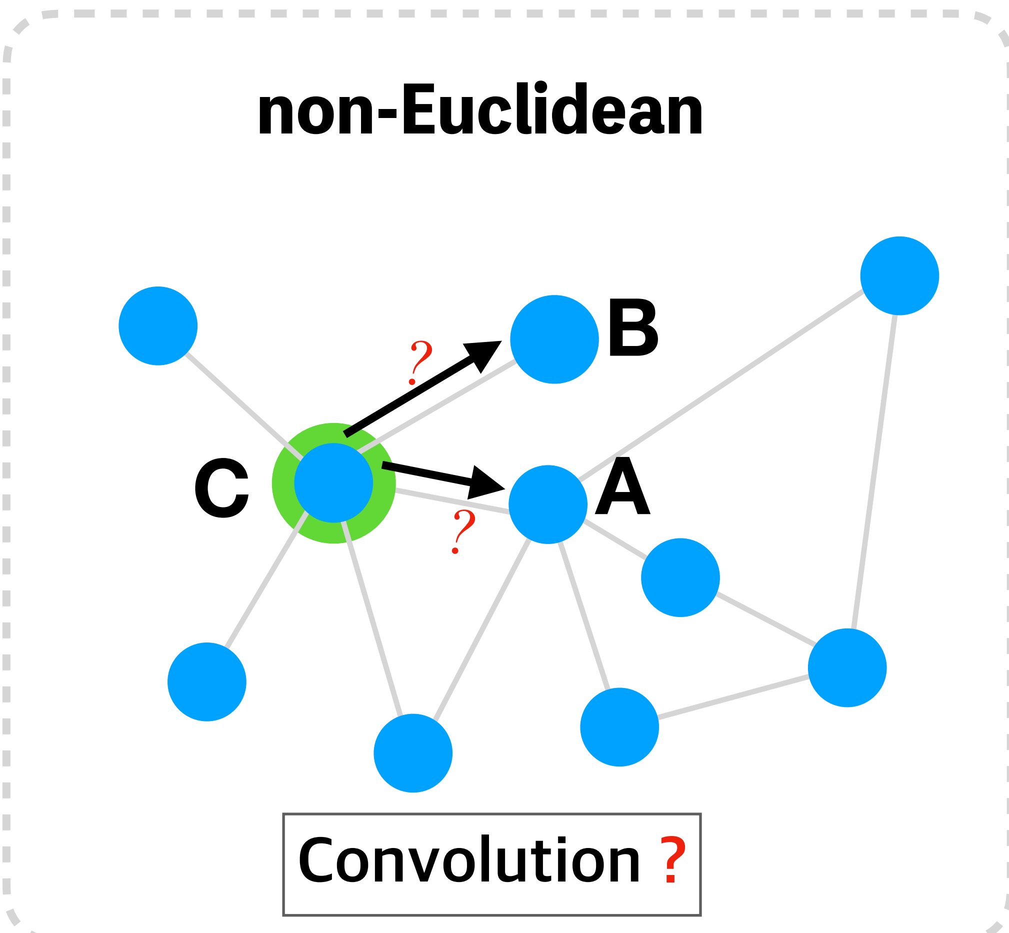
Each pixel has **fixed** number of neighbors

Convolution

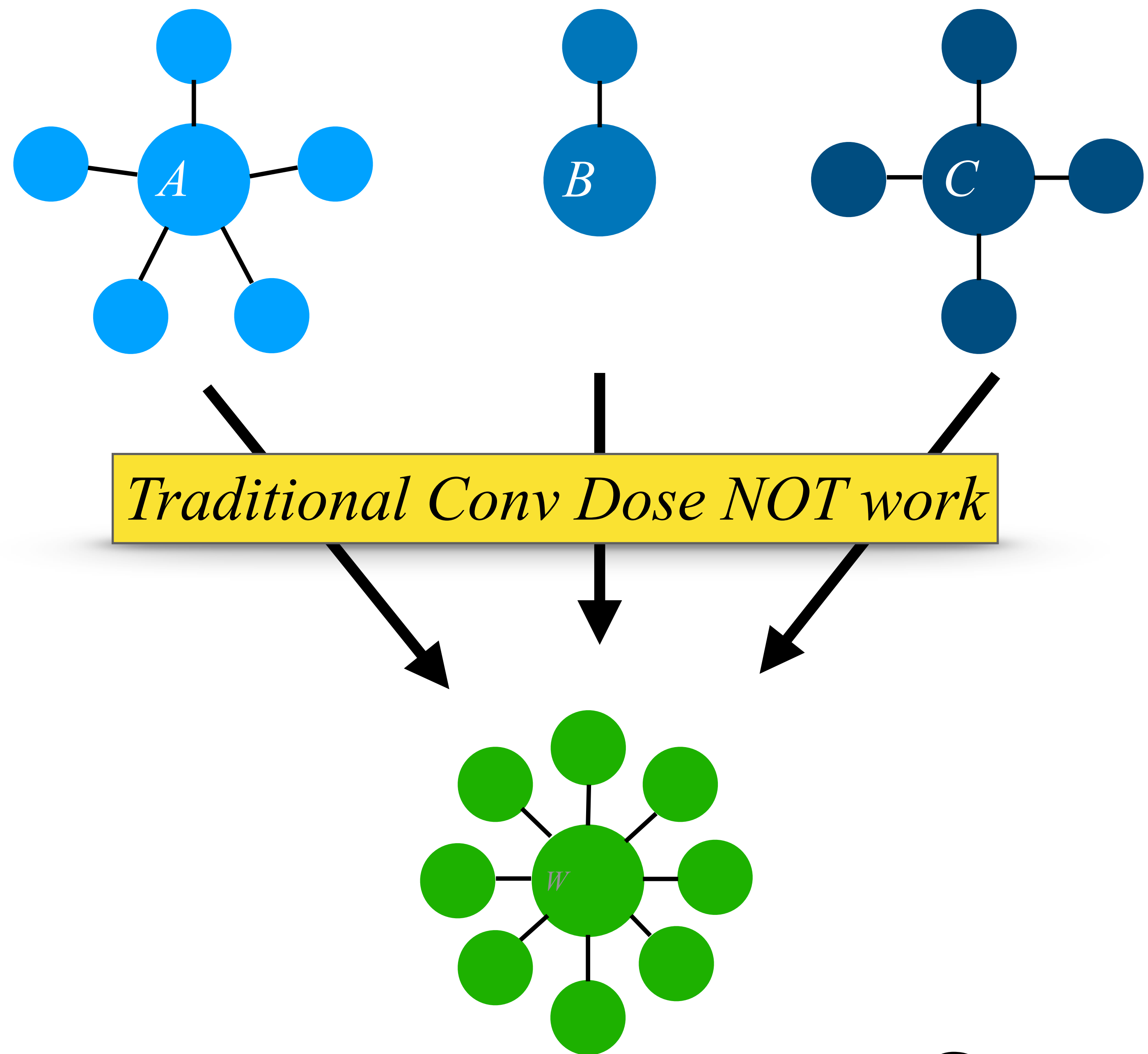


Credit: Prof. Dr. Johannes Maucher

Challenge for ConvNet on Graphs



dynamic # of neighbors



fixed # of weights ?

What is Graph Convolution

- **Convolution Theorem**
  - Fourier transform of the convolution of two functions is equal to the point-wise multiplication of their Fourier transforms.

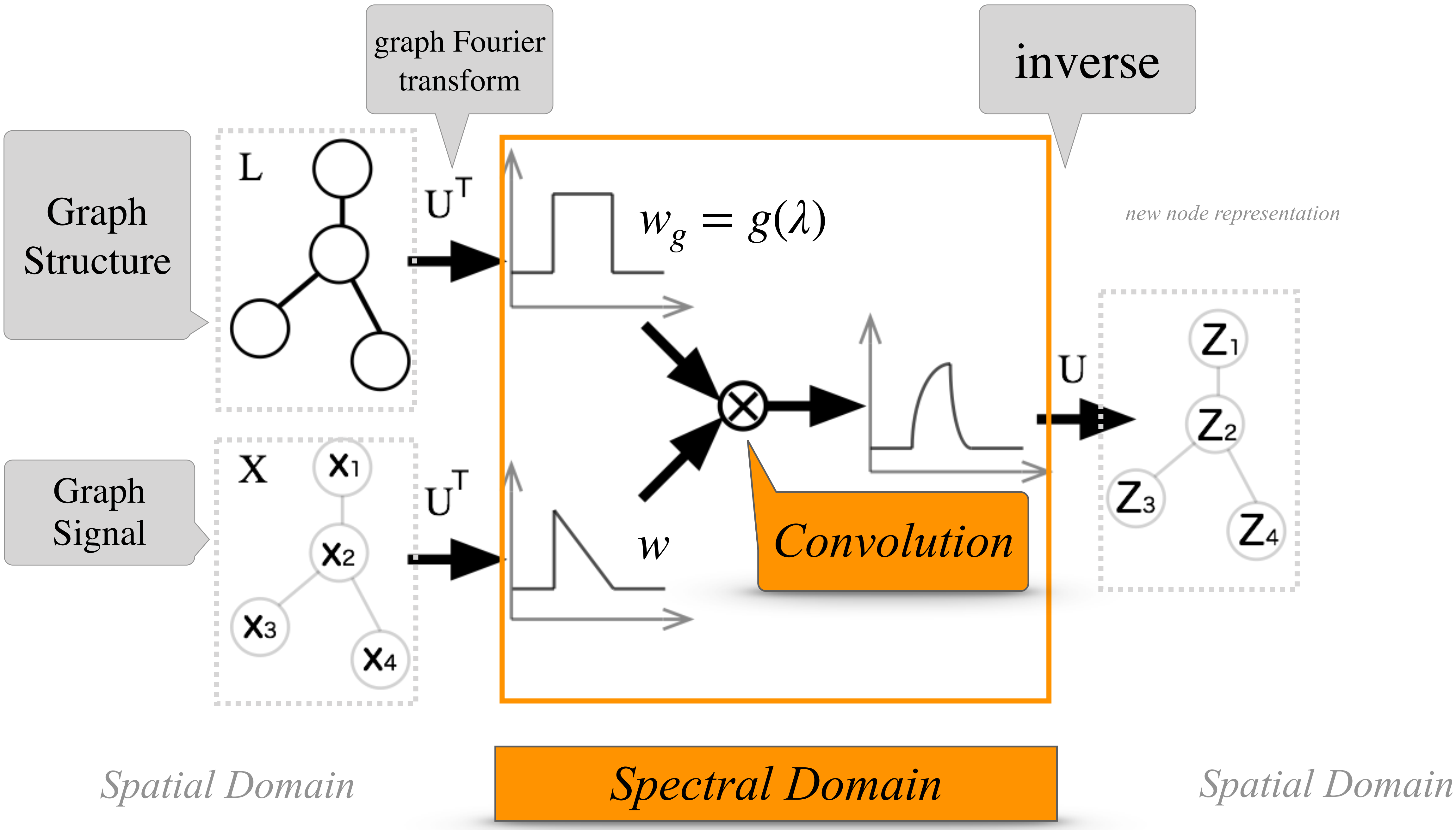
$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$$

$\boxed{\textbf{Space convolution = frequency multiplication}}$

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

We can do the convolution in the spectral domain, such that avoiding the issues.

Convolution on Graph Data



$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

*Chung, Fan RK. **Spectral graph theory**. Vol. 92. American Mathematical Soc., 1997.*

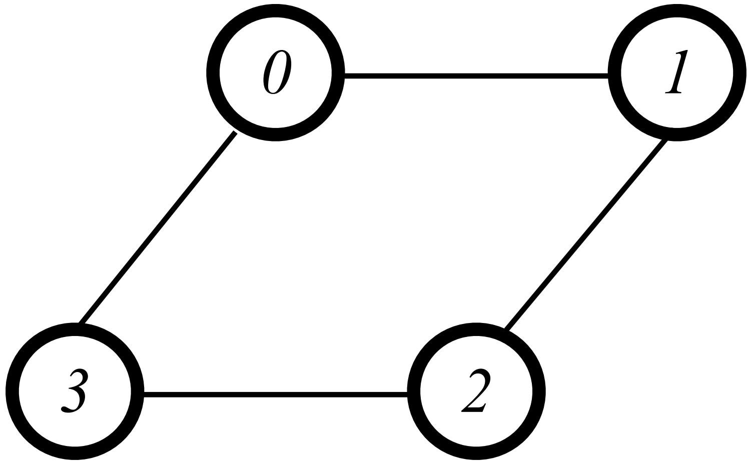
Convolution on Graph Data

Convolution theorem: $f * g = \mathscr{F}^{-1}\{\mathscr{F}\{f\} \cdot \mathscr{F}\{g\}\}$

What is the Fourier Transform on Graphs?

What is the Inverse Fourier Transform on Graphs?

Graph Fourier Transform



$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 2 & & & \\ & 2 & & \\ & & 2 & \\ & & & 2 \end{bmatrix}$$

Graph Laplacian

$$L = D - A$$

Normalized Laplacian

$$\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

$$\hat{L} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & -\frac{1}{2} & 1 \end{bmatrix}$$

Example credit: Dr. Muhan Zhang, Peking University

Graph Fourier Transform

Eigen-decomposition

$$\hat{L} = U \Lambda U^{\top}$$

$$U = \begin{bmatrix} -\frac{1}{2} & 0 & -\frac{\sqrt{2}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{\sqrt{2}}{2} & 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{\sqrt{2}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{2}}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & 2 \end{bmatrix}$$

Eigenvalue Λ $\leftrightarrow$ Frequency

Eigenvector U $\leftrightarrow$ Frequency Component

Orthonormal Basis of a Graph

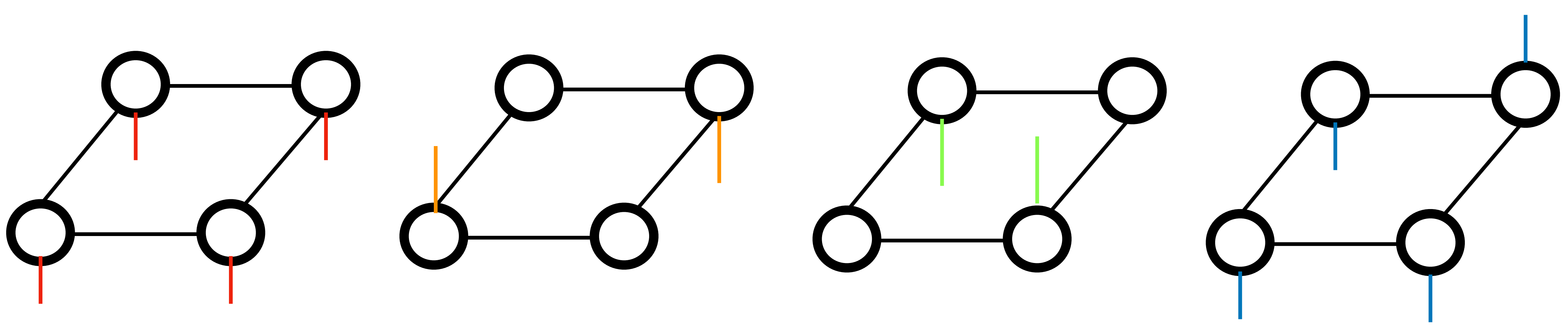
Eigenvector U $\leftrightarrow$ Frequency Component

$$U = \begin{bmatrix} -\frac{1}{2} & 0 & -\frac{\sqrt{2}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{\sqrt{2}}{2} & 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{\sqrt{2}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{2}}{2} & 0 & -\frac{1}{2} \end{bmatrix}$$

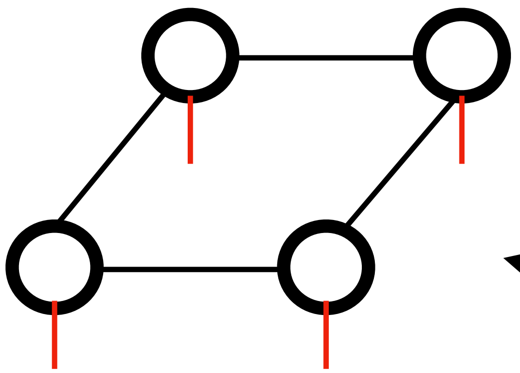
They form an **Orthonormal basis** for the vector space of graph signals

- unit vectors
- orthogonal to each other

$\mathbf{v}_i^T \mathbf{v}_j = 0 \quad$ for $i \neq j$



Frequencies of a Graph



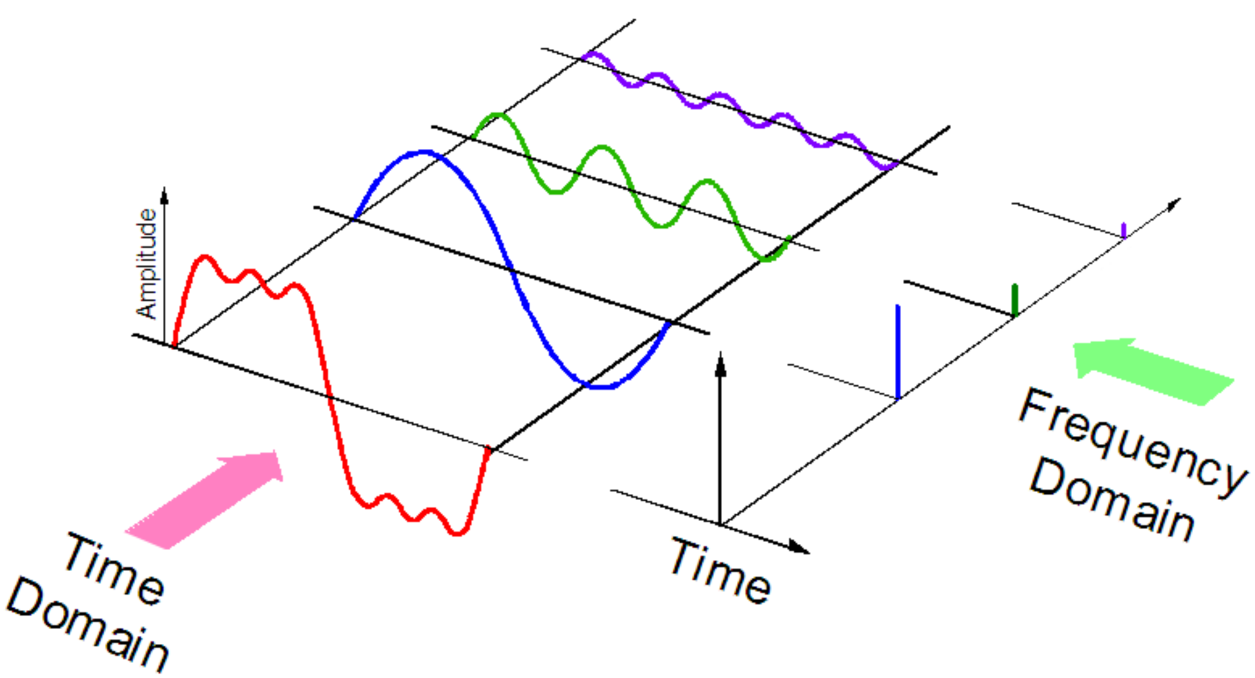
$$\Lambda = \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & 2 \end{bmatrix}$$

Eigenvalue Λ $\leftrightarrow$ Frequency

For graph,

low frequency ↔ smooth signal

- 0's frequency components are all the same (does not vary)



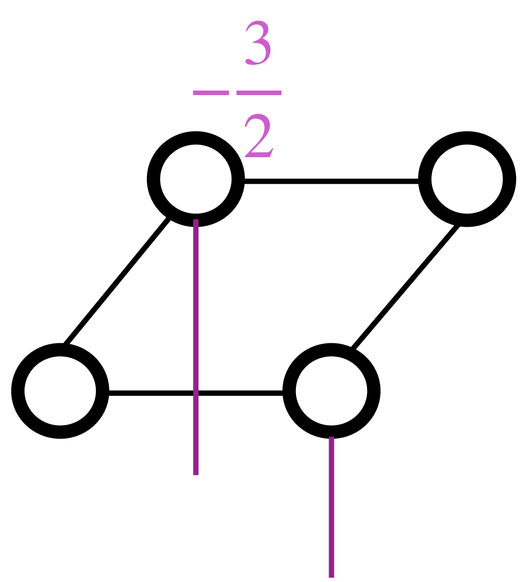
For sound waves,

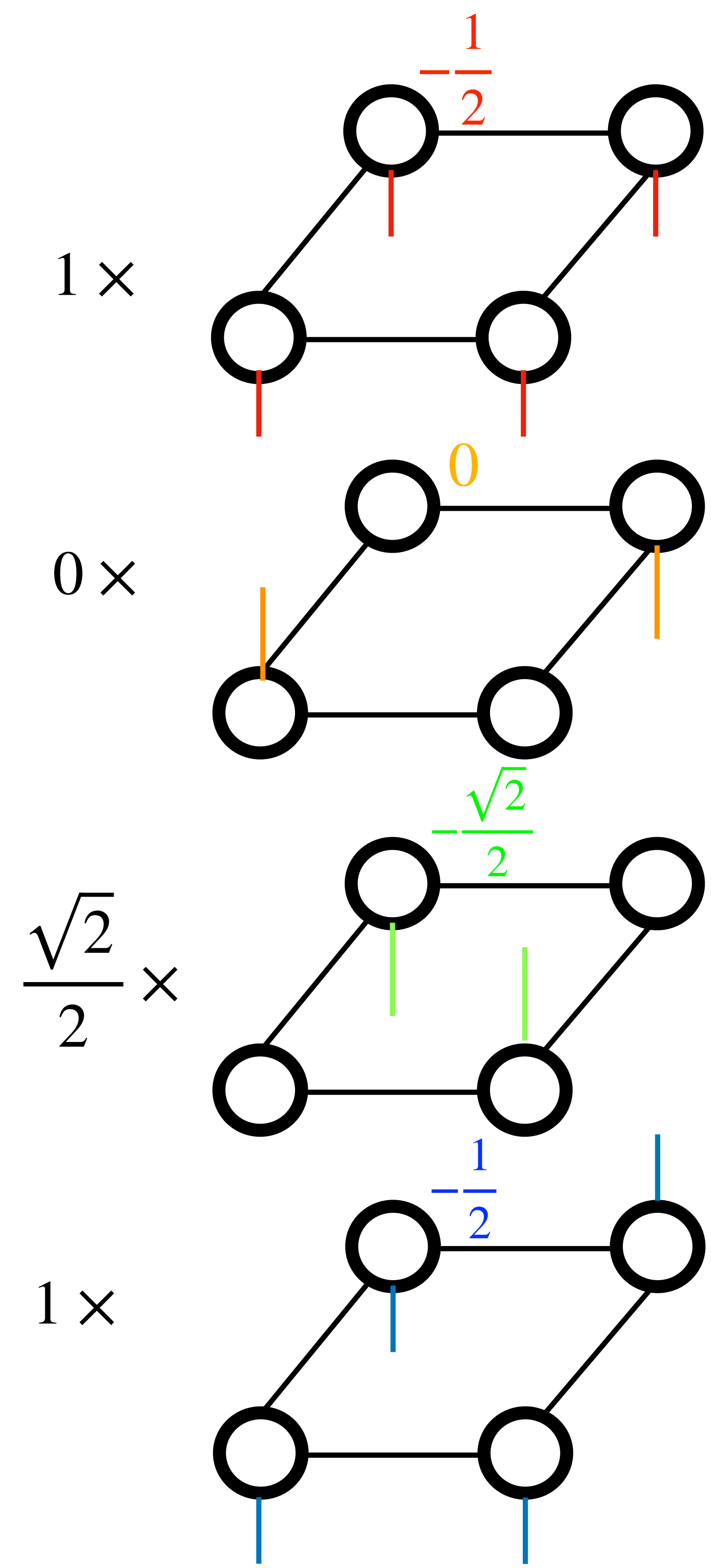
low frequency ↔ slowly varying signal

Graph Fourier Transform

$$-\frac{3}{2} = \left(1 \times -\frac{1}{2}\right) + (0 \times 0) + \left(\frac{\sqrt{2}}{2} \times -\frac{\sqrt{2}}{2}\right) + \left(1 \times -\frac{1}{2}\right)$$

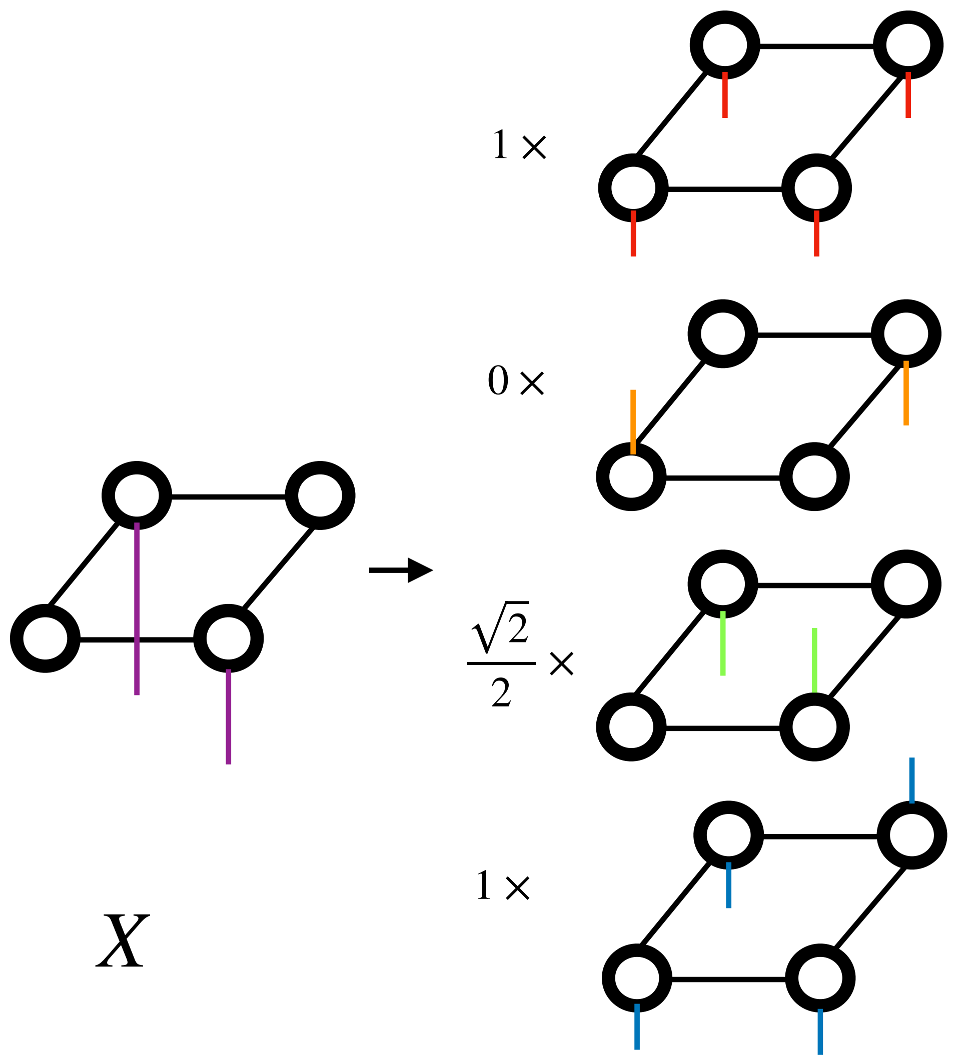
$$X = \begin{bmatrix} -\frac{3}{2} \\ 0 \\ -\frac{1}{2} \\ 0 \end{bmatrix}$$





Graph Fourier Transform

Graph Domain | **Frequency Domain**



X

Graph Domain | **Frequency Domain**

$$\begin{bmatrix} -\frac{3}{2} \\ 0 \\ -\frac{1}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & 0 & -\frac{\sqrt{2}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{\sqrt{2}}{2} & 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{\sqrt{2}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{2}}{2} & 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -\frac{\sqrt{2}}{2} \\ 1 \end{bmatrix}$$

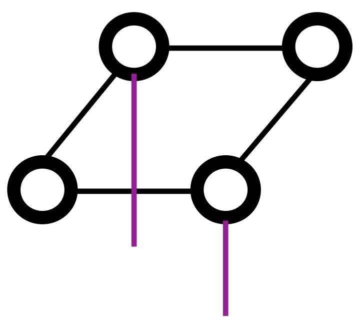
X $\quad$ U $\quad$ $\tilde{X}$

$$X = U\tilde{X}$$

Graph Fourier Transform

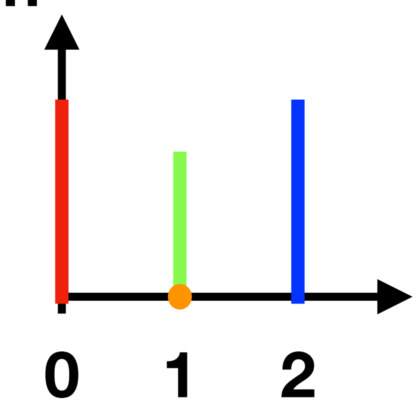
Graph Domain

X



$$\longrightarrow \begin{bmatrix} 1 \\ 0 \\ -\frac{\sqrt{2}}{2} \\ 1 \end{bmatrix}$$

Frequency Domain



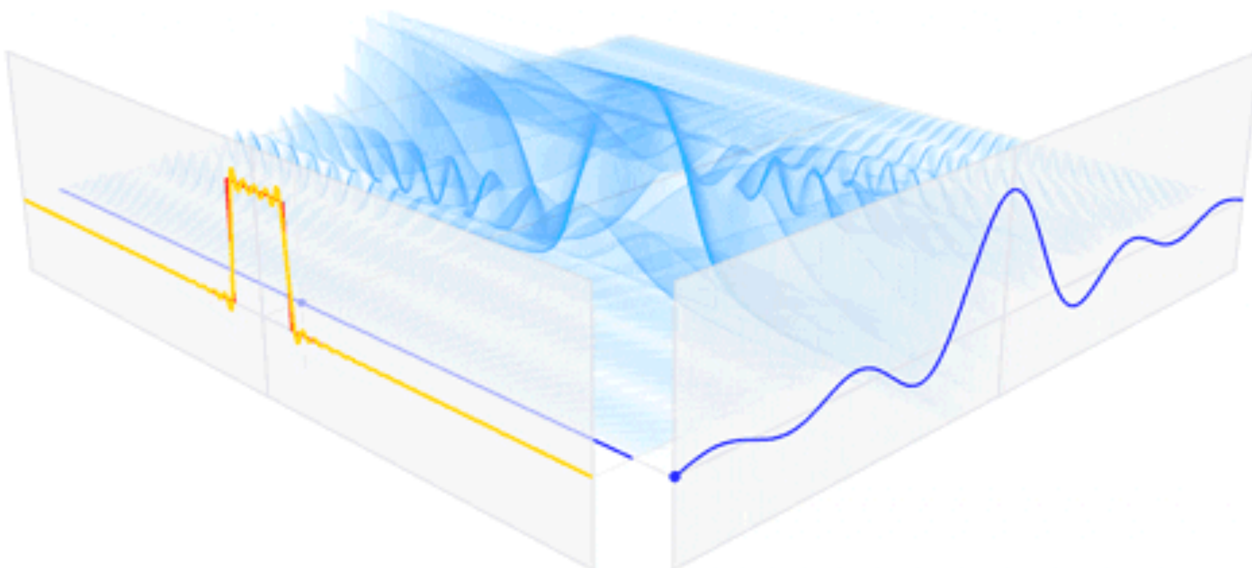
$$\Lambda = \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & 2 \end{bmatrix}$$

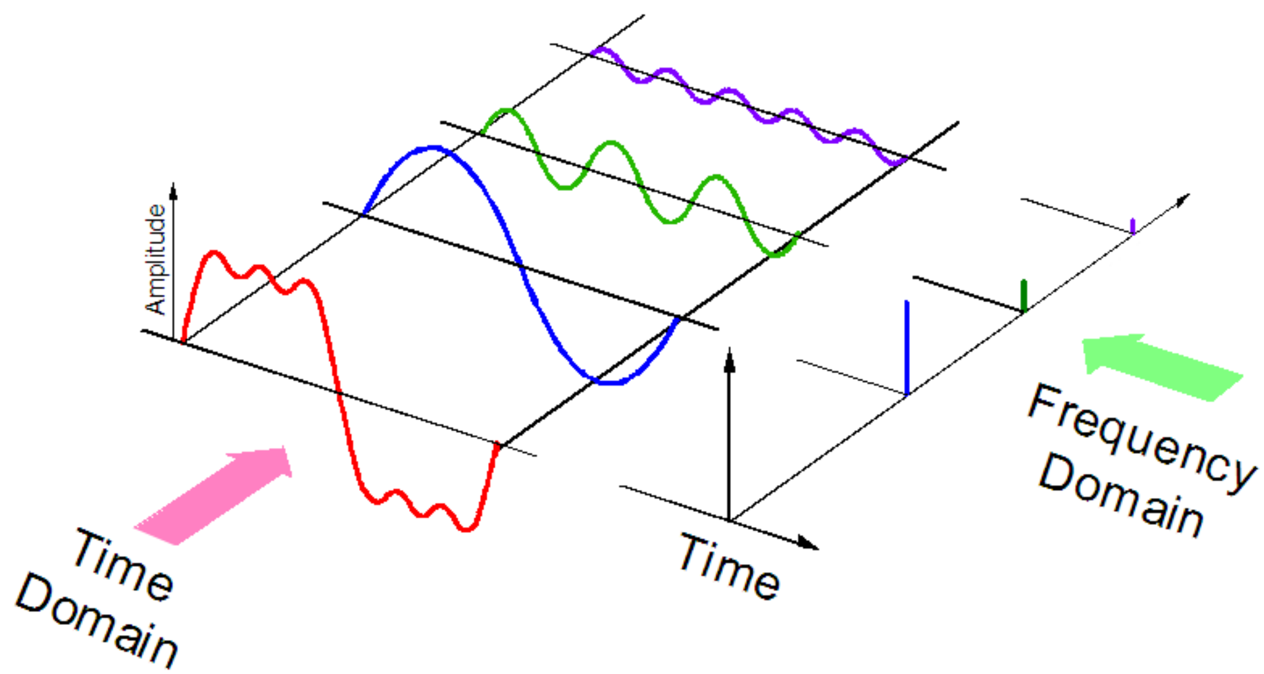
Graph Fourier transform: $\tilde{X} = U^T X$ **(U is an orthogonal matrix)**

Inverse graph Fourier transform: $X = U\tilde{X}$

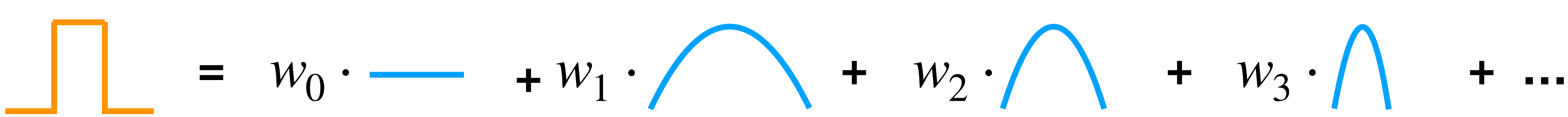
Spectral Analysis

$$\int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx$$

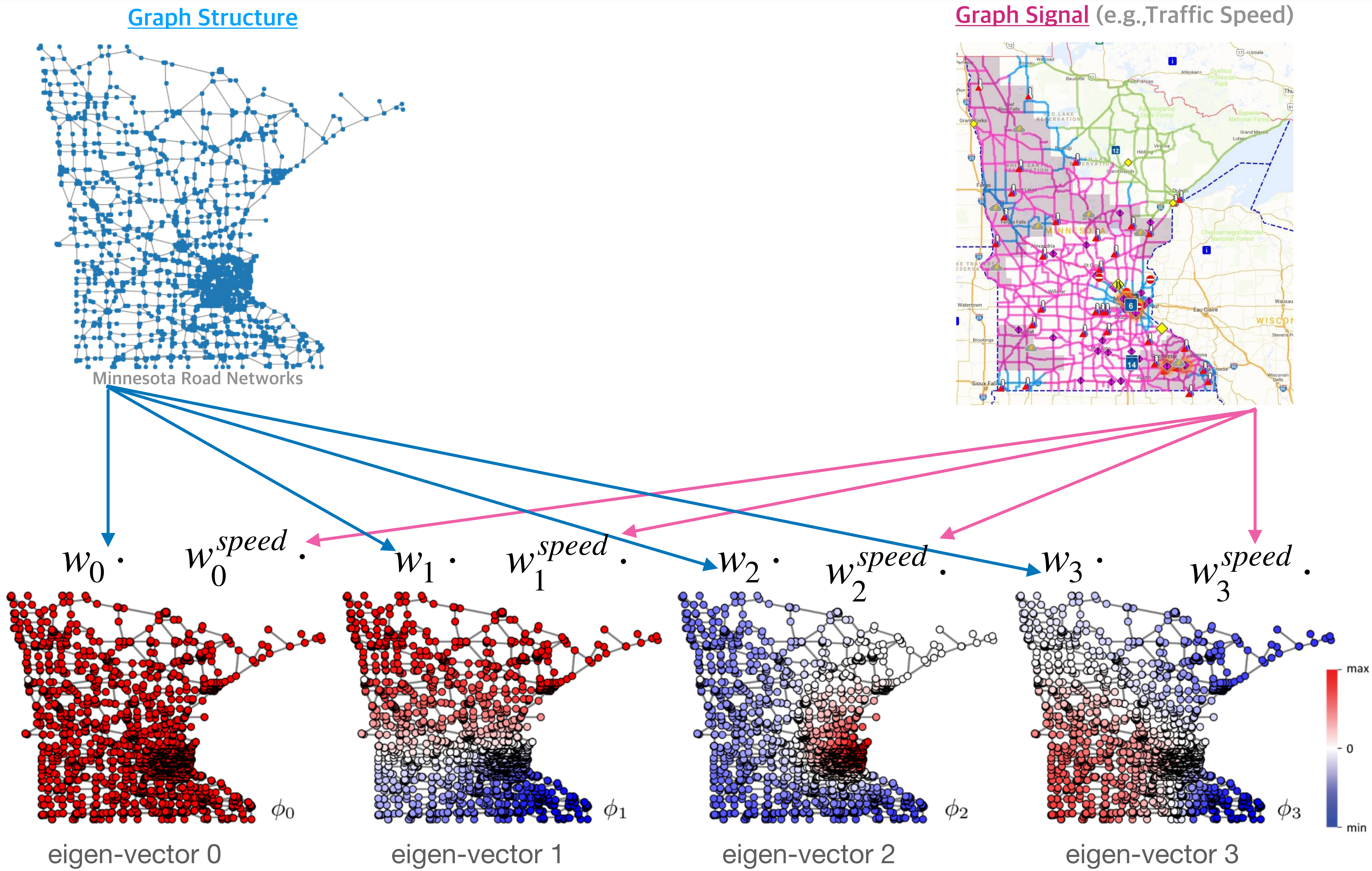




credit: giphy



Spectral Analysis for Graph



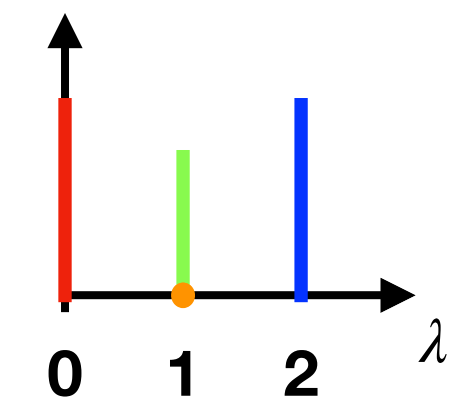
Graph Fourier Transform (Spectral Decomposition)

Convolution on Graph Data

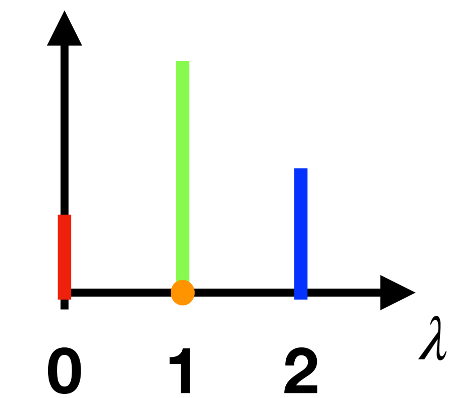
- Convolution theorem: $f * g = \mathscr{F}^{-1}\{\mathscr{F}\{f\} \cdot \mathscr{F}\{g\}\}$
- What is each of the components **on a graph**?
- Fourier transform of f: $U^T f$
- Inverse Fourier transform of f: Uf
- $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$
- Finally, $g * X = U(U^T g \cdot U^T X) = U g(\Lambda) U^T X$

Convolution on Graph Data

$$\hat{L} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & -\frac{1}{2} & 1 \end{bmatrix} \qquad \begin{bmatrix} -\frac{3}{2} \\ 0 \\ -\frac{1}{2} \\ 0 \end{bmatrix}$$



1 graph Fourier transform



$$\boxed{g_\theta} * \boxed{x} = \boxed{U\boxed{g_\theta(\Lambda)\boxed{U^\top x}}}$$

2 convolution (theorem)

3 Inverse graph Fourier transform

Chung, Fan RK. ***Spectral graph theory***. *Vol. 92. American Mathematical Soc., 1997.*

Convolution on Graph Data

$$g_\theta * x = U g_\theta(\Lambda) U^\top x$$

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$$

How to design g_θ ?

$$g_\theta(\Lambda) = \begin{pmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_n \end{pmatrix}$$

This is the weight matrix/ convolution kernel / mask in CNN

Graph Convolutional Neural Network

Ver 1.0

$$g_\theta * x = U \begin{pmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_n \end{pmatrix} (\Lambda) U^\top x$$

It works, but it's expensive to calculate.
We can approximate it with the polynomial of Λ

Graph Convolutional Neural Network

Ver 2.0

$$\begin{pmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_n \end{pmatrix}(\Lambda)$$

$$\downarrow$$

$$g_\theta(\Lambda) \approx \sum_{k=0}^{K-1} \theta_k \Lambda^k$$

Polynomial of Λ (k free parameters, $k < |N|$)

Graph Convolutional Neural Network

Ver 2.0

$$y = \sigma(\mathbf{U} g_\theta(\Lambda) \mathbf{U}^T x) = \sigma(\mathbf{U} \sum_{k=0}^{K-1} \theta_k \Lambda^k \mathbf{U} x) = \sigma(\sum_{k=0}^{K-1} \theta_k L^k x)$$

$$L^2 = U\Lambda U^T U \Lambda U^T = U \Lambda^2 U^T \qquad U^T U = I$$

Benefit: No more eigen-decomposition

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." Advances in neural information processing systems 29 (2016).

Graph Convolutional Neural Network

Ver 2.0

$$y = \sigma\left(Ug_\theta(\Lambda)U^T x\right) = \sigma\left(U\sum_{k=0}^{K-1}\theta_k\Lambda^k Ux\right) = \sigma\left(\sum_{k=0}^{K-1}\theta_k L^k x\right)$$

- The matrix power is still expensive
- How **Not** to calculate L^k?

ChebNet

Chebyshev polynomial

$$g_\theta(\Lambda) \approx \sum_{k=0}^{K-1} \theta_k \Lambda^k \longrightarrow g_\theta(\Lambda) \approx \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{\Lambda})$$

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), ; T_0(x) = 1, ; T_1(x) = x 。$$

- The Chebyshev polynomial can be calculated recursively, so we can avoid calculating L^k
- k-localized filter: Consider the neighbors up to a distance of k hops in the graph

The "GCN"

$$g_\theta * x \approx \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{L})x$$

When k is 2, and

$$g_\theta * x \approx \theta_0 x + \theta_1 (L - I_N)x = \theta_0 x - \theta_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x$$

With only two free parameters. How about fewer?

$$g_\theta \star x \approx \theta \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x$$

The **"GCN"**

Thomas N. Kipf, Max Welling, Semi-Supervised Classification with Graph Convolutional Networks (ICLR 2017)

Spectral View of GCN

$$g_\theta * f = Ug_\theta U^\top f \quad = g_\theta(L) \cdot f$$

$$= g_\theta(L) \cdot f$$

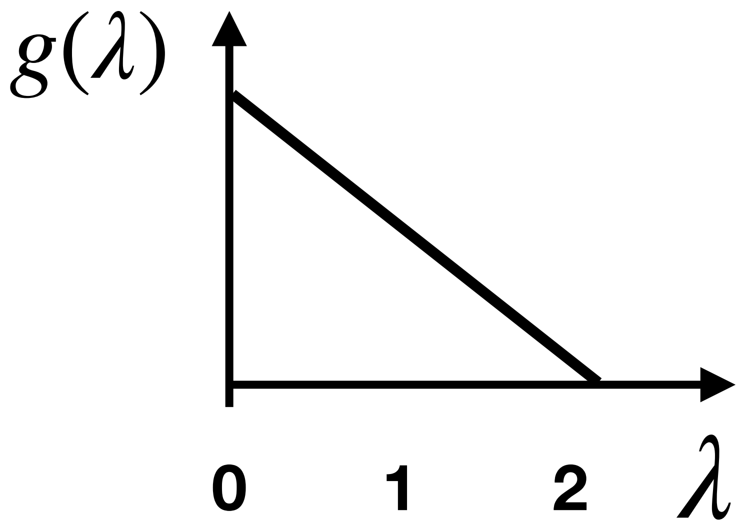
$$= (2 \cdot I - L) \cdot f$$

$$= (2 \cdot I - (I - A)) \cdot f$$

$g_\lambda = g(\lambda) = 2 - \lambda$

1-order polynomial without weights

A and L are normalized



All eigenvalues of the **normalized** symmetric Laplacian satisfy:

$$0 < \lambda_0 \leq \ldots \leq \lambda_{n-1} \leq 2$$

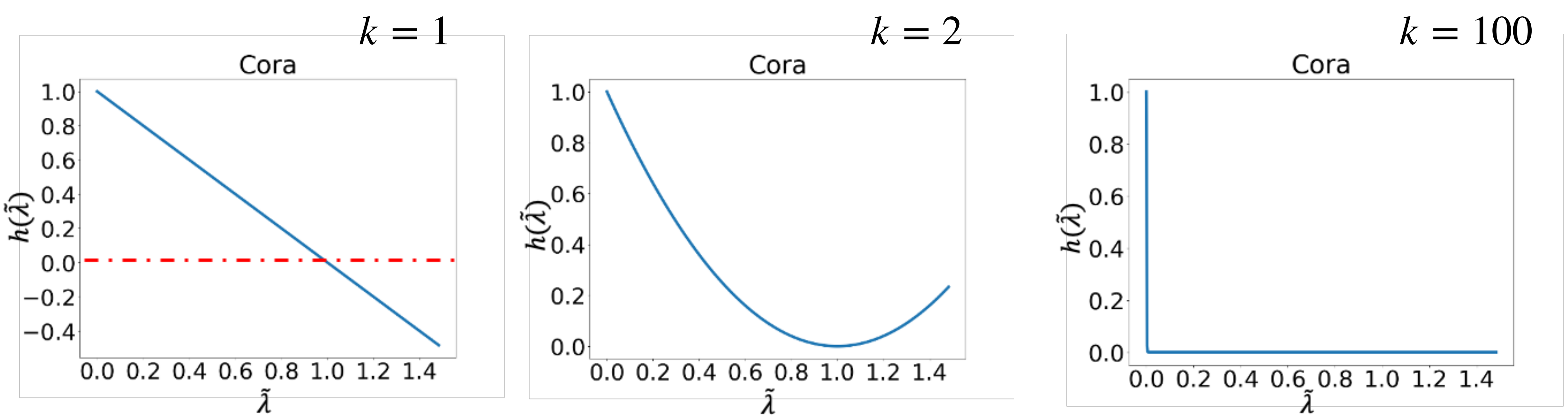
Kipf, T. N., & Welling, M. (2017). ***Semi-supervised classification with graph convolutional networks.*** *ICLR,*

Spectral View of GCN

Cause of over-smoothing: low-pass graph filter

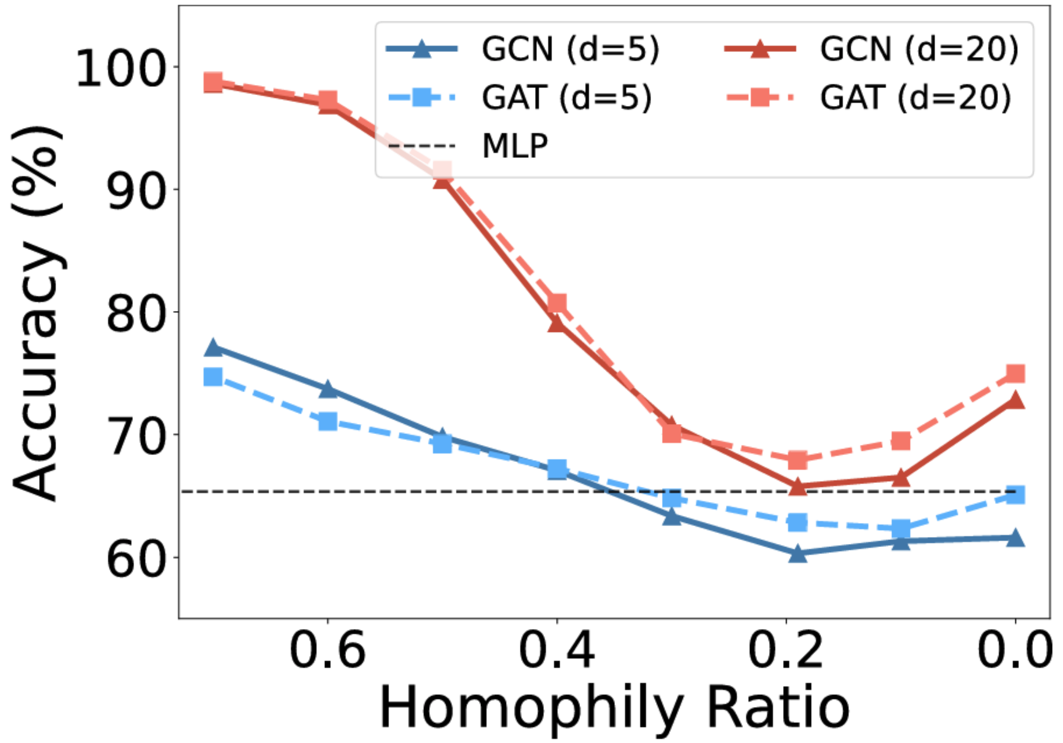
Keep smooth (low frequency) signal
reducing the impact of noise or sharp variations (high-frequency signals)

GCN for Cora graph

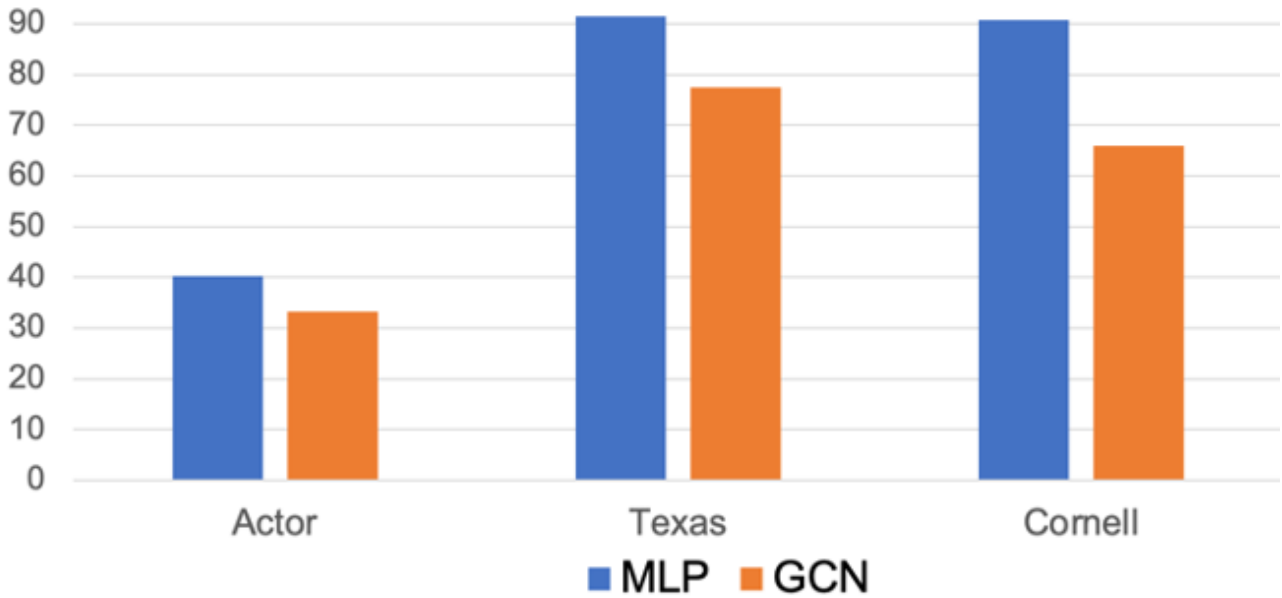


Loss of node-specific information: As GNN layers increase, the node representations become indistinguishable, losing discriminative information necessary for tasks like node classification.

Spectral View of GCN



Dai, Enyan, et al. "Label-wise graph convolutional network for heterophilic graphs." Learning on Graphs Conference. PMLR, 2022.



Talk by Zhewei Zhe:HKUST (GZ), 2023, Graph Convolutional Networks: Theory and Fundamentals

On **Heterophilic** graphs, GCN can achieve even worse performance the MLP

Homophilic graphs: Similarity-based connections
Heterophilic graphs: Difference-based connections

Spatial View of GCN

If constraint g_θ is polynomial

$$g_\theta * f = Ug_\theta U^\top f \quad = g_\theta(L) \cdot f$$

$g_\theta = g(\theta) = 2 - \theta$

$$= g_\theta(L) \cdot f$$

$$= (2 \cdot I - L) \cdot f$$

1-order polynomial without weights

$$= (2 \cdot I - (I - A)) \cdot f$$

A and L are normalized

$$= (I + A) \cdot f$$

$$= \boxed{f} + \boxed{A \cdot f}$$

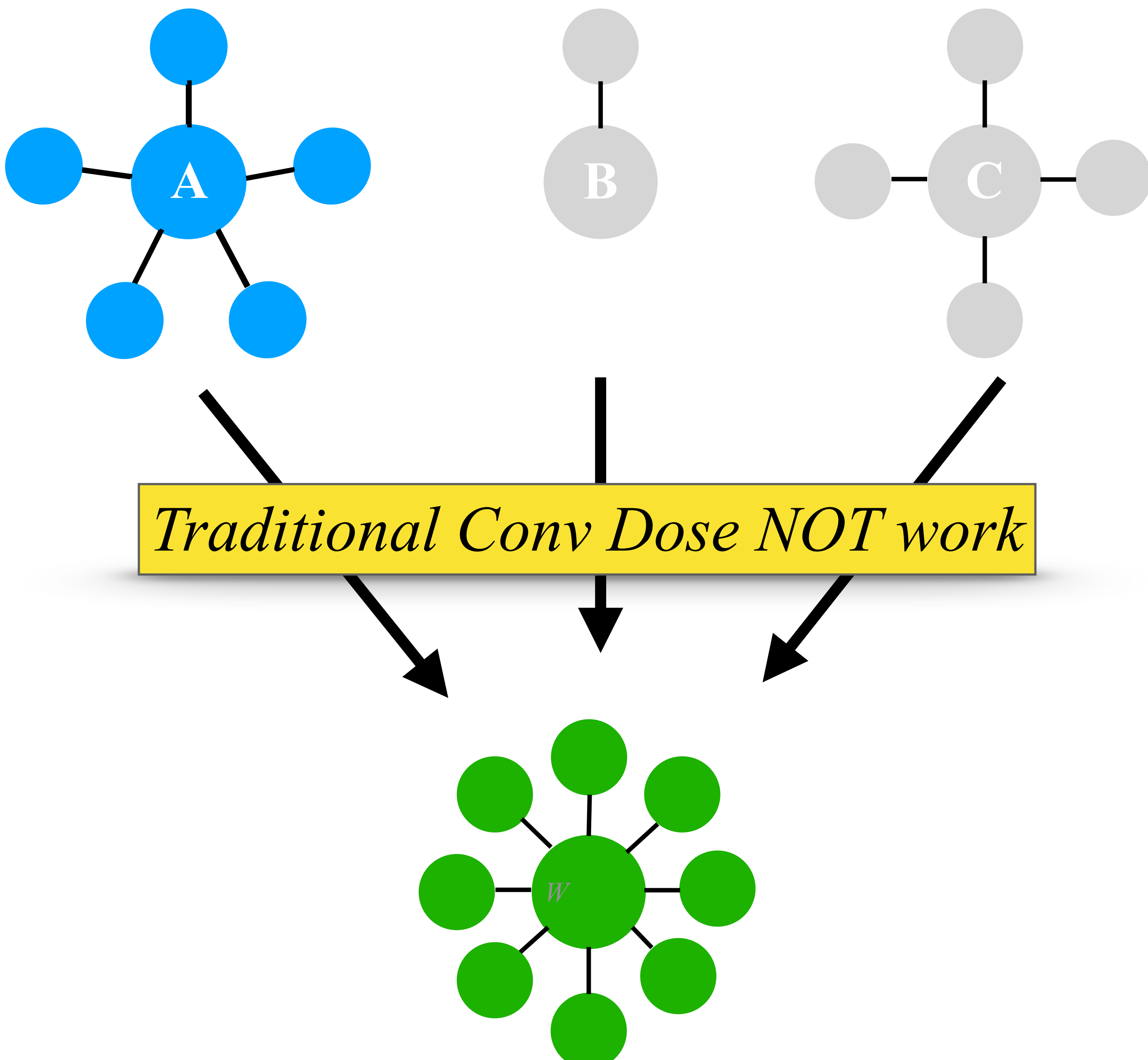
Self   Average of neighbors

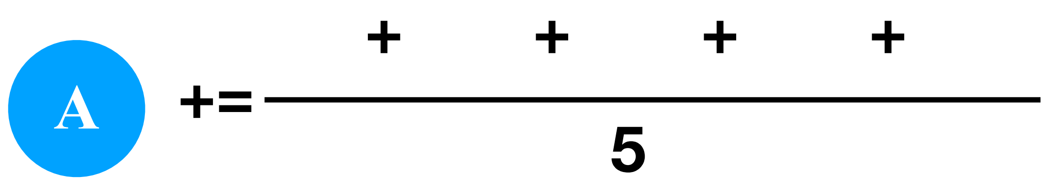
We reach here with the spectral method, but it also makes sense in spatial domain

Kipf, T. N., & Welling, M. (2017). ***Semi-supervised classification with graph convolutional networks***. ICLR,

Convolution on Graph Data

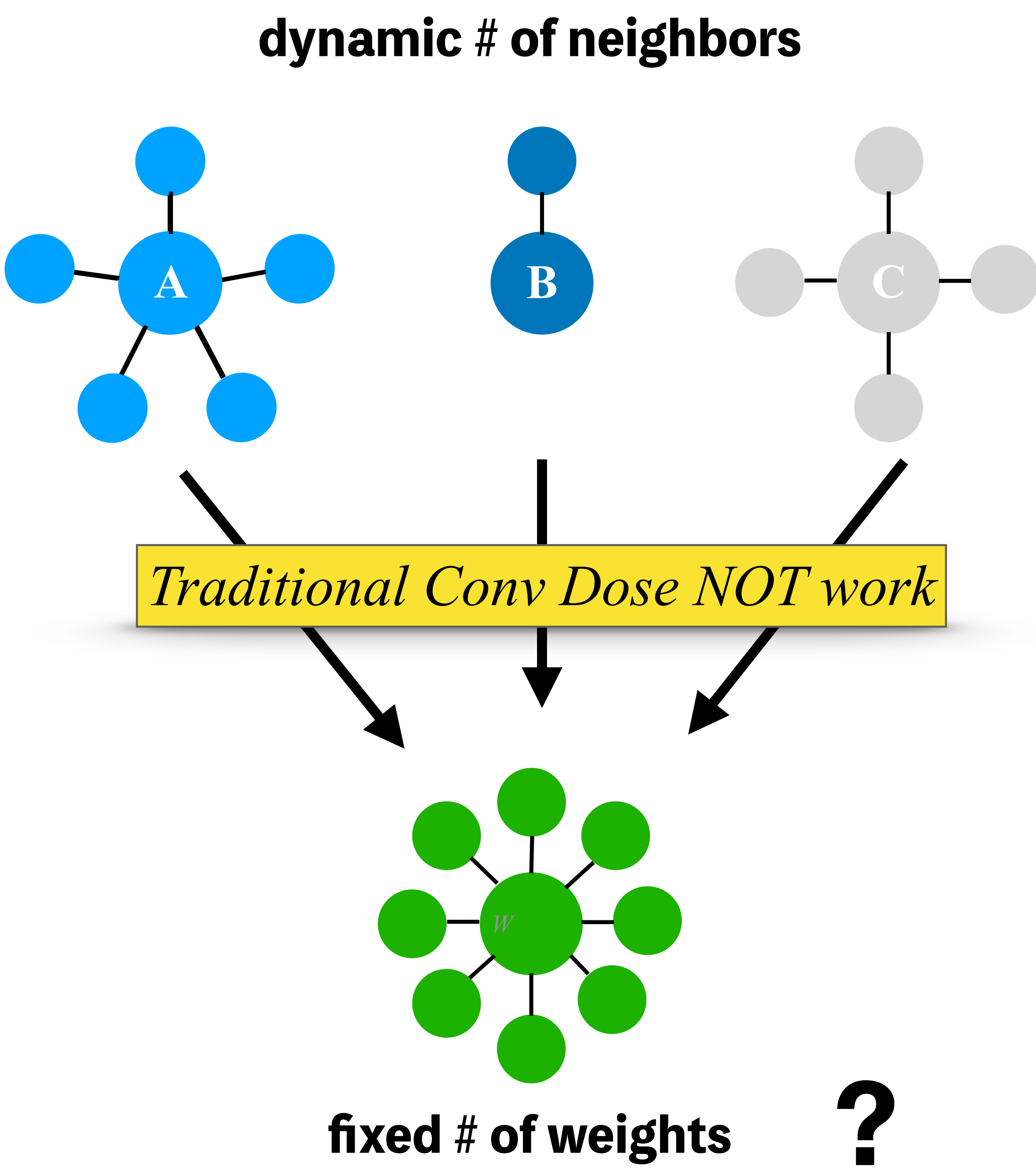
dynamic # of neighbors

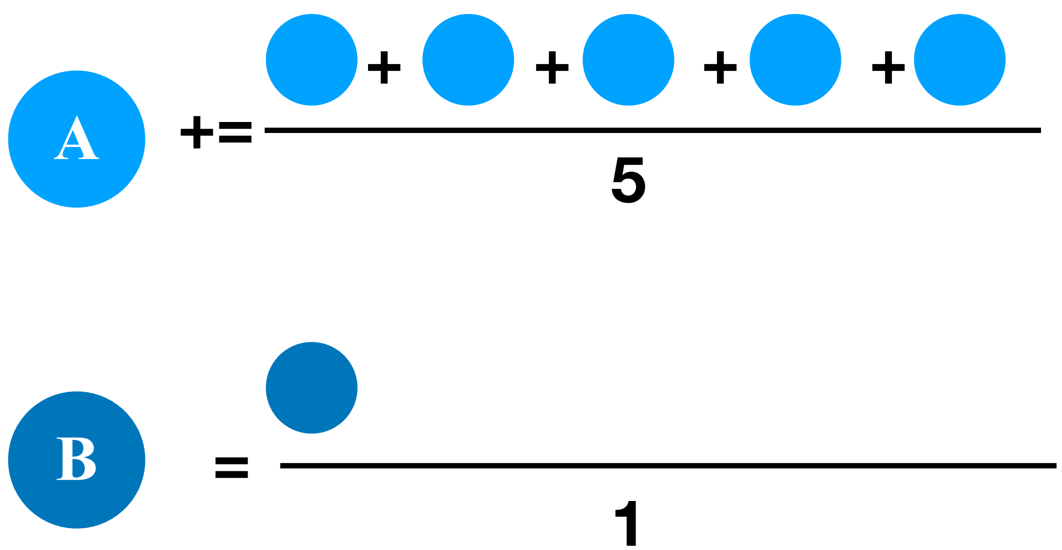




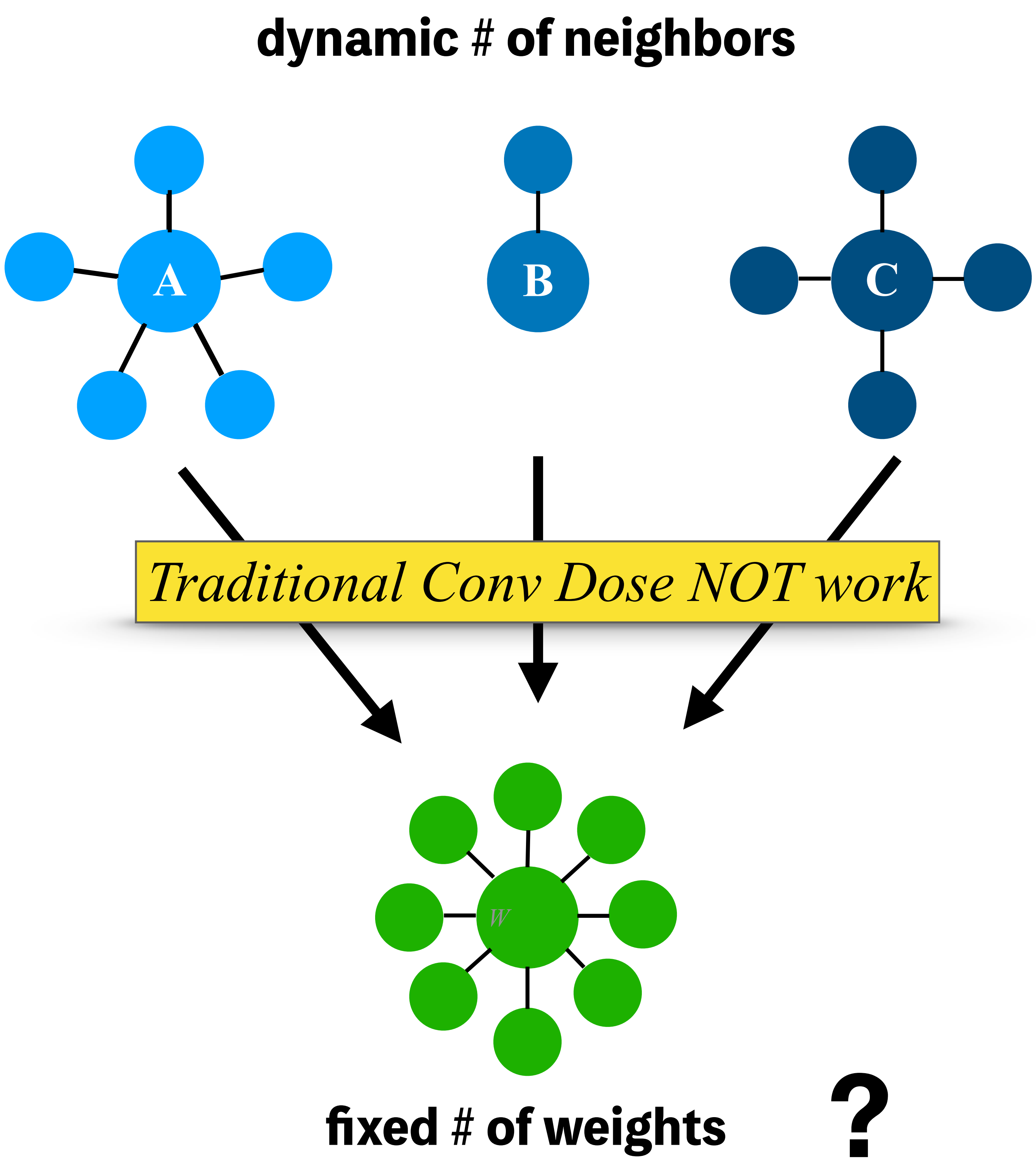
fixed # of weights **?**

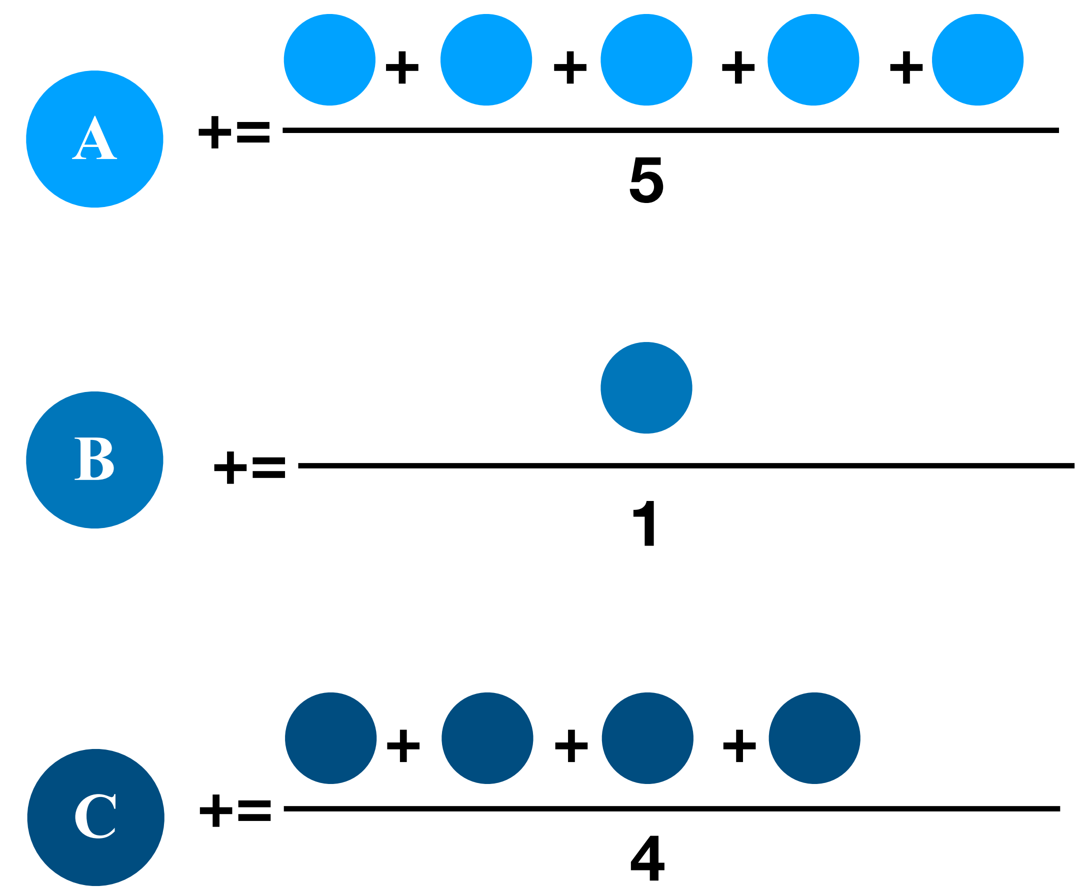
Convolution on Graph Data





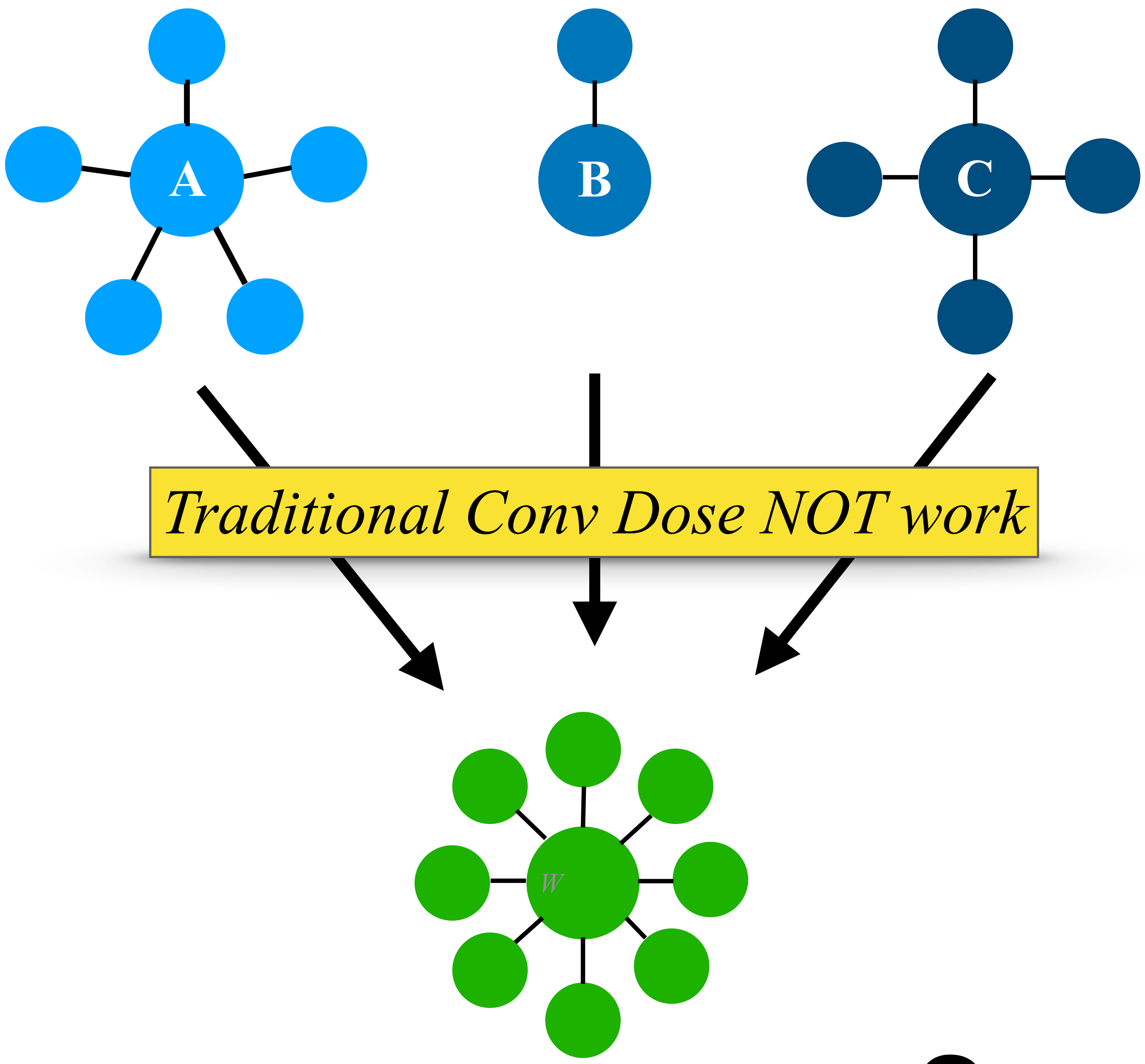
Convolution on Graph Data





Convolution on Graph Data

dynamic # of neighbors



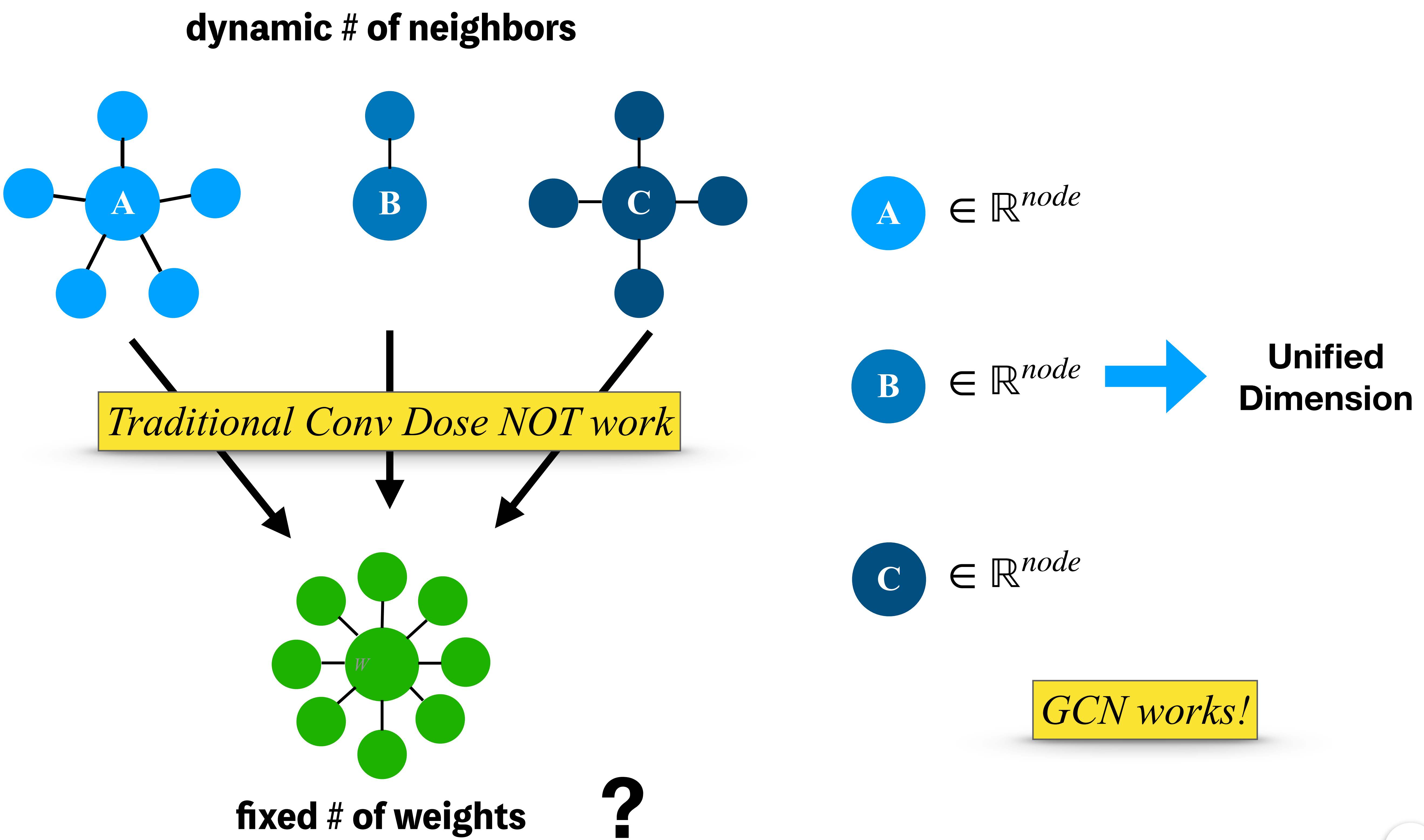
fixed # of weights **?**

A $\in \mathbb{R}^{node}$

B $\in \mathbb{R}^{node}$

C $\in \mathbb{R}^{node}$

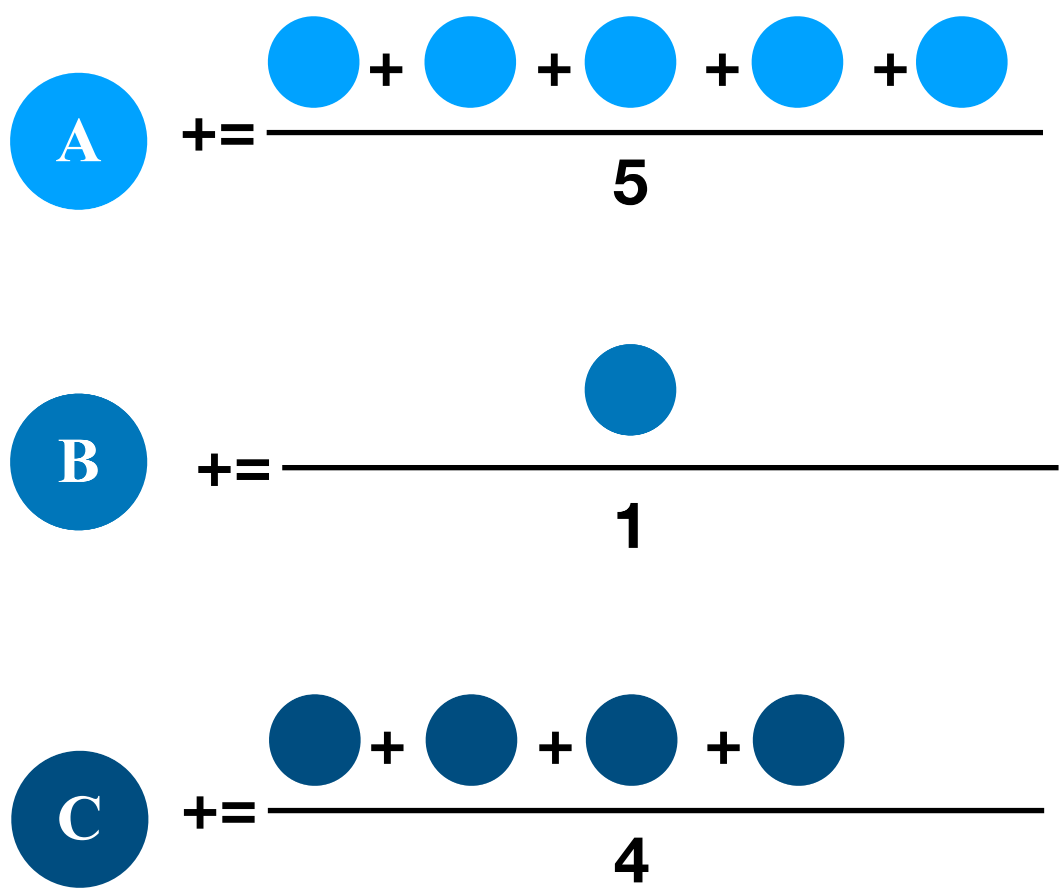
Convolution on Graph Data



Convolution on Graph Data

If constraint g_θ is polynomial, rational or exp function

$$g_\theta * f = U g_\theta U^\top f = \overset{\text{Self}}{\boxed{f}} + \overset{\text{Average of neighbors}}{\boxed{A \cdot f}}$$

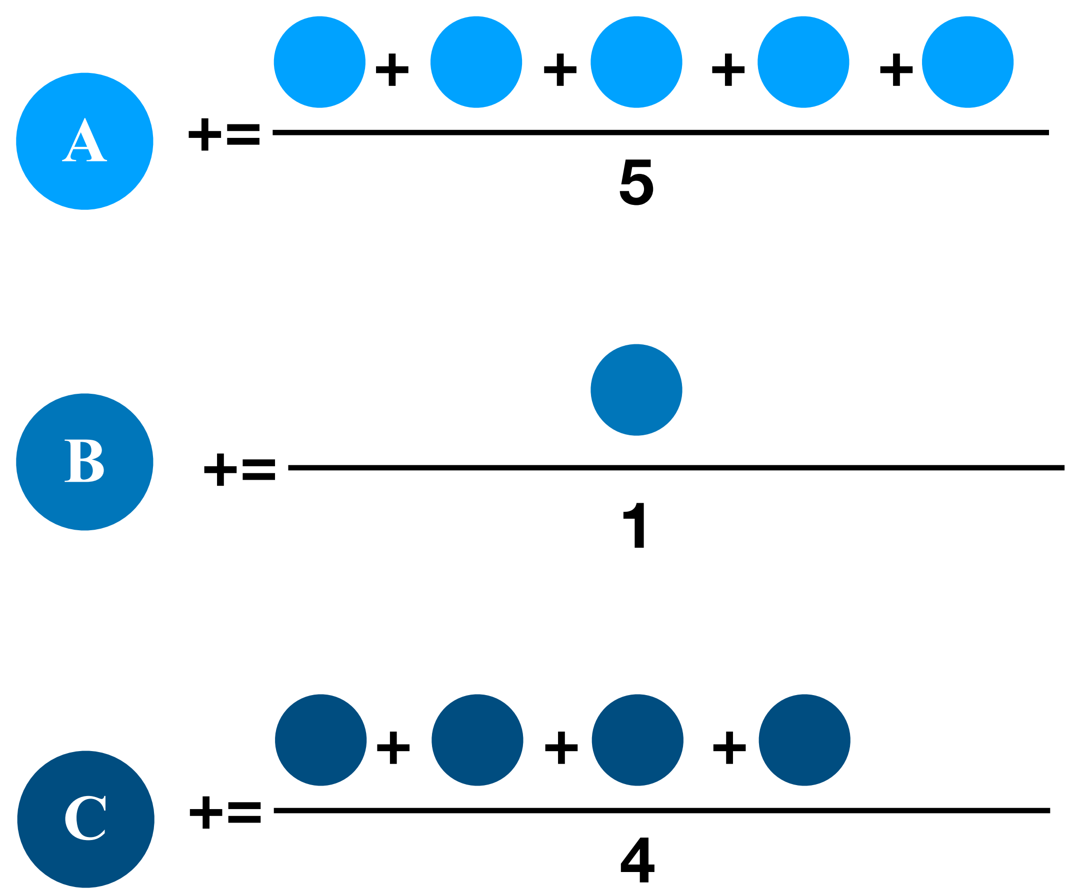


Kipf, T. N., & Welling, M. (2017). ***Semi-supervised classification with graph convolutional networks***. ICLR,

Convolution on Graph Data

If constraint g_θ is polynomial, rational or exp function

$$g_\theta * f = U g_\theta U^\top f \quad = \overset{\text{Self}}{\boxed{f}} + \overset{\text{Average of neighbors}}{\boxed{A \cdot f}}$$

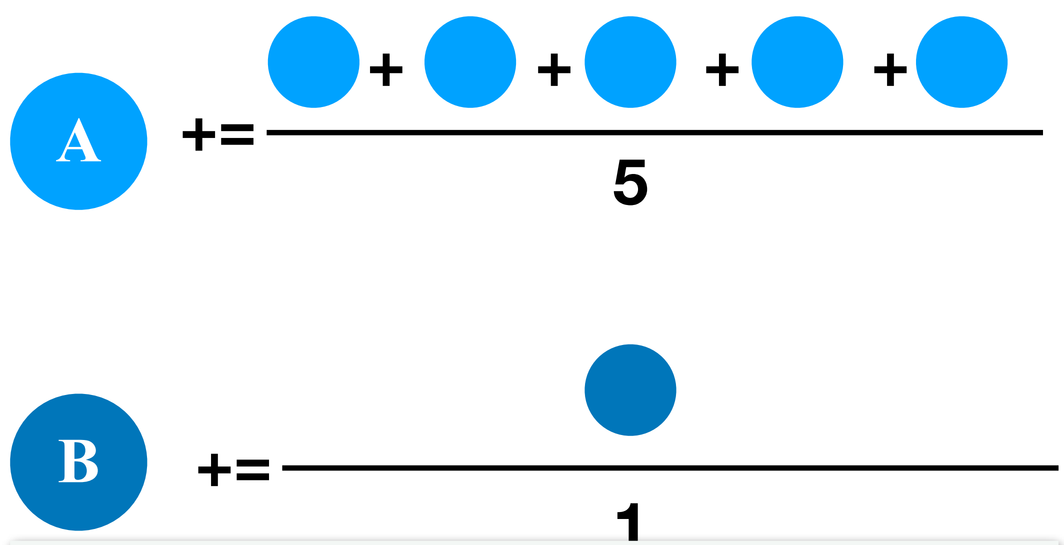


Kipf, T. N., & Welling, M. (2017). ***Semi-supervised classification with graph convolutional networks***. ICLR,

Convolution on Graph Data

If constraint g_θ is polynomial, rational or exp function

$$g_\theta * f = U g_\theta U^\top f = \overset{\text{Self}}{\boxed{f}} + \overset{\text{Average of neighbors}}{\boxed{A \cdot f}}$$



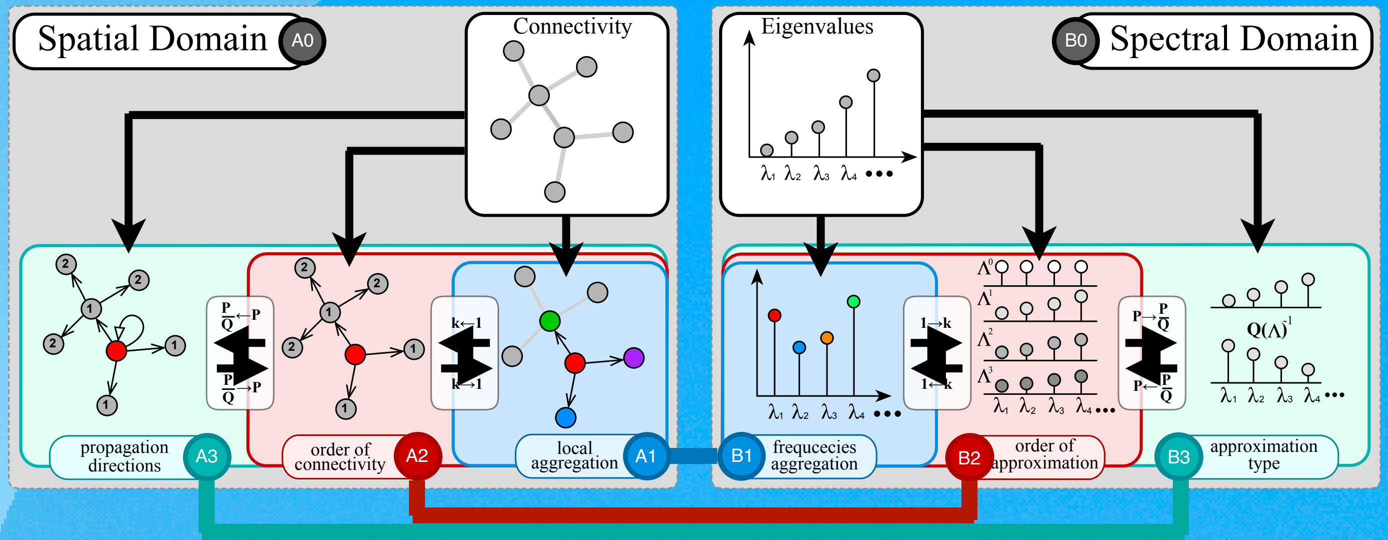
Convolution theorem

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

Space convolution = frequency multiplication

*Kipf, T. N., & Welling, M. (2017). **Semi-supervised classification with graph convolutional networks**. ICLR,*

Half Time Break (30 min)



Paper Collection

Agenda

- **First Half (1 hour 30 min)**
  - *Background*: **unified frameworks for GNN** (35 min)
  - *Preliminary*: **graph convolutions** (40 min)
  - BREAK (15min)
- **Second Half (1 hour)**
  - *Introduction*: **a new unified framework** (40 min)
  - *Future directions* (20min)
  - Q&A (15min)

Agenda

- Formal Definitions
- Normalization
- Case Study 1: Linear
- Case Study 2: Polynomial
- Case Study 3: Rational
- Comparison between Spectral v.s. Spatial

Spatial & Spectral Methods

- **Spatial** Methods $g(A, X) = g_\theta(A)X$

  *Function of **graph** (matrix)*

- **Spectral** Methods $g(\Lambda, X) = Ug_\theta(\Lambda)U^T X$

  *Function of **eigenvalue** of (graph matrix)*

Normalization and its Reasons

Normalized vs Unnormalized

"Mean" *"Sum"*

Notations	Descriptions
$\mathbf{A}$	Adjacency matrix
$\mathbf{L}$	Graph Laplacian
$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$	Adjacency with self loop
$\mathbf{D}^{-1}\mathbf{A}$	Random walk row normalized adjacency
$\mathbf{A}\mathbf{D}^{-1}$	Random walk column normalized adjacency
$\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$	Symmetric normalized adjacency
$\tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}$	Left renormalized adjacency, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$
$\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1}$	Right renormalized
$\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$	Symmetric renormalized

Chen, Zhiqian, et al. ***"Bridging the gap between spatial and spectral domains: A unified framework for graph neural networks."*** *ACM Computing Surveys 2023*

Normalization and its Reasons

- Spatial reason
  - Suppose a two-cluster partitioning for A and B

- **Ratio Cut**: $cut(A,B)(\frac{1}{|A|}+\frac{1}{|B|})$

  Unnormalized

  - Use **# node** to cluster graph

- **Normalized Cut**: $cut(A,B)(\frac{1}{Vol(A)}+\frac{1}{Vol(B)})$

  Normalized

  - Use **# link** to cluster graph

Von Luxburg, Ulrike. "**A tutorial on spectral clustering.**" Statistics and computing 17 (2007): 395-416.

Normalization and its Reasons

- Spectral reason

- **Eigenvalue** $\in [0, \lambda_{max}]$
  - λ_{max} < max degree of the graph

Unnormalized

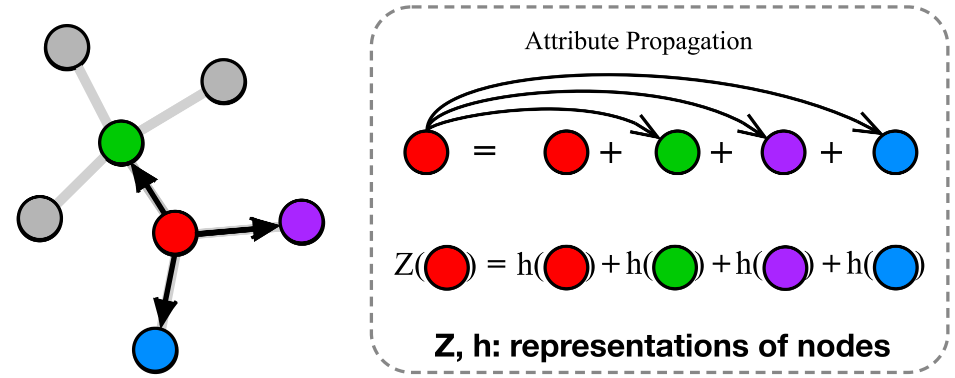
- **Eigenvalue** $\in [0,2]$
  - random walk or symmetric normalization

Normalized

*Von Luxburg, Ulrike. "**A tutorial on spectral clustering.**" Statistics and computing 17 (2007): 395-416.*

Case Study 1: GCN

The "GCN" *Thomas N. Kipf et al. (2017)*

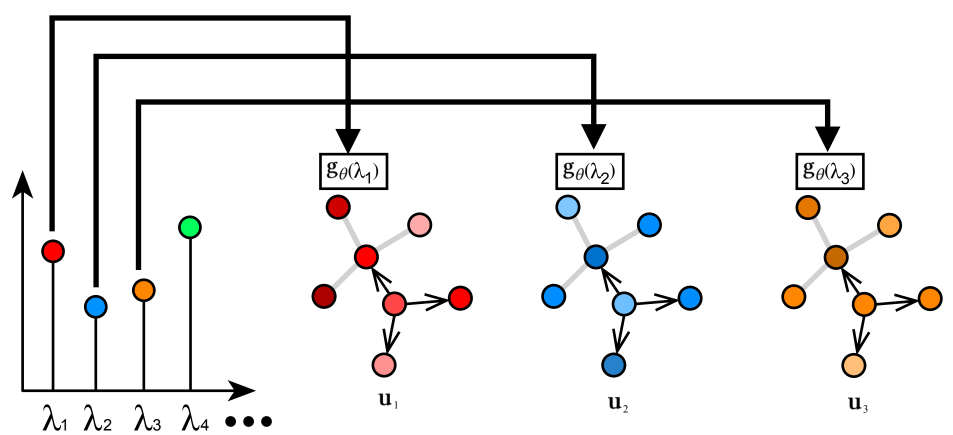


$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}\hat{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I} + \mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \boxed{(\mathbf{I} + \tilde{\mathbf{A}})}\mathbf{X}$$

In form of $g_\theta(A)X$, and $g_\theta(A) = \tilde{A}^0 + \tilde{A}^1$

Case Study 1: GCN

GCN *Thomas N. Kipf et al. (2017)*



$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{L}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2-\Lambda)\mathbf{U}^\top\mathbf{X}$$

In form of $Ug_\theta(\Lambda)U^TX$

Case Study 1: GCN

GCN *Thomas N. Kipf et al. (2016)*

Spatial

Average neighbors

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I} + \mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I} + \tilde{\mathbf{A}})\mathbf{X}$$

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I} + \mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2 - \Lambda)\mathbf{U}^{\top}\mathbf{X}$$

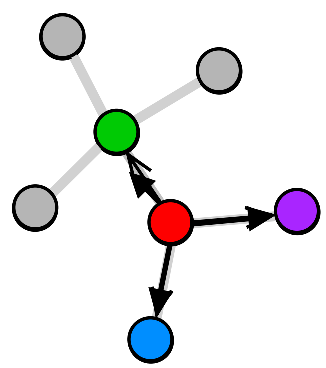
Spectral

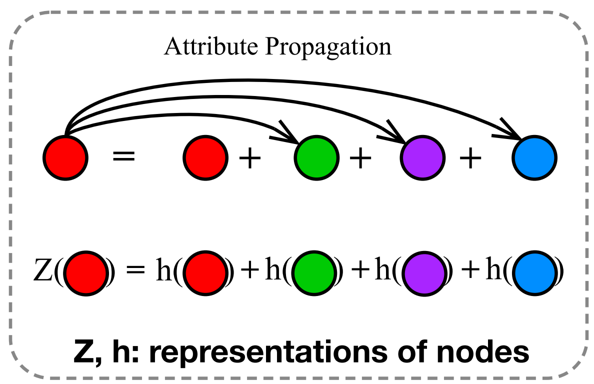
Linear function

Spatial-based GNN: Linear

Function of **graph matrix** $g(A)X$

Linear





GCN *Thomas N. Kipf et al. (2016)*

$$\mathbf{Z} = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X} = \hat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

GraphSAGE

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

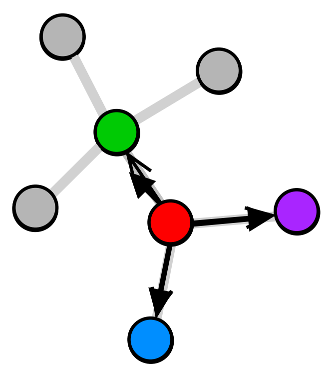
w/ mean aggregator

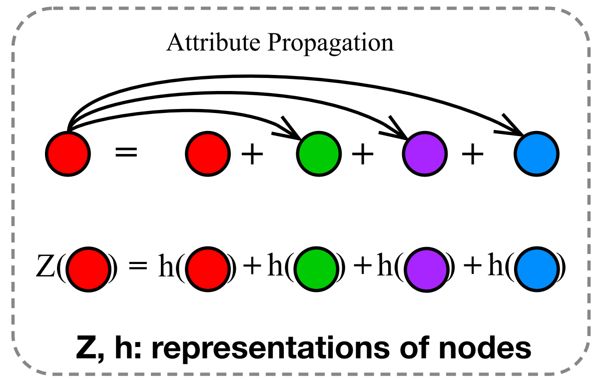
Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." Advances in neural information processing systems 30 (2017).

Spatial-based GNN: Linear

Function of **graph matrix** $g(A)X$

Linear





GCN *Thomas N. Kipf et al. (2016)*

$$\mathbf{Z} = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X} = \hat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

GraphSAGE *Will Hamilton et al. (2017)*

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

GIN

$$\mathbf{Z} = (1+\epsilon)\cdot\mathbf{h}(v) + \sum_{u_j\in\mathcal{N}(v_i)}\mathbf{h}\left(u_j\right) = [(1+\epsilon)\mathbf{I}+\mathbf{A}]\mathbf{X}$$

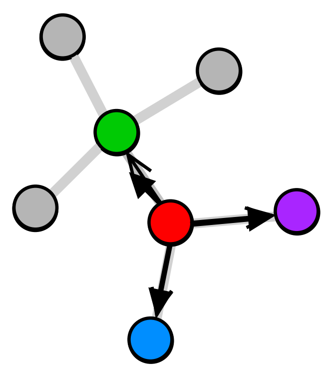
Control hyperparameter

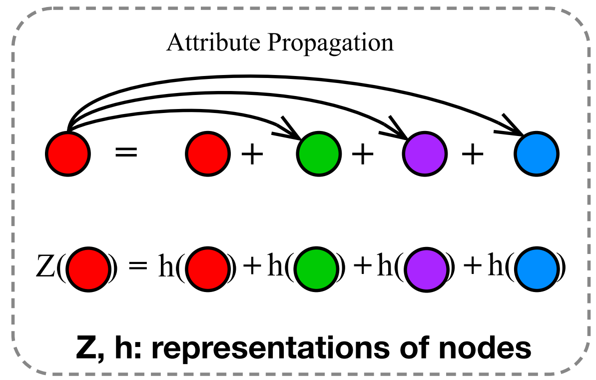
Xu, Keyulu, et al. "How Powerful are Graph Neural Networks?." International Conference on Learning Representations. 2018.

Spatial-based GNN: Linear

Function of **graph matrix** $g(A)X$

Linear





GCN *Thomas N. Kipf et al. (2016)*

$$\mathbf{Z} = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X} = \hat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

GraphSAGE *Will Hamilton et al. (2017)*

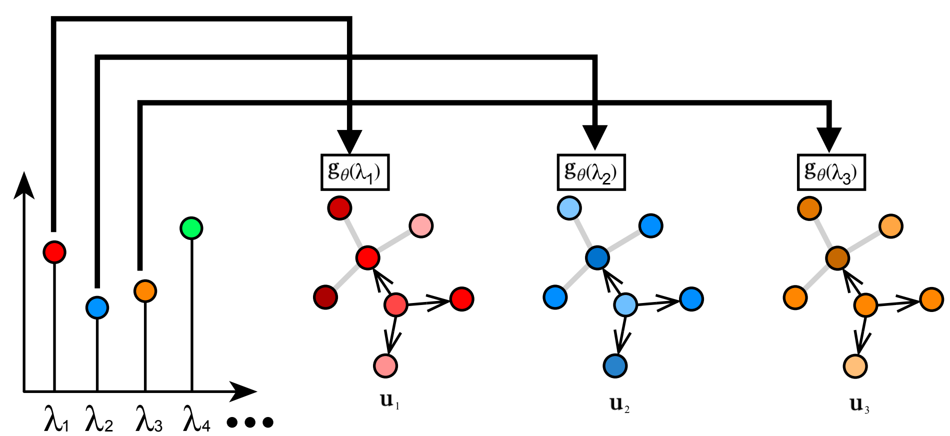
$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

GIN

$$\mathbf{Z} = (1+\epsilon)\cdot\mathbf{h}(v) + \sum_{u_j\in\mathcal{N}(v_i)}\mathbf{h}_{(u_j)} = [(1+\epsilon)\mathbf{I}+\mathbf{A}]\mathbf{X}$$

Function of **eigenvalue** $Ug_\theta(\Lambda)U^TX$

Linear



GCN *Thomas N. Kipf et al. (2016)*

$$\mathbf{Z} = \tilde{\mathbf{A}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{L}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}-\mathbf{L}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2-\Lambda)\mathbf{U}^\top\mathbf{X}$$

GraphSAGE *Will Hamilton et al. (2017)*

$$Z = D^{-\frac{1}{2}}(I+A)D^{-\frac{1}{2}}X = (I+\tilde{A})X = (2I-\tilde{L})X = U(2-\Lambda)U^\top X$$

GIN *Xukeyu Lu et al. (2019)*

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}[(1+\epsilon)\mathbf{I}+\mathbf{A}]\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}[(2+\epsilon)\mathbf{I}-\tilde{\mathbf{L}}]\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2+\epsilon-\Lambda)\mathbf{U}^\top\mathbf{X}$$

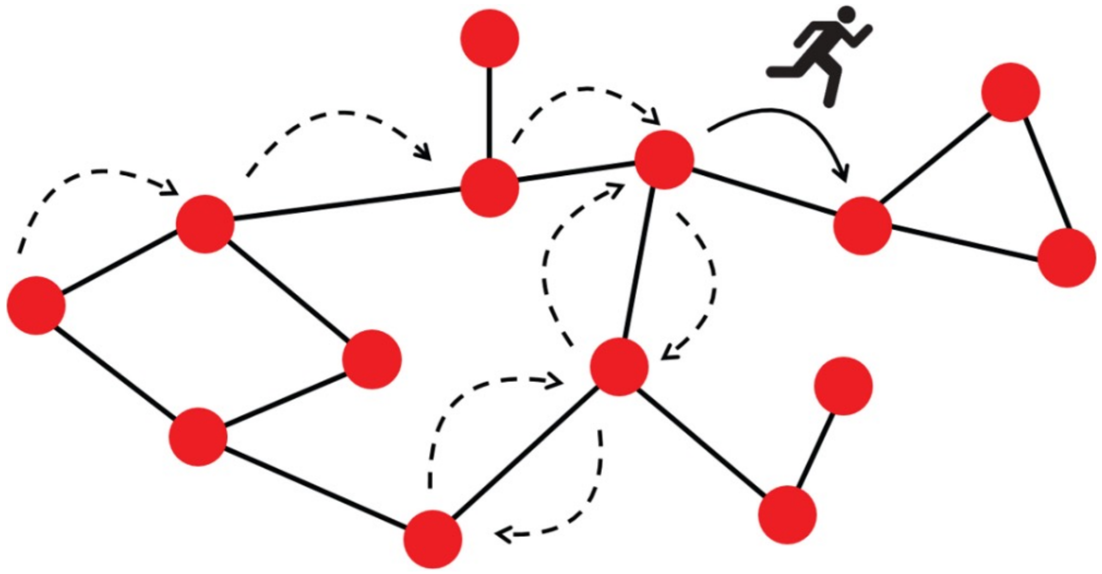
Case Study 2: DeepWalk

- Draw a group of random paths from a graph

$$\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$$

- Let the window size (path length) of skip-gram be $2t$+1and the current node is the (t+1)-th

$$\mathbf{Z} = \frac{1}{t+1}(\mathbf{I} + \tilde{\mathbf{A}} + \tilde{\mathbf{A}}^2 + \ldots + \tilde{\mathbf{A}}^t)\,\mathbf{X}$$

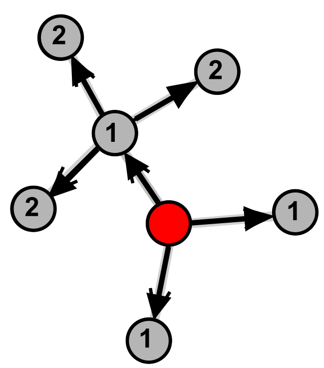


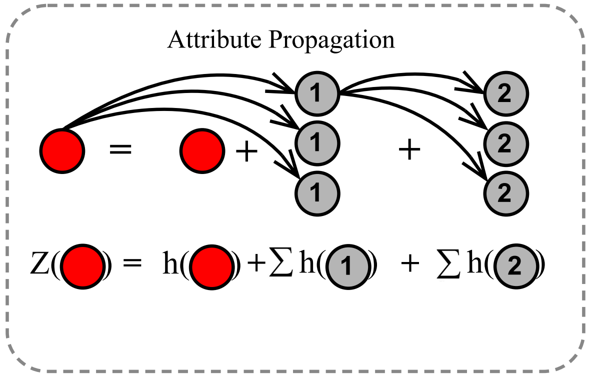
Img credit: DOI: (10.1002/sim.9346)

Spectral-based GNN: Polynomial

Function of **graph matrix** $\quad g(A)X$

Polynomial





DeepWalk *Bryan Perozzi et al. (2014)*

$$\mathbf{Z} = \frac{1}{t+1}\left(\mathbf{I} + \tilde{\mathbf{A}} + \tilde{\mathbf{A}}^2 + \ldots + \tilde{\mathbf{A}}^t\right)\mathbf{X} = \frac{1}{t+1}\mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

ChebyNet *Defferrard, Michael et al. (2016)*

$$\mathbf{Z} = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \left[\tilde{\theta}_0\mathbf{I} + \tilde{\theta}_1(\mathbf{I} - \tilde{\mathbf{A}}) + \tilde{\theta}_2(\mathbf{I} - \tilde{\mathbf{A}})^2 + \ldots\right]\mathbf{X} = \left(\phi\mathbf{I} + \sum_{i=1}^{k} \psi_i \tilde{\mathbf{A}}^i\right)\mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

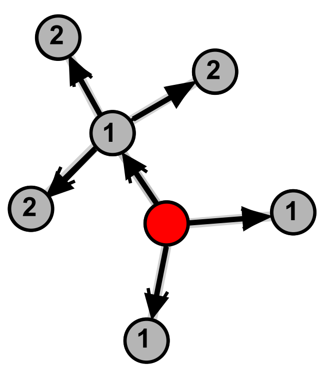
Chebyshev polynomial (1st kind) of L

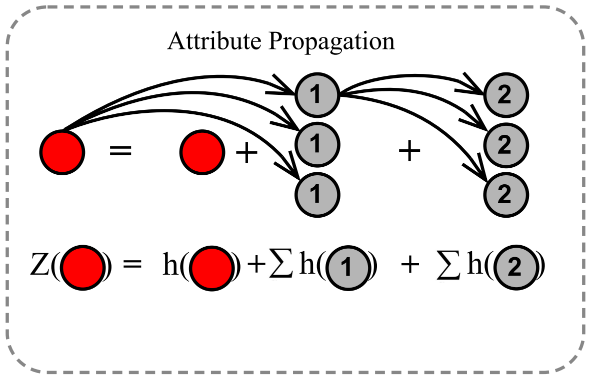
*Defferrard, et al., "**Convolutional neural networks on graphs with fast localized spectral filtering.**" Advances in neural information processing systems 29 (2016).*

Spectral-based GNN: Polynomial

Function of **graph matrix** $g(A)X$

Polynomial





DeepWalk *Bryan Perozzi et al. (2014)*

$$\mathbf{Z} = \frac{1}{t+1}\left(\mathbf{I}+\tilde{\mathbf{A}}+\tilde{\mathbf{A}}^2+\ldots+\tilde{\mathbf{A}}^t\right)\mathbf{X} = \frac{1}{t+1}\mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

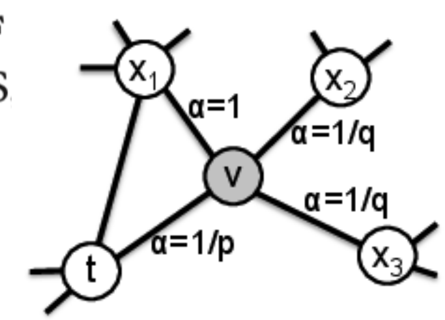
ChebyNet *Defferrard, Michael et al. (2016)*

$$\mathbf{Z} = \sum_{k=0}^{K-1}\theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \left[\tilde{\theta}_0\mathbf{I} + \tilde{\theta}_1(\mathbf{I}-\tilde{\mathbf{A}}) + \tilde{\theta}_2(\mathbf{I}-\tilde{\mathbf{A}})^2 + \ldots\right]\mathbf{X} = \left(\phi\mathbf{I} + \sum_{i=1}^{k}\psi_i\tilde{\mathbf{A}}^i\right)\mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

Node2Vec *Aditya Grover et al. (2016)*

$$\mathbf{Z} = \left(\frac{1}{p}\cdot\overbrace{\mathbf{I}}^{\text{source}} + \overbrace{\tilde{\mathbf{A}}}^{\text{BFS}} + \frac{1}{q}\overbrace{(\tilde{\mathbf{A}}^2-\tilde{\mathbf{A}})}^{\text{DFS}}\right)\mathbf{X}$$

$$P(t \rightarrow x) = \begin{cases} \frac{1}{p} & \text{if } d(t,x) = 0, \text{ return to the source} \\ 1 & \text{if } d(t,x) = 1, \text{ BSF} \\ \frac{1}{q} & \text{if } d(t,x) = 2, \text{ DFS} \end{cases}$$

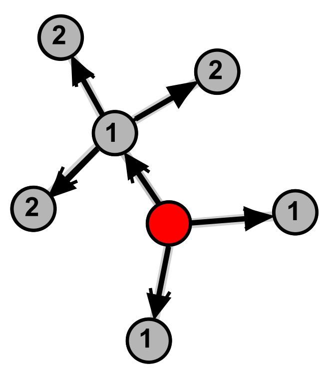


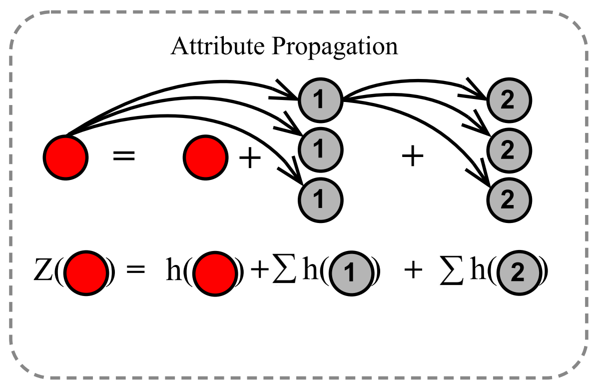
$$\mathbf{Z} = \left(\frac{1}{p}\cdot\mathbf{I} + \tilde{\mathbf{A}} + \frac{1}{q}\left(\tilde{\mathbf{A}}^2-\tilde{\mathbf{A}}\right)\right)\mathbf{X} = \left[\frac{1}{p}\mathbf{I} + \left(1-\frac{1}{q}\right)\tilde{\mathbf{A}} + \frac{1}{q}\tilde{\mathbf{A}}^2\right]\mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

Spectral-based GNN: Polynomial

Function of **graph matrix** $g(A)X$

Polynomial





DeepWalk *Bryan Perozzi et al. (2014)*

$$\mathbf{Z} = \frac{1}{t+1}\left(\mathbf{I} + \tilde{\mathbf{A}} + \tilde{\mathbf{A}}^2 + \ldots + \tilde{\mathbf{A}}^t\right)\mathbf{X} = \frac{1}{t+1}\mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

ChebyNet *Defferrard, Michael et al. (2016)*

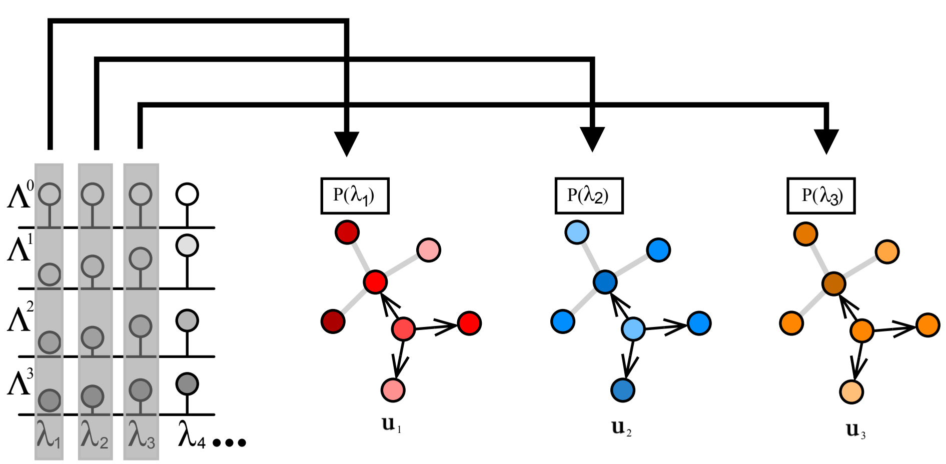
$$\mathbf{Z} = \sum_{k=0}^{K-1}\theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \left[\tilde{\theta}_0\mathbf{I} + \tilde{\theta}_1(\mathbf{I}-\tilde{\mathbf{A}}) + \tilde{\theta}_2(\mathbf{I}-\tilde{\mathbf{A}})^2 + \ldots\right]\mathbf{X} = \left(\phi\mathbf{I} + \sum_{i=1}^{k}\psi_i\tilde{\mathbf{A}}^i\right)\mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

Node2Vec *Aditya Grover et al. (2016)*

$$\mathbf{Z} = \left(\frac{1}{p}\cdot\mathbf{I} + \tilde{\mathbf{A}} + \frac{1}{q}\left(\tilde{\mathbf{A}}^2 - \tilde{\mathbf{A}}\right)\right)\mathbf{X} = \left[\frac{1}{p}\mathbf{I} + \left(1-\frac{1}{q}\right)\tilde{\mathbf{A}} + \frac{1}{q}\tilde{\mathbf{A}}^2\right]\mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

Function of **eigenvalue** $Ug_\theta(\Lambda)U^TX$

Polynomial



DeepWalk *Bryan Perozzi et al. (2014)*

$$\mathbf{Z} = \frac{1}{t+1}\left(\mathbf{I} + (\mathbf{I}-\tilde{\mathbf{L}}) + (\mathbf{I}-\tilde{\mathbf{L}})^2 + \ldots + (\mathbf{I}-\tilde{\mathbf{L}})^t\right)\mathbf{X} = \mathbf{U}\left(\theta_0 + \theta_1\mathbf{\Lambda} + \theta_2\mathbf{\Lambda}^2 + \ldots + \theta_t\mathbf{\Lambda}^t\right)\mathbf{U}^\top\mathbf{X}$$

ChebyNet *Defferrard, Michael et al. (2016)*

$$\mathbf{Z} = \sum_{k=0}^{K-1}\theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \mathbf{U}\left(\tilde{\theta}_0\cdot 1 + \tilde{\theta}_1\Lambda + \tilde{\theta}_2\Lambda^2 + \ldots\right)\mathbf{U}^\top\mathbf{X}$$

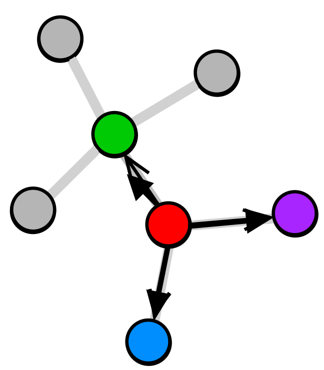
Node2Vec *Aditya Grover et al. (2016)*

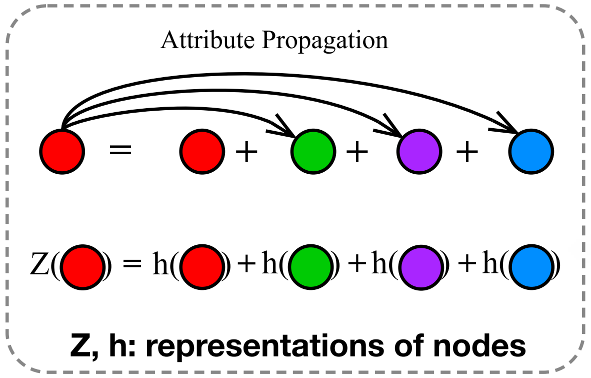
$$\mathbf{Z} = \left[\left(1+\frac{1}{p}\right)\mathbf{I} - \left(1+\frac{1}{q}\right)\tilde{\mathbf{L}} + \frac{1}{q}\tilde{\mathbf{L}}^2\right]\mathbf{X} = \mathbf{U}\left[\left(1+\frac{1}{p}\right) - \left(1+\frac{1}{q}\right)\tilde{\Lambda} + \frac{1}{q}\tilde{\Lambda}^2\right]\mathbf{U}^\top\mathbf{X}$$

Linear v.s. Polynomial

Function of **graph matrix** $g(A)X$

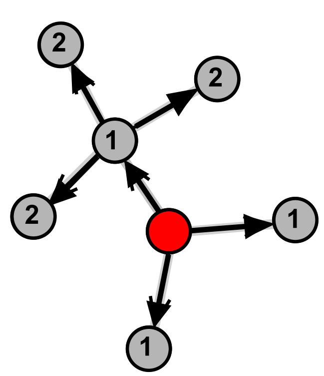
Linear

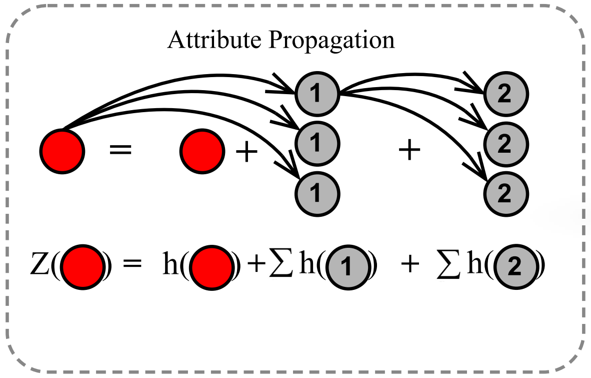




*only consider the **direct** neighbors*

Polynomial





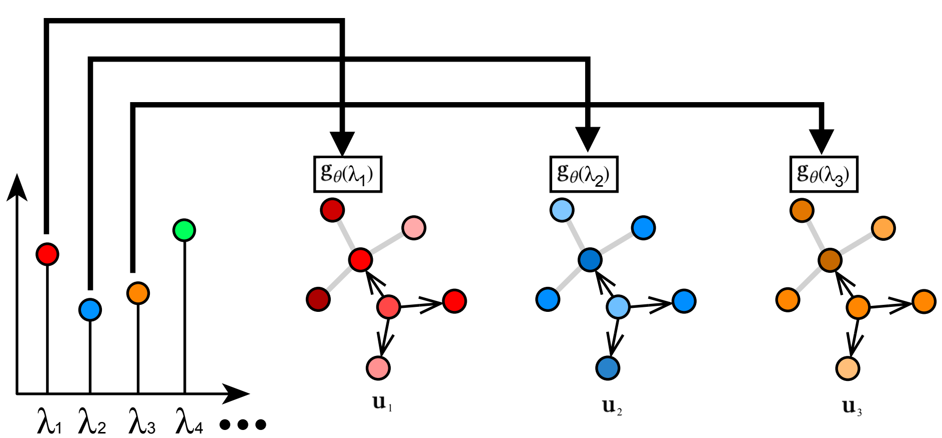
*consider the **higher-order** neighbors*

Linear v.s. Polynomial

"spectral response functions"

Function of **eigenvalue** $\quad Ug_{\theta}(\Lambda)U^{T}X$

Linear



GCN *Thomas N. Kipf et al. (2016)*

$$\mathbf{Z} = \tilde{\mathbf{A}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{L}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}-\mathbf{L}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2-\Lambda)\mathbf{U}^{\top}\mathbf{X}$$

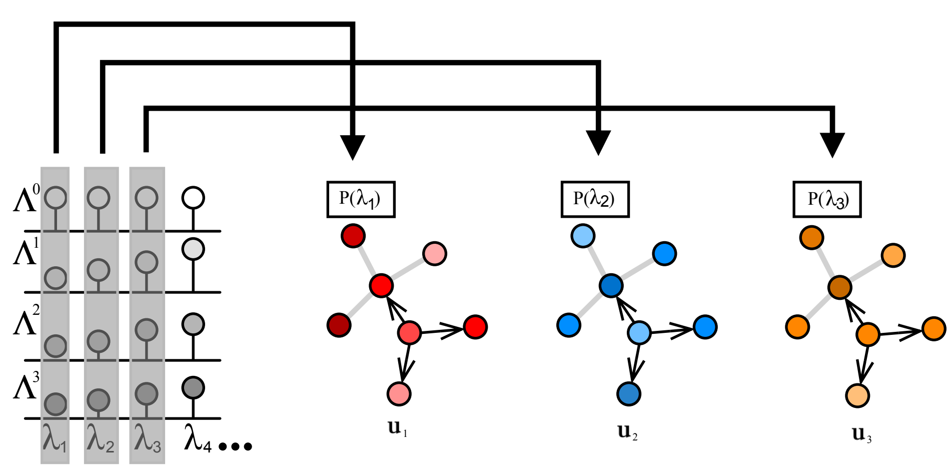
GraphSAGE *Will Hamilton et al. (2017)*

$$Z = D^{-\frac{1}{2}}(I+A)D^{-\frac{1}{2}}X = (I+\tilde{A})X = (2I-\tilde{L})X = U(2-\Lambda)U^{\top}X$$

GIN *Xukeyu Lu et al. (2019)*

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}[(1+\epsilon)\mathbf{I}+\mathbf{A}]\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}[(2+\epsilon)\mathbf{I}-\tilde{\mathbf{L}}]\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2+\epsilon-\Lambda)\mathbf{U}^{\top}\mathbf{X}$$

Polynomial



DeepWalk *Bryan Perozzi et al. (2014)*

$$\mathbf{Z} = \frac{1}{t+1}\left(\mathbf{I}+(\mathbf{I}-\tilde{\mathbf{L}})+(\mathbf{I}-\tilde{\mathbf{L}})^2+\ldots+(\mathbf{I}-\tilde{\mathbf{L}})^t\right)\mathbf{X} = \mathbf{U}\left(\theta_0+\theta_1\mathbf{\Lambda}+\theta_2\mathbf{\Lambda}^2+\ldots+\theta_t\mathbf{\Lambda}^t\right)\mathbf{U}^{\top}\mathbf{X}$$

ChebyNet *Defferrard, Michael et al. (2016)*

$$\mathbf{Z} = \sum_{k=0}^{K-1}\theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \mathbf{U}\left(\tilde{\theta}_0\cdot 1+\tilde{\theta}_1\Lambda+\tilde{\theta}_2\Lambda^2+\ldots\right)\mathbf{U}^{\top}\mathbf{X}$$

Node2Vec *Aditya Grover et al. (2016)*

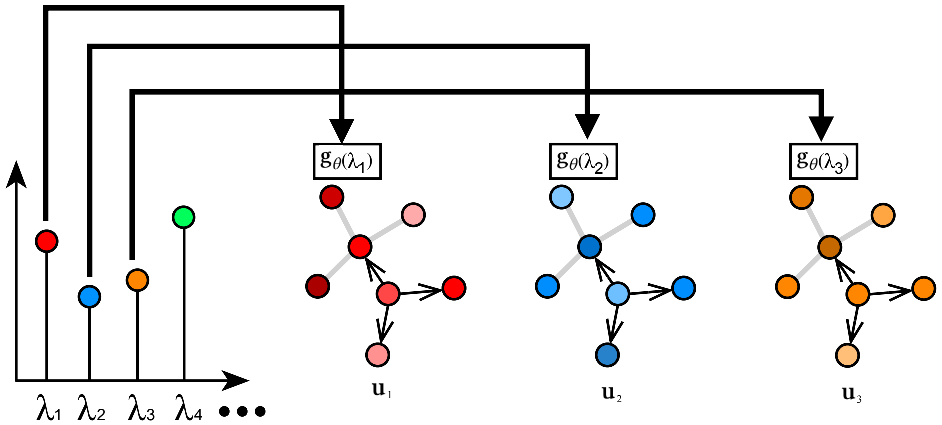
$$\mathbf{Z} = \left[\left(1+\frac{1}{p}\right)\mathbf{I}-\left(1+\frac{1}{q}\right)\tilde{\mathbf{L}}+\frac{1}{q}\tilde{\mathbf{L}}^2\right]\mathbf{X} = \mathbf{U}\left[\left(1+\frac{1}{p}\right)-\left(1+\frac{1}{q}\right)\tilde{\Lambda}+\frac{1}{q}\tilde{\Lambda}^2\right]\mathbf{U}^{\top}\mathbf{X}$$

Linear v.s. Polynomial

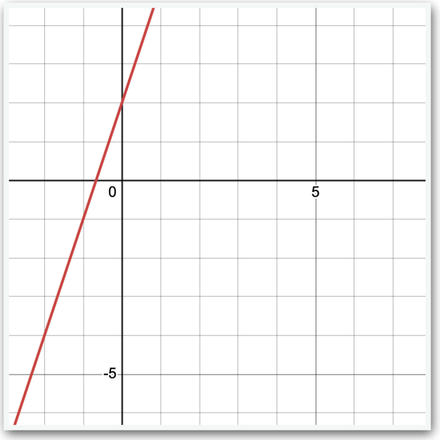
"spectral response functions"

Function of **eigenvalue** $Ug_\theta(\Lambda)U^TX$

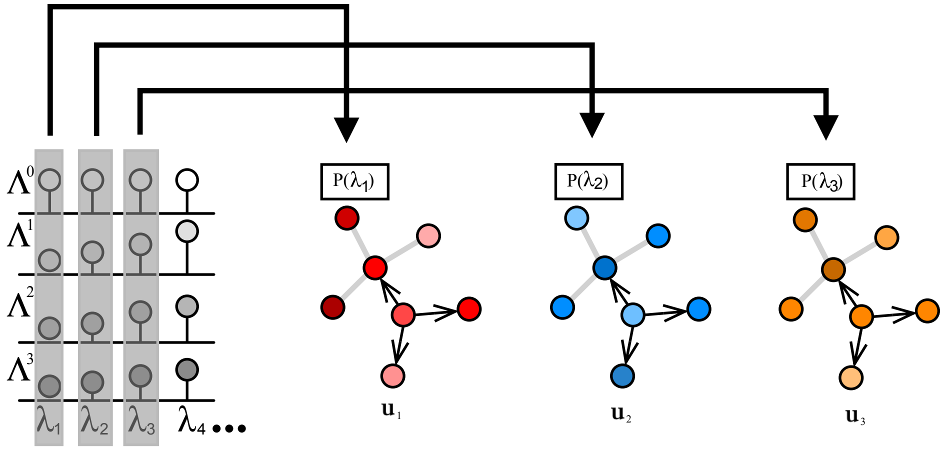
Linear



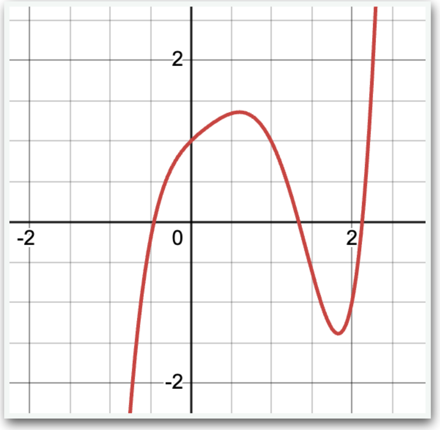
$y = 3x + 2$



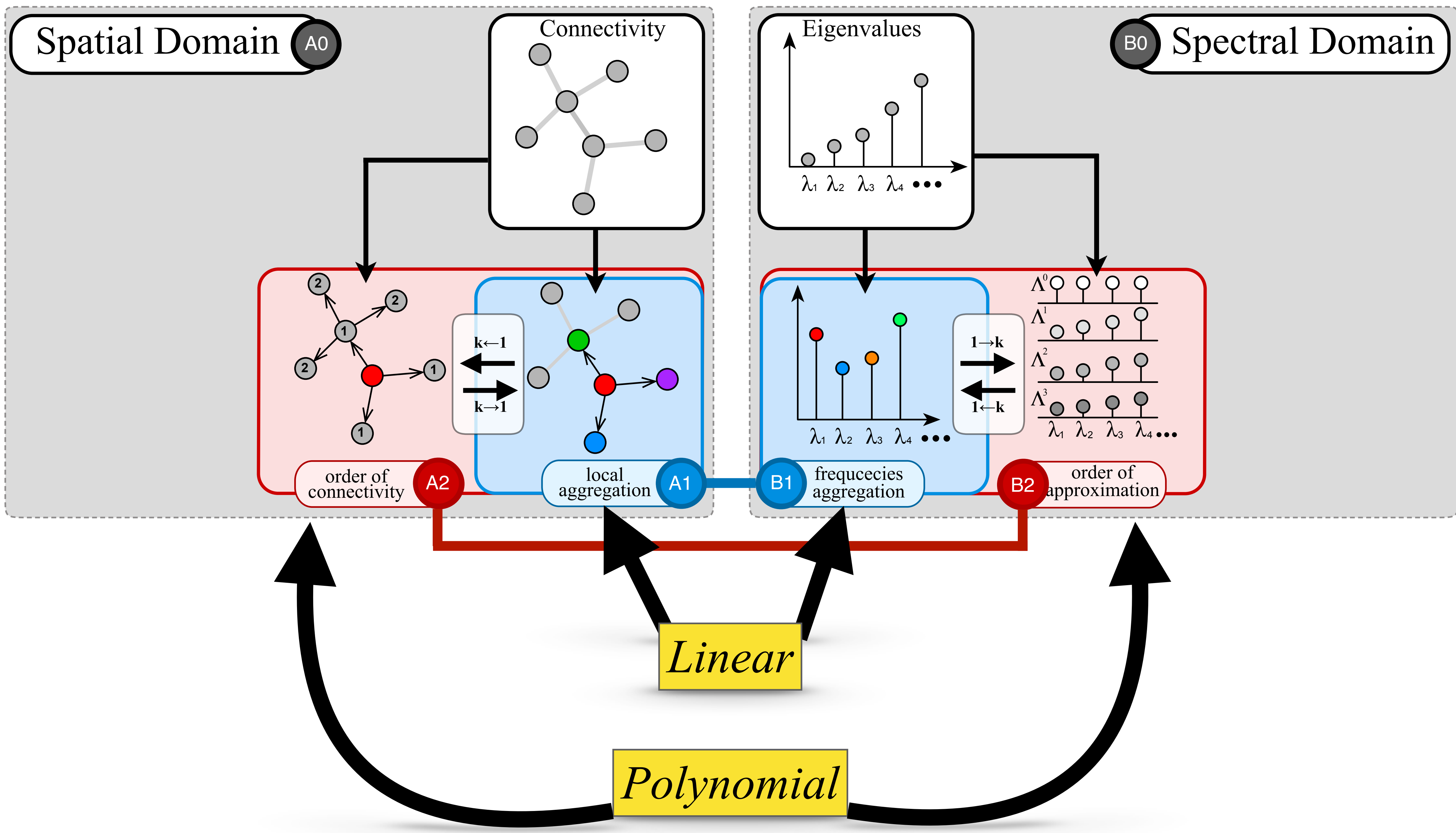
Polynomial



$y = x^5 - 3x^4 + 2x^3 - 0.3x^2 + x + 1$



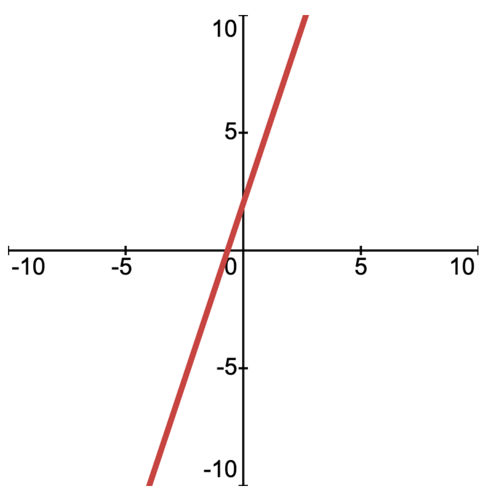
Linear v.s. Polynomial



Beyond Polynomial: Rational Model

GCN

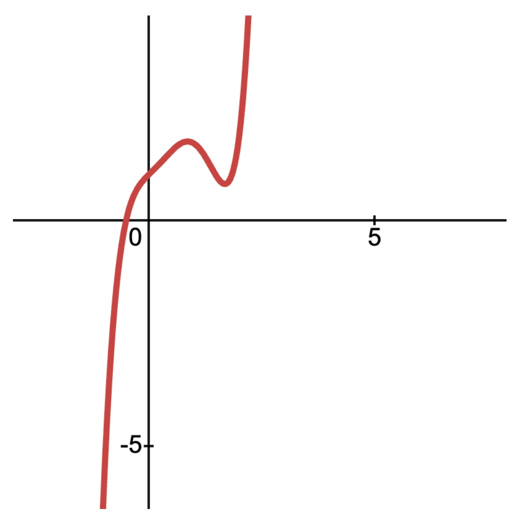
$$y = 3x + 2$$



Linear

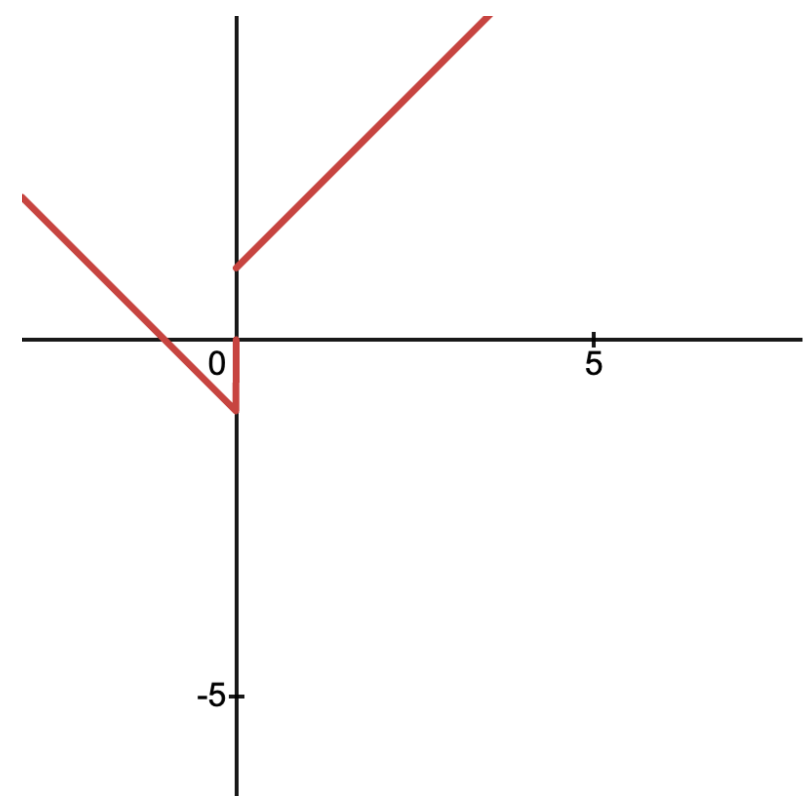
ChebNet

$$y = x^5 - 3x^4 + 2x^3 - 0.3x^2 + x + 1$$



Polynomial

What if non-smooth function



Defferrard, et al., "**Convolutional neural networks on graphs with fast localized spectral filtering.**" Advances in neural information processing systems 29 (2016).

Beyond Polynomial: Rational Model

Polynomial approximation

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

Cheaper
Less accurate

simple form, well known properties
computationally **easy** to use

notorious for oscillations between exact-fit value
only high degree can model **complicated** structure
poor interpolatory/extrapolatory/asymptotic properties

Rational approximation

$$f(x) = \frac{p(x)}{q(x)}$$

More expensive
More accurate

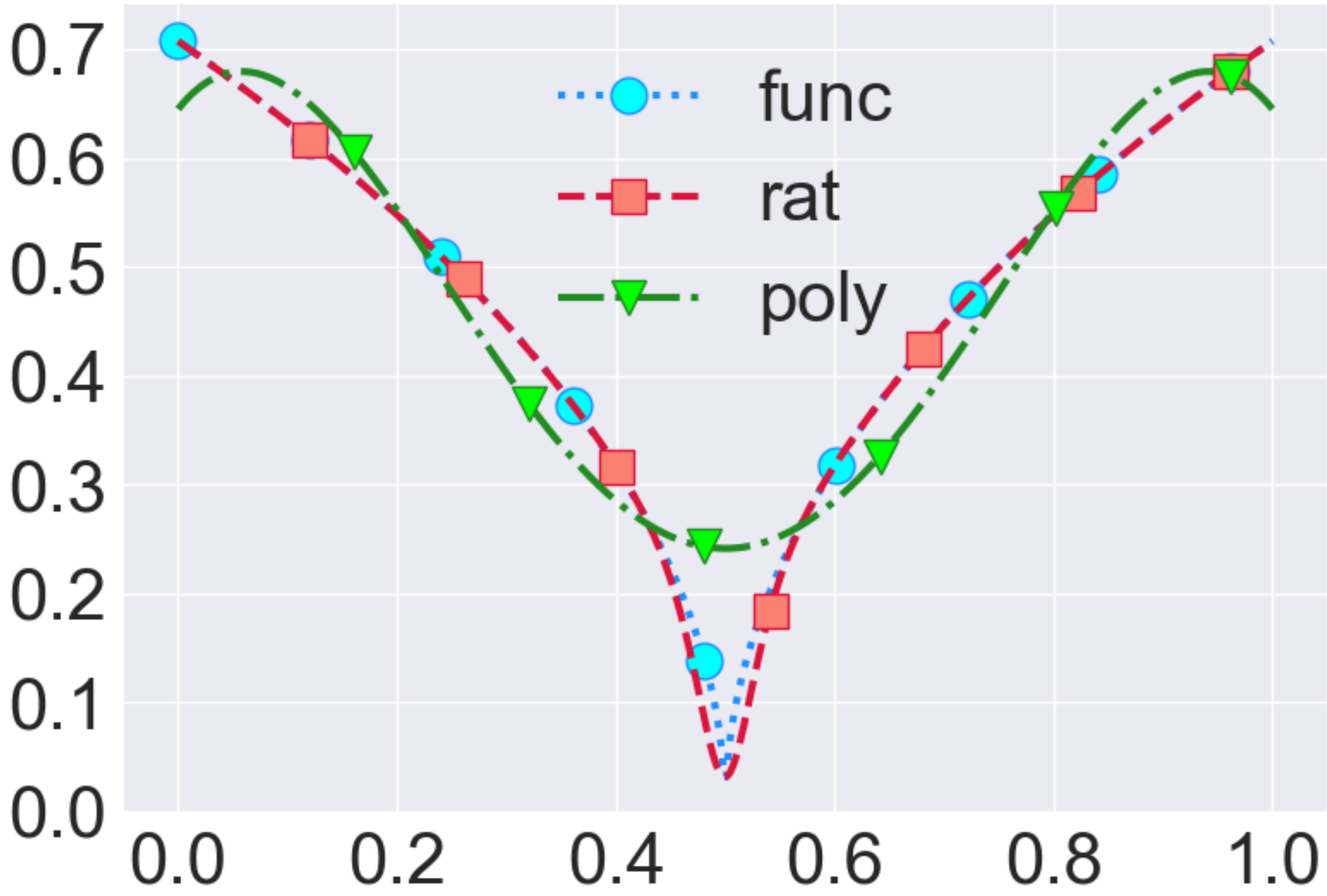
moderately **simple** form, not well-known properties
moderately **easy** to handle computationally

excellent for oscillations between exact-fit value
model **complicated** structure with a fairly low degree
excellent interpolatory/extrapolatory/asymptotic properties

Trefethen, Lloyd N. ***Approximation theory and approximation practice***, *extended edition*. Society for Industrial and Applied Mathematics, 2019.

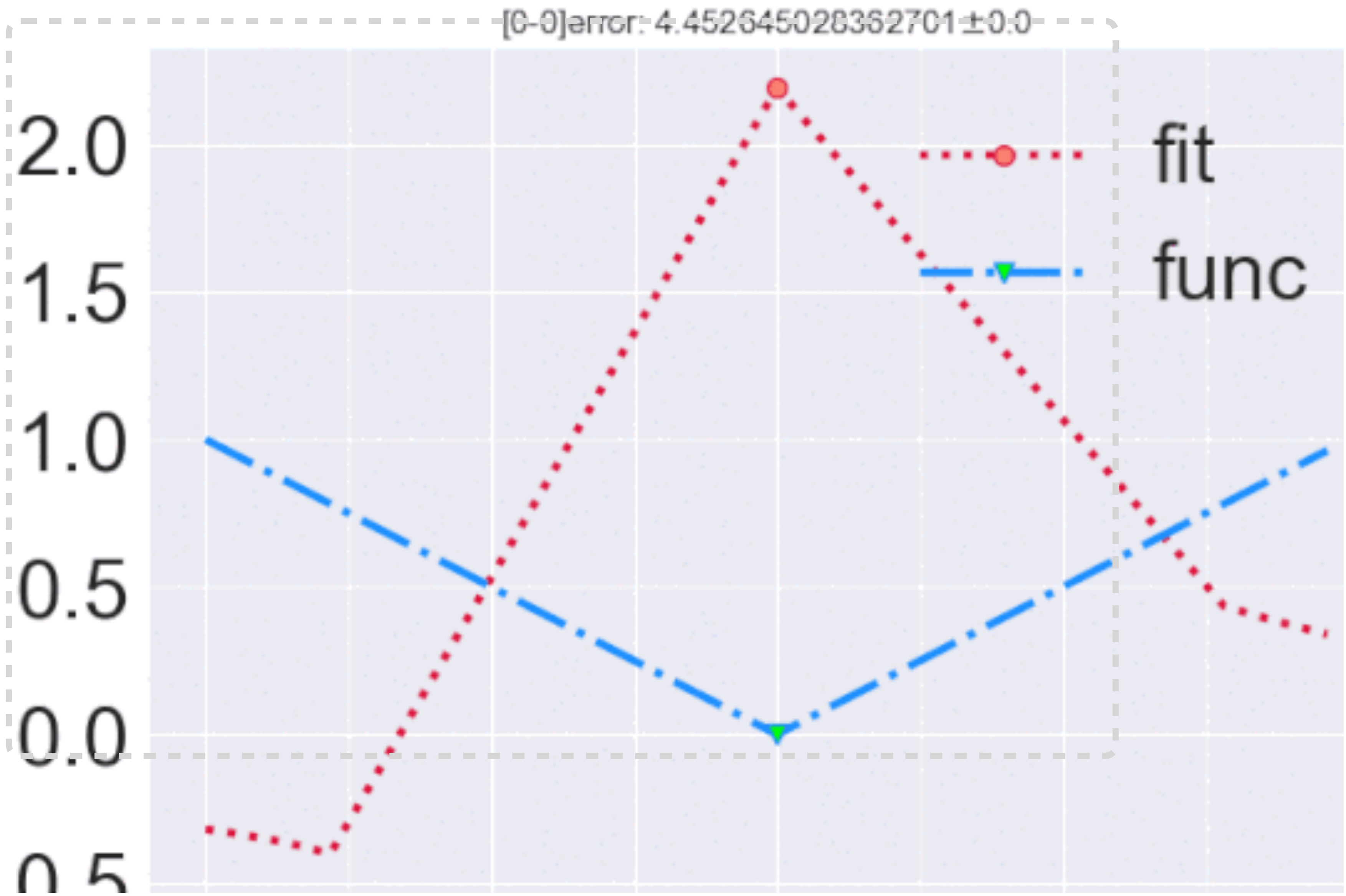
Beyond Polynomial: Rational Model

func: target function;
poly: polynomial approximation
rat: rational approximation



Telgarsky, M. (2017, July). ***Neural networks and rational functions***. *In International Conference on Machine Learning*

Beyond Polynomial: Rational Model



[Screenshots in every 100 epochs]

Rational Neural Network: iteratively close to the target

*Boullé, N., Nakatsukasa, Y., & Townsend, A. (2020). **Rational neural networks**. Advances in neural information processing systems*

*Zhiqian Chen, et al. **Rational Neural Networks for Approximating Graph Convolution Operator on Jump Discontinuities**, ICDM 2018*

Polynomial v.s. Rational

Polynomial

Thomas N. Kipf et al. (2016)

$$\mathbf{g} * x = \mathbf{U}\,\mathbf{g}(\mathbf{\Lambda})\,\mathbf{U}^\intercal x$$

$$\approx \mathbf{U} \sum_k \theta_k T_k(\tilde{\mathbf{\Lambda}})\,\mathbf{U}^\intercal x \qquad (\tilde{\mathbf{\Lambda}} = \frac{2}{\lambda_{max}}\mathbf{\Lambda} - \mathbf{I_N})$$

$$= \sum_k \theta_k T_k(\tilde{\mathbf{L}}) x \qquad (\mathbf{U}\,\mathbf{\Lambda}^k\,\mathbf{U}^\intercal = (\mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^\intercal)^k)$$

$$= \mathbf{P}(\mathbf{L})\, x$$

Rational

Z. Chen et al. (2018)

$$\mathrm{g}_\theta * x = \mathbf{U}\,\mathrm{g}_\theta\,\mathbf{U}^\intercal x \qquad \text{(convolution theorem)}$$

$$\approx \mathbf{U} \frac{\sum_{i=0}^{m} \psi_i \tilde{\mathbf{\Lambda}}^i}{1 + \sum_{j=1}^{n} \phi_j \tilde{\mathbf{\Lambda}}^j} \mathbf{U}^\intercal x \qquad (\tilde{\mathbf{\Lambda}} = \frac{\mathbf{\Lambda}}{\lambda_{max}})$$

$$= \mathbf{U} \frac{\mathbf{P}(\mathbf{\Lambda})}{\mathbf{Q}(\mathbf{\Lambda})} \mathbf{U}^\intercal x, \qquad \text{(define P and Q)}$$

$$= \mathbf{P}(\mathbf{L})\,\mathbf{Q}(\mathbf{L})^{-1} x.$$

$\mathbf{L}$ graph Laplacian

x node attributes

$$= \boxed{\mathbf{P}(\mathbf{L})}\,\boxed{x}$$

$$= \boxed{\mathbf{P}(\mathbf{L})}\,\boxed{\mathbf{Q}(\mathbf{L})^{-1}}\,\boxed{x}$$

Polynomial v.s. Rational

Polynomial

$\boxed{\mathbf{P}(\mathbf{L})}\ x$

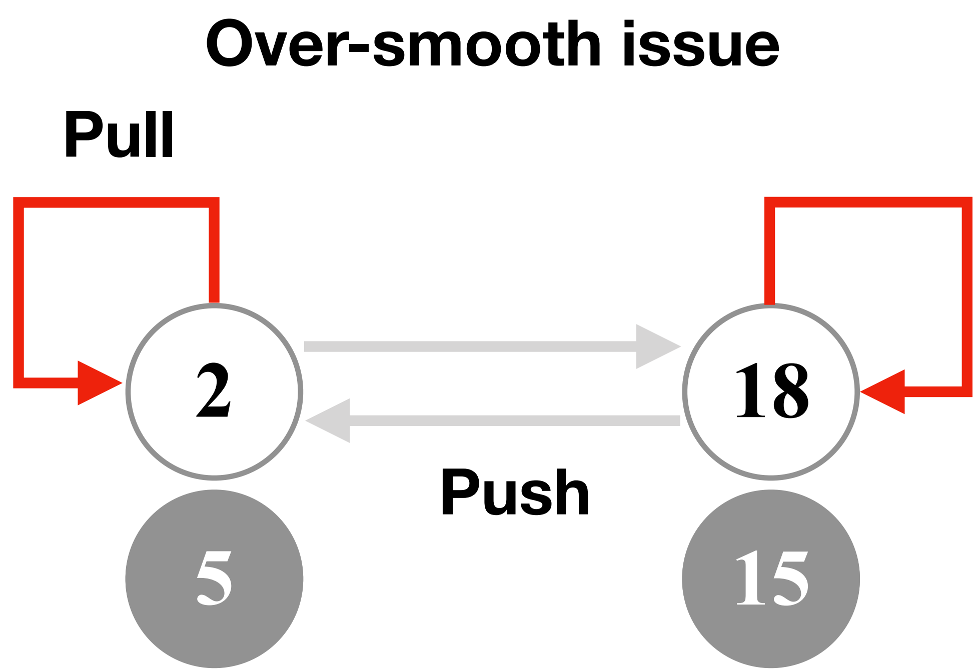
Label propagation

Rational

$\boxed{\mathbf{P}(\mathbf{L})}\boxed{\mathbf{Q}(\mathbf{L})^{-1}}\ x$

Label propagation

Reverse Label propagation



Polynomial v.s. Rational

Polynomial

$\boxed{\mathbf{P}(\mathbf{L})}\ x$

Label propagation

Rational

$\boxed{\mathbf{P}(\mathbf{L})}\boxed{\mathbf{Q}(\mathbf{L})^{-1}}\ x$

Reverse Label propagation

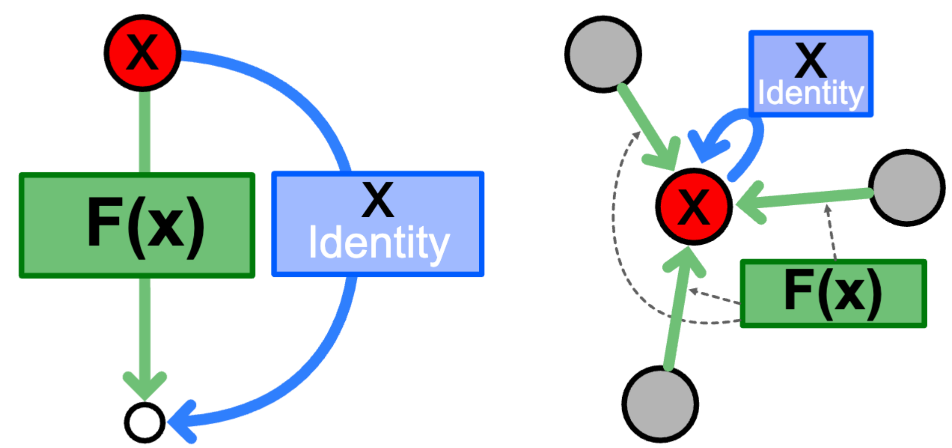


FIG. 6. Left: Residual Learning $x' = F(x) + x$; Right: Rational Aggregation: $x' = F(x) + x$

Polynomial v.s. Rational

Polynomial

$\boxed{\mathbf{P}(\mathbf{L})}\ x$

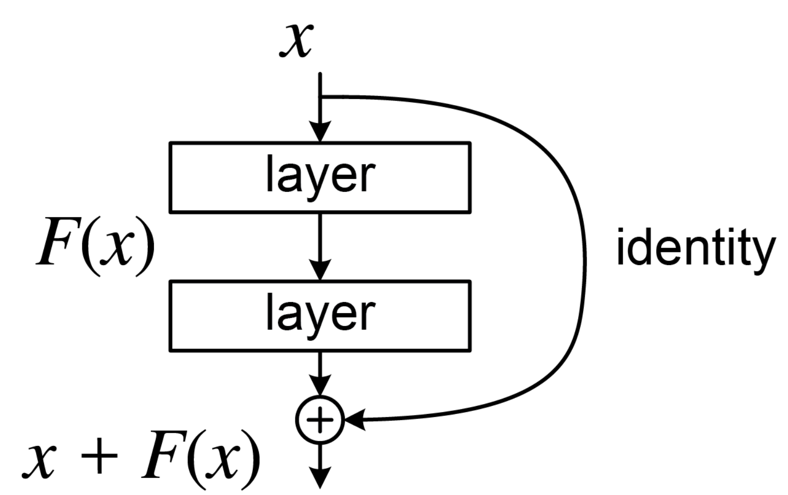
Label propagation

Difference?

Rational

$\boxed{\mathbf{P}(\mathbf{L})}\boxed{\mathbf{Q}(\mathbf{L})^{-1}}\ x$

Reverse Label propagation



Multi-layer Graph Convolution

Polynomial v.s. Rational

Polynomial

$\boxed{\mathbf{P}(\mathbf{L})}\ x$

Label propagation

Rational

$\boxed{\mathbf{P}(\mathbf{L})}\boxed{\mathbf{Q}(\mathbf{L})^{-1}}\ x$

Reverse Label propagation

Polynomial Implementation

Single layer $x + A \cdot x$

Multiple-layer $x' = x + A \cdot x \rightarrow x'' = \boxed{x'} + A \cdot x'$

*Add the **last** f at each iteration*

Rational Implementation

Single layer $x + A \cdot x$

Multiple-layer $x' = x + A \cdot x \rightarrow x'' = \boxed{x} + A \cdot x'$

*Add the **original** f at each iteration*

Beyond Polynomial: Rational Model

Johannes Klicpera et al. (2018)

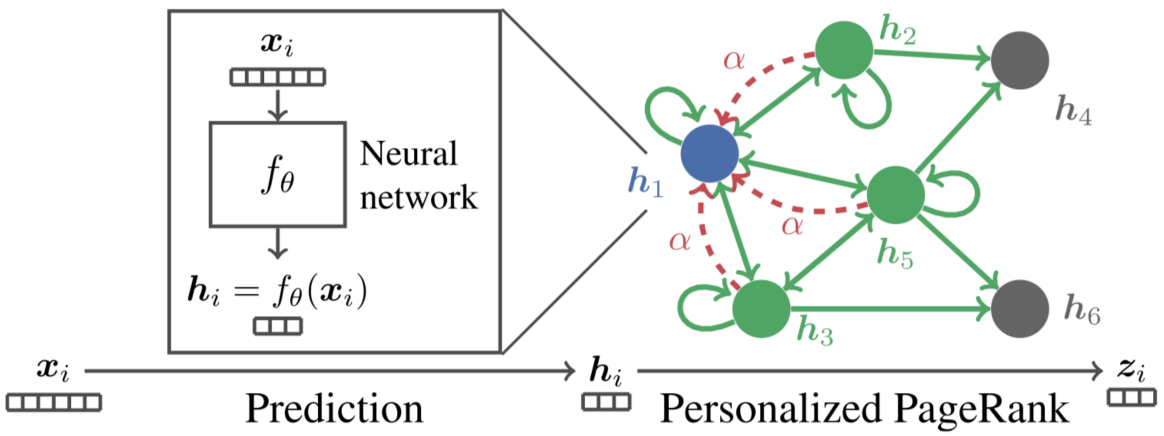
Personalized Page Rank (information retrieval)

$$\boldsymbol{\pi}_{\text{ppr}}(\boldsymbol{i}_x) = (1-\alpha)\hat{\tilde{\boldsymbol{A}}}\boldsymbol{\pi}_{\text{ppr}}(\boldsymbol{i}_x) + \alpha \boldsymbol{i}_x$$

$(1-\alpha)$ α

PPNP

Personalized Propagation of Neural Predictions



Use personalized PageRank matrix $\boldsymbol{\Pi}_{\text{ppr}}$ to propagate further while retaining information about root node, adjust via teleport probability α:

$$\boldsymbol{\Pi}_{\text{ppr}} = \alpha\left(\boldsymbol{I}_n - (1-\alpha)\hat{\tilde{\boldsymbol{A}}}\right)^{-1}$$

$$\frac{\alpha}{1-(1-\alpha)\lambda}$$

Beyond Polynomial: Rational Model

Johannes Klicpera et al. (2018)

Personalized Page Rank (information retrieval)

$$\boldsymbol{\pi}_{\text{ppr}}\left(\boldsymbol{i}_x\right) = (1-\alpha)\hat{\tilde{\mathbf{A}}}\boldsymbol{\pi}_{\text{ppr}}\left(\boldsymbol{i}_x\right) + \alpha\boldsymbol{i}_x$$

$(1-\alpha)$ α

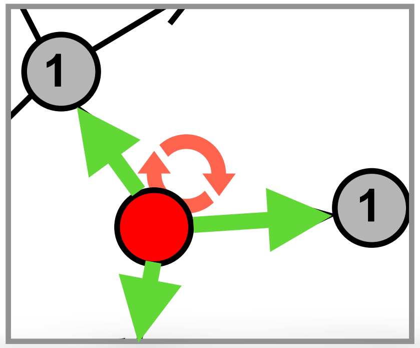
Filippo Maria Bianchi et al. (2018)

ARMA (time series)

$$\bar{\mathbf{X}}^{(t+1)} = a\mathbf{M}\bar{\mathbf{X}}^{(t)} + b\mathbf{X}$$

a b

$$\textit{next} = \alpha\ \textit{Push} + \beta\ \textit{Pull}$$



One iteration

K iterations →K-order rational function

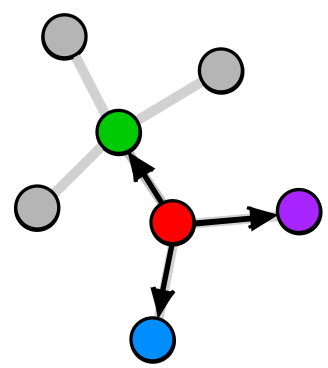
Why Rational, and Why Not?

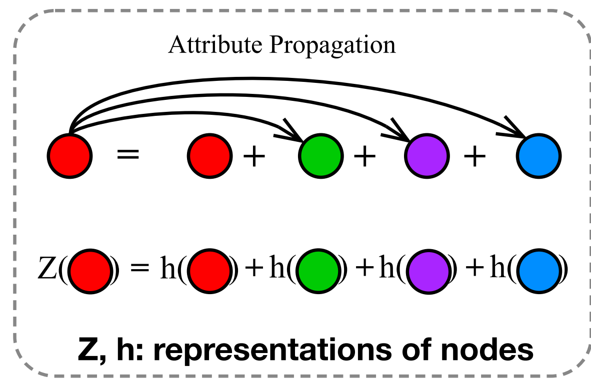
- **Yes**
  - Non-smooth functions (Spectral) Physical meaning of non-smooth func in spatial?
  - Avoid over-smoothing (Spatial)
  - Approximation theory
    - rational is better than polynomial when order ≥ 5 Imply 5 iterations/layers
- **No**
  - Computational Complexity: Matrix Inversion $\mathcal{O}(n^3)$

Spatial-based GNN

Function of **graph matrix** $g(A)X$

Linear





GCN *Thomas N. Kipf et al. (2016)*

$$\mathbf{Z} = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X} = \hat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

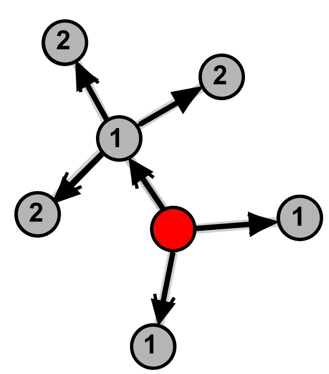
GraphSAGE *Will Hamilton et al. (2017)*

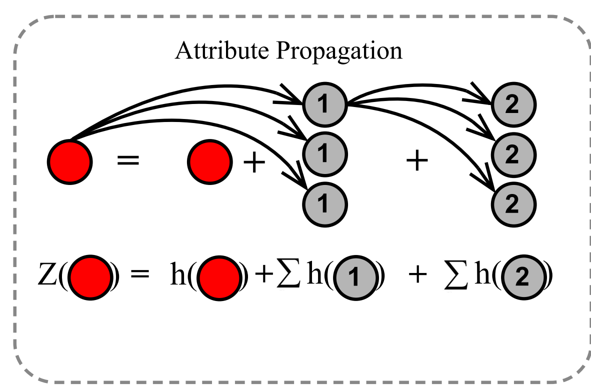
$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

GIN *Xukeyu Lu et al. (2019)*

$$\mathbf{Z} = (1+\epsilon)\cdot\mathbf{h}(v) + \sum_{u_j\in\mathcal{N}(v_i)}\mathbf{h}(u_j) = [(1+\epsilon)\mathbf{I}+\mathbf{A}]\mathbf{X}$$

Polynomial





DeepWalk *Bryan Perozzi et al. (2014)*

$$\mathbf{Z} = \frac{1}{t+1}\left(\mathbf{I}+\tilde{\mathbf{A}}+\tilde{\mathbf{A}}^2+\ldots+\tilde{\mathbf{A}}^t\right)\mathbf{X} = \frac{1}{t+1}\mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

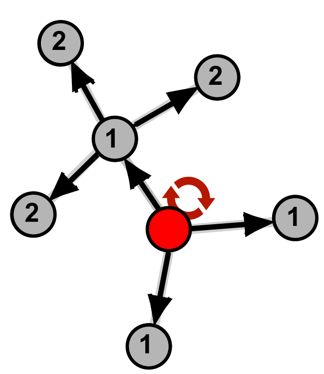
ChebyNet *Defferrard, Michael et al. (2016)*

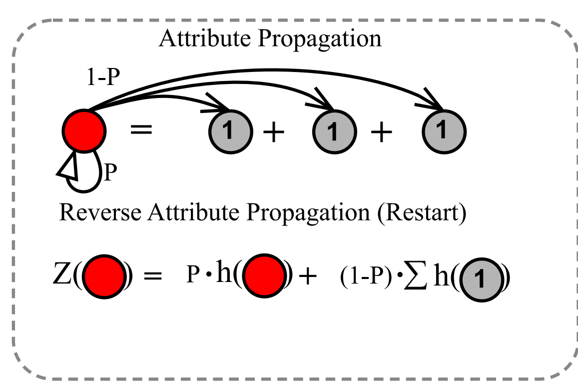
$$\mathbf{Z} = \sum_{k=0}^{K-1}\theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \left[\tilde{\theta}_0\mathbf{I}+\tilde{\theta}_1(\mathbf{I}-\tilde{\mathbf{A}})+\tilde{\theta}_2(\mathbf{I}-\tilde{\mathbf{A}})^2+\ldots\right]\mathbf{X} = \left(\phi\mathbf{I}+\sum_{i=1}^{k}\psi_i\tilde{\mathbf{A}}^i\right)\mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

Node2Vec *Aditya Grover et al. (2016)*

$$\mathbf{Z} = \left(\frac{1}{p}\cdot\mathbf{I}+\tilde{\mathbf{A}}+\frac{1}{q}\left(\tilde{\mathbf{A}}^2-\tilde{\mathbf{A}}\right)\right)\mathbf{X} = \left[\frac{1}{p}\mathbf{I}+\left(1-\frac{1}{q}\right)\tilde{\mathbf{A}}+\frac{1}{q}\tilde{\mathbf{A}}^2\right]\mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

Rational





Personalized PageRank *Johannes Klicpera et al. (2018)*

$$\mathbf{Z} = \frac{\alpha}{\mathbf{I}-(1-\alpha)\tilde{\mathbf{A}}}\mathbf{X}$$

ARMA Filter *Filippo Maria Bianchi et al. (2018)*

$$\mathbf{Z} = \frac{b}{\mathbf{I}-a\tilde{\mathbf{A}}}\mathbf{X}$$

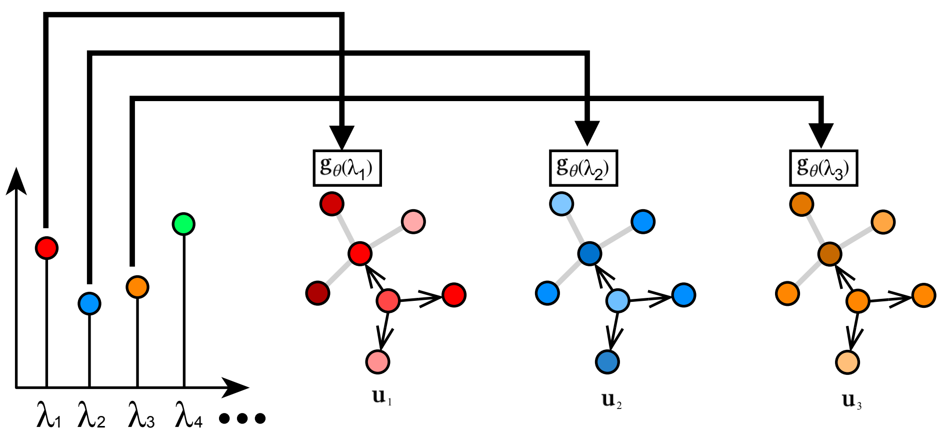
Auto Regressive Filter *Qimai Li et al. (2019)*

$$\mathbf{Z} = (\mathbf{I}+\alpha\tilde{\mathbf{L}})^{-1}\mathbf{X} = \frac{\mathbf{I}}{\mathbf{I}+\alpha(\mathbf{I}-\tilde{\mathbf{A}})}\mathbf{X}$$

Spectral-based GNN

Function of **eigenvalue** $Ug_\theta(\Lambda)U^TX$

Linear



GCN *Thomas N. Kipf et al. (2016)*

$$\mathbf{Z} = \tilde{\mathbf{A}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{L}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}-\mathbf{L}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2-\Lambda)\mathbf{U}^\top\mathbf{X}$$

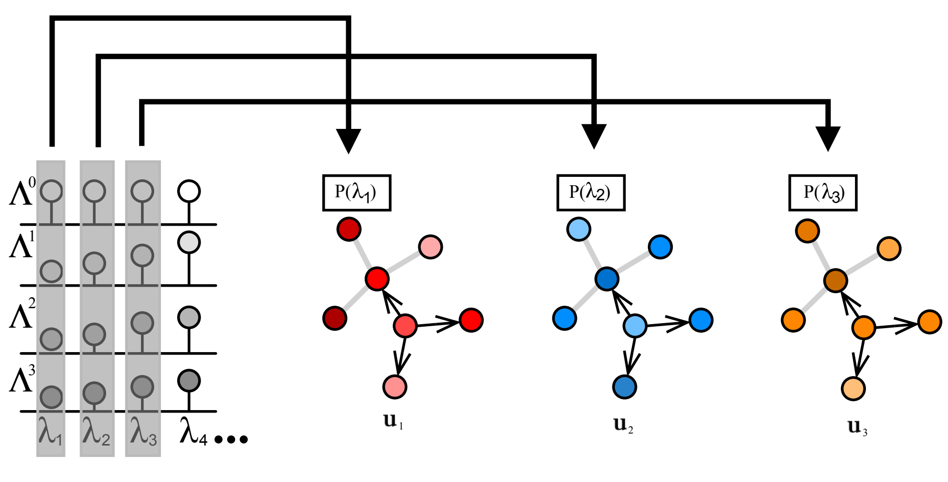
GraphSAGE *Will Hamilton et al. (2017)*

$$\mathrm{Z} = \mathrm{D}^{-\frac{1}{2}}(\mathrm{I}+\mathrm{A})\mathrm{D}^{-\frac{1}{2}}\mathrm{X} = (\mathrm{I}+\tilde{\mathrm{A}})\mathrm{X} = (2\mathrm{I}-\tilde{\mathrm{L}})\mathrm{X} = \mathrm{U}(2-\Lambda)\mathrm{U}^\top\mathrm{X}$$

GIN *Xukeyu Lu et al. (2019)*

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}[(1+\epsilon)\mathbf{I}+\mathbf{A}]\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}[(2+\epsilon)\mathbf{I}-\tilde{\mathrm{L}}]\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2+\epsilon-\Lambda)\mathbf{U}^\top\mathbf{X}$$

Polynomial



DeepWalk *Bryan Perozzi et al. (2014)*

$$\mathbf{Z} = \frac{1}{t+1}\left(\mathbf{I}+(\mathbf{I}-\tilde{\mathbf{L}})+(\mathbf{I}-\tilde{\mathbf{L}})^2+\ldots+(\mathbf{I}-\tilde{\mathbf{L}})^t\right)\mathbf{X} = \mathbf{U}\left(\theta_0+\theta_1\boldsymbol{\Lambda}+\theta_2\boldsymbol{\Lambda}^2+\ldots+\theta_t\boldsymbol{\Lambda}^t\right)\mathbf{U}^\top\mathbf{X}$$

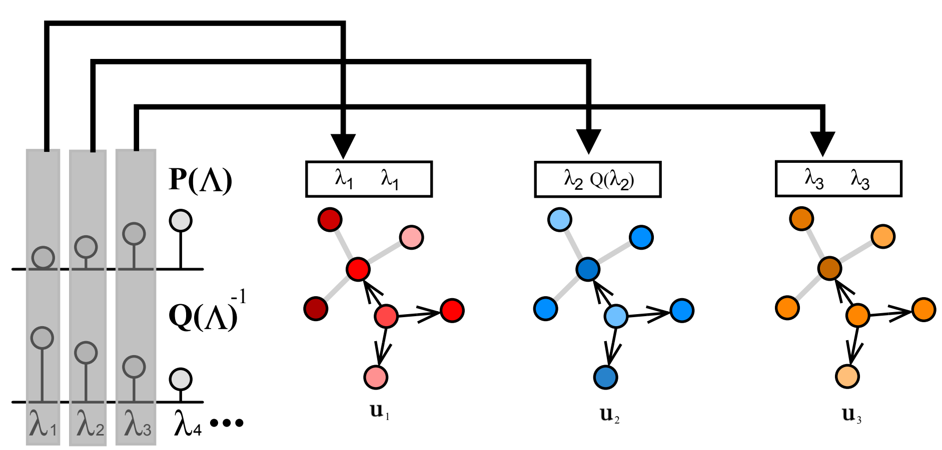
ChebyNet *Defferrard, Michael et al. (2016)*

$$\mathbf{Z} = \sum_{k=0}^{K-1}\theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \mathbf{U}\left(\tilde{\theta}_0\cdot 1+\tilde{\theta}_1\Lambda+\tilde{\theta}_2\Lambda^2+\ldots\right)\mathbf{U}^\top\mathbf{X}$$

Node2Vec *Aditya Grover et al. (2016)*

$$\mathbf{Z} = \left[\left(1+\frac{1}{p}\right)\mathbf{I}-\left(1+\frac{1}{q}\right)\tilde{\mathbf{L}}+\frac{1}{q}\tilde{\mathbf{L}}^2\right]\mathbf{X} = \mathbf{U}\left[\left(1+\frac{1}{p}\right)-\left(1+\frac{1}{q}\right)\tilde{\Lambda}+\frac{1}{q}\tilde{\Lambda}^2\right]\mathbf{U}^\top\mathbf{X}$$

Rational



Personalized PageRank *Johannes Klicpera et al. (2018)*

$$\mathbf{Z} = \frac{\alpha}{\mathbf{I}-(1-\alpha)(\mathbf{I}-\tilde{\mathrm{L}})}\mathbf{X} = \mathbf{U}\frac{\alpha}{\alpha\mathbf{I}+(1-\alpha)\Lambda}\mathbf{U}^\top\mathbf{X}$$

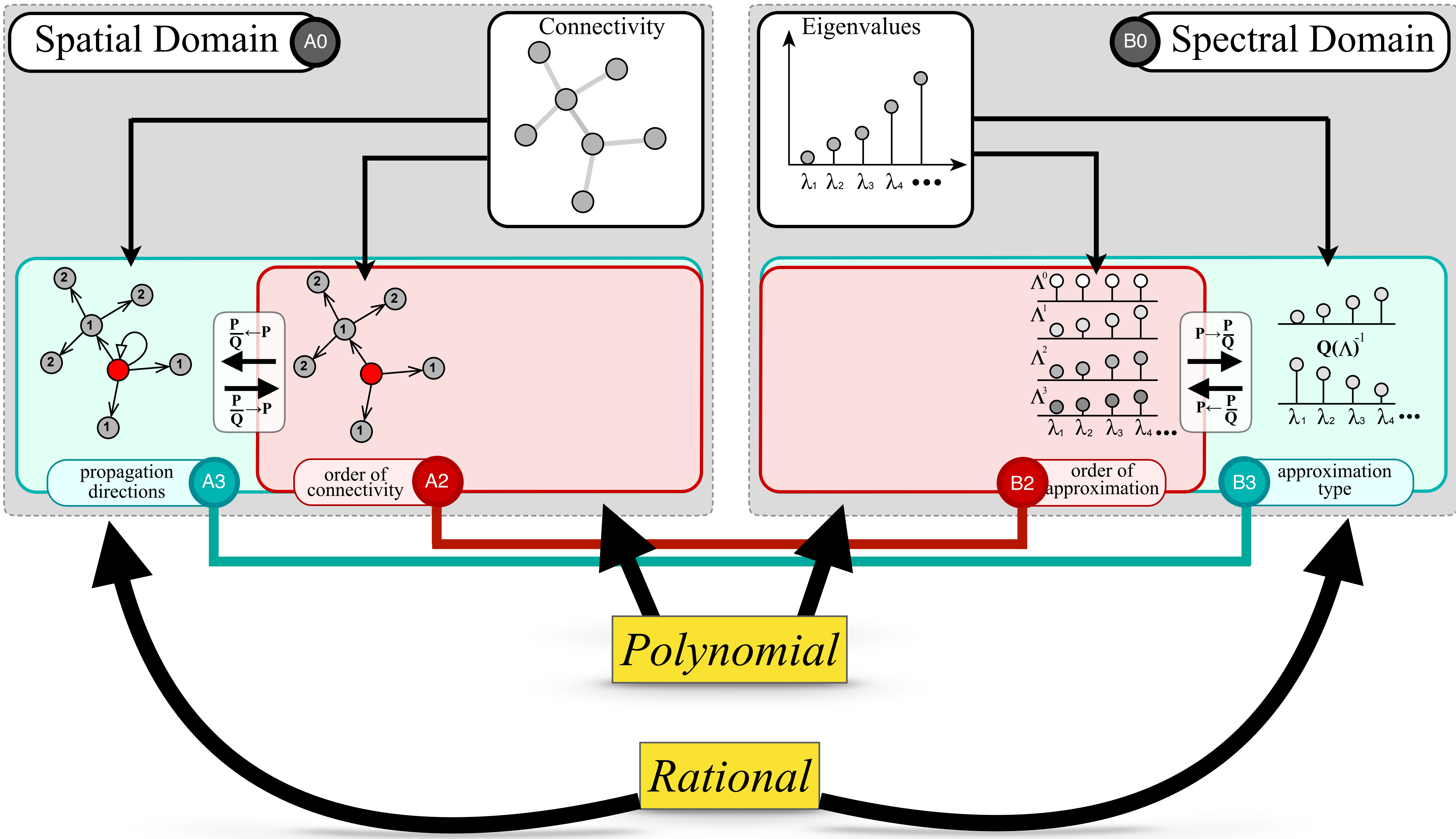
ARMA Filter *Filippo Maria Bianchi et al. (2018)*

$$\mathbf{Z} = \frac{b}{1-a(\mathbf{I}-\tilde{\mathbf{L}})}\mathbf{X} = \mathbf{U}\frac{b}{(1-a)\mathrm{I}+a\Lambda}\mathbf{U}^\top\mathbf{X}$$

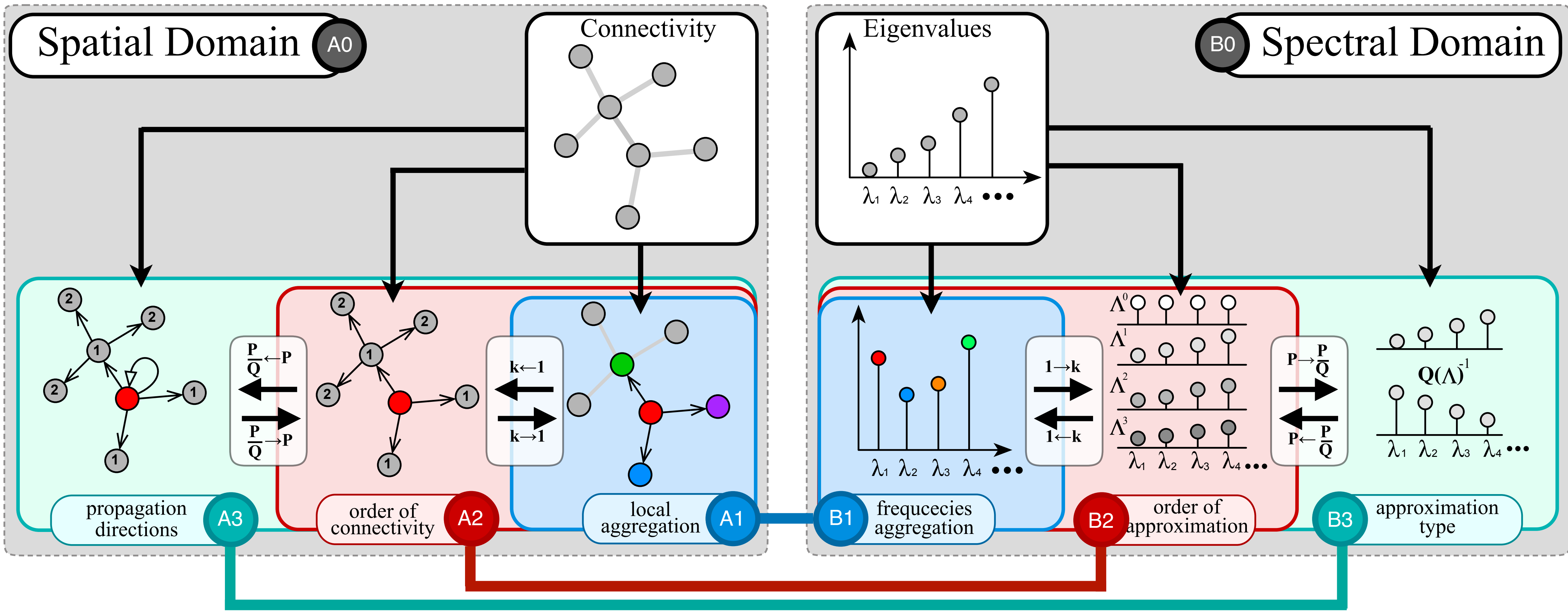
Auto Regressive Filter *Qimai Li et al. (2019)*

$$\mathbf{Z} = (\mathbf{I}+\alpha\tilde{\mathbf{L}})^{-1}\mathbf{X} = \mathbf{U}\frac{1}{1+\alpha(1-\Lambda)}\mathbf{U}^\top\mathbf{X}$$

Rational v.s. Polynomial



The Unified Framework



	Linear	Polynomial	Rational
Spatial			
Spectral			

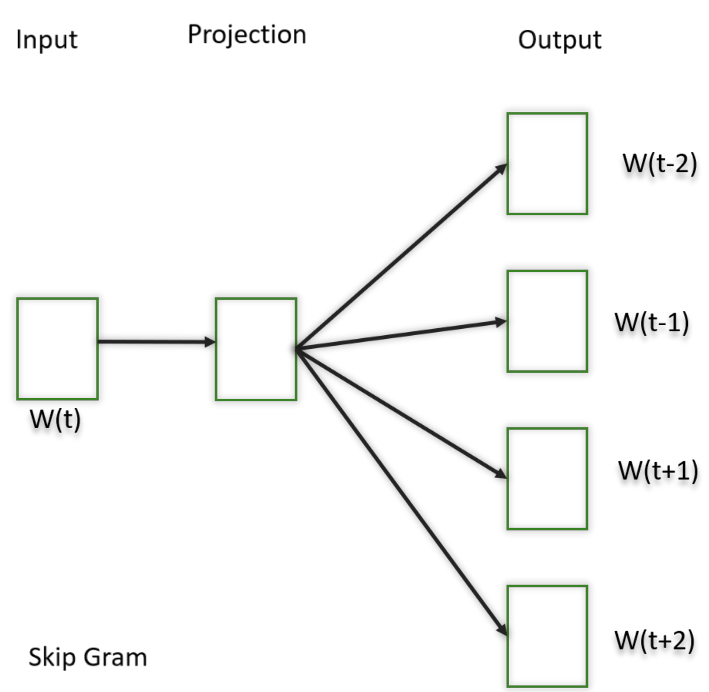
Complex, Powerful

Spatial v.s. Spectral

Spatial, local ⟷ **Spectral, global**

Word2Vec

Tomas Mikolvet et al. (2013)



W2V as Implicit MF

Omer Levy et al. (2014)

$$WC^T = M$$

Shifted PMI (co-occurrence matrix)

Matrix Factorization, O(n^3)

I am similar to neighbors

Co-Occurrence matrix decomposition

Levy, Omer, and Yoav Goldberg. "**Neural word embedding as implicit matrix factorization.**" Advances in neural information processing systems 27 (2014).

Spatial v.s. Spectral

Spatial, local ⟷ **Spectral, global**

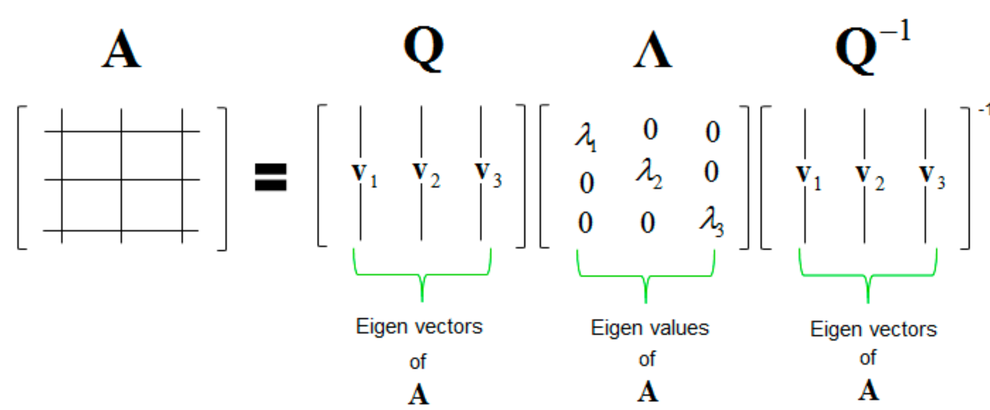
SpectralNet

Uri Shaham et al. (2018)

$$L_{\text{SpectralNet}}(\theta) = \frac{1}{m^2} \sum_{i,j=1}^{m} W_{i,j} \|y_i - y_j\|^2,$$

$$\mathbb{E}\left[yy^T\right] = I_{k \times k}$$

Spectral Clustering



Spectral Decomposition

Complexity	Spatial	Spectral
Space	Only involves local neighbors each time	Matrix factorization takes more
Time	Many iterations, trade-off between #iteration vs convergence	One-time expensive matrix factorization

Shaham, Uri, et al. "**Spectralnet: Spectral clustering using deep neural networks.**" ICLR (2018).

Spatial v.s. Spectral

	Methodology	Computation	Space Complexity	Stability
Spectral	Global	One-step	High	Exact
Spatial	Local	Iterative	Low	Approximate

Agenda

- **First Half (1 hour 30 min)**
  - *Background*: **unified frameworks for GNN** (45 min)
  - *Preliminary*: **graph convolutions** (45 min)
  - BREAK (30 min)
- **Second Half (1 hour)**
  - *Introduction*: **a new unified framework** (40 min)
  - *Future directions* (20min)
  - Q&A (15 min)

Future Directions

- PDE
  - Waves v.s. Diffusions is similar to Rational v.s. Polynomial

2.5 COMPARISON OF WAVES AND DIFFUSIONS

Property		Waves	Diffusions
(i)	Speed of propagation?	Finite ($\leq c$)	Infinite
(ii)	Singularities for $t > 0$?	Transported along characteristics (speed $= c$)	Lost immediately
(iii)	Well-posed for $t > 0$?	Yes	Yes (at least for bounded solutions)
(iv)	Well-posed for $t < 0$?	Yes	No
(v)	Maximum principle	No	Yes
(vi)	Behavior as $t \to +\infty$?	Energy is constant so does not decay	Decays to zero (if ϕ integrable)
(vii)	Information	Transported	Lost gradually

Rational Polynomial

Strauss, W. A. (2007). ***Partial differential equations: An introduction***. John Wiley & Sons.

Future Directions

- Spectral graph beyond simple graph
  - Signed
  - **Directed**
  - Higher-order
    - hypergraph
    - simplicial complex: hodge and homology theory
  - Fractional Laplacian

Levie, Ron, et al. "***Transferability of spectral graph convolutional neural networks.***" *Journal of Machine Learning Research 22.272 (2021): 1-59.*

Future Directions

- Hodge Laplacian

$$L_k := L_k^{\textbf{down}} + L_k^{\textbf{up}}$$

$$L_k^{\textbf{down}} := B_k^\top B_k$$

$$L_k^{\textbf{up}} := B_{k+1} B_{k+1}^\top$$

L_1 is normal graph Laplacian

- GCN: 0**st-order** information propagate over 1**nd-order** connectivity

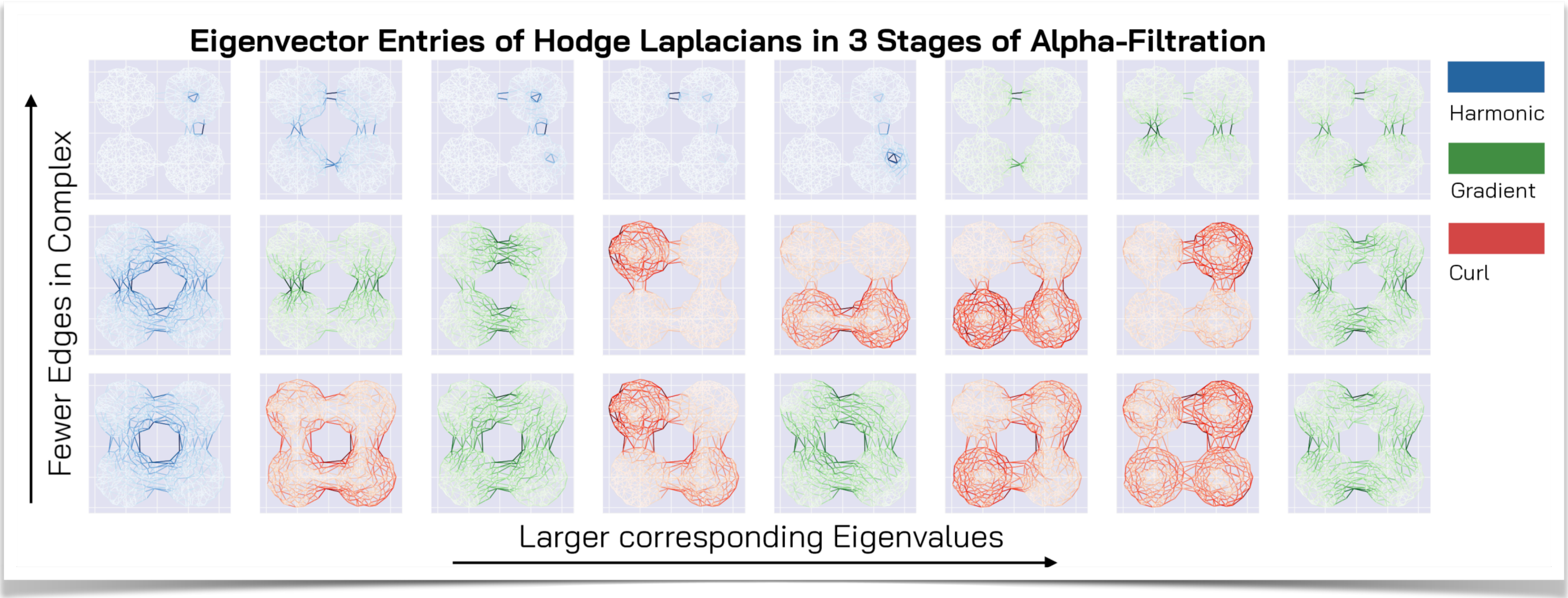
Function of **graph matrix** $g(A)X$

- xGCN: (x)**st-order** information propagate over (x+1)**nd-order** connectivity

Grande, Vincent P., and Michael T. Schaub. "Disentangling the Spectral Properties of the Hodge Laplacian: not all small Eigenvalues are Equal." ICASSP 2024. IEEE, 2024.

Future Directions

- Hodge decomposition
  - Decompose dynamics into 3 categories



Grande, Vincent P., and Michael T. Schaub. "**Disentangling the Spectral Properties of the Hodge Laplacian: not all small Eigenvalues are Equal.**" ICASSP 2024. IEEE, 2024.

Future Directions

- Quantum Computing for Spectral Method
  - Quantum algorithms such as the Quantum Phase Estimation (QPE) algorithm can be used to find the eigenvalues and eigenvectors of a matrix more efficiently than classical algorithms

- *Classical Algorithm:* $\mathcal{O}(n^3)$
- *QPE:* $\mathcal{O}((log(n))^2/\epsilon)$ *with precision* ϵ

Conclusion

- Connection between spectral and spatial domain
  - Spatial: function of adjacency matrix
  - Spectral: function of eigenvalues
- Linear, polynomial and rational function
  - more power, more computation
- Computation
  - Spatial method: iterative and cheap approximation
  - Spectral method: one-step, expensive and exact

Thank You & Q/A

Awesome Spectral Graph Neural Networks

PRs Welcome awesome

Contents

https://github.com/XGraph-Team/Spectral-Graph-Survey