



MDS 2024'

Unifying Spectral and Spatial Graph Neural Networks

Chen, Zhiqian
Assistant Professor

Mississippi State University



Zhang, Lei
Assistant Professor

Northern Illinois University



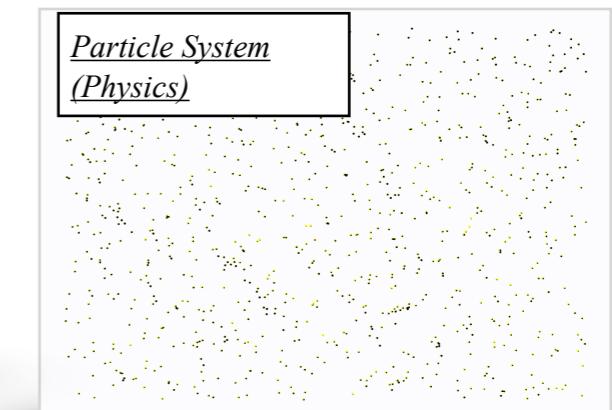
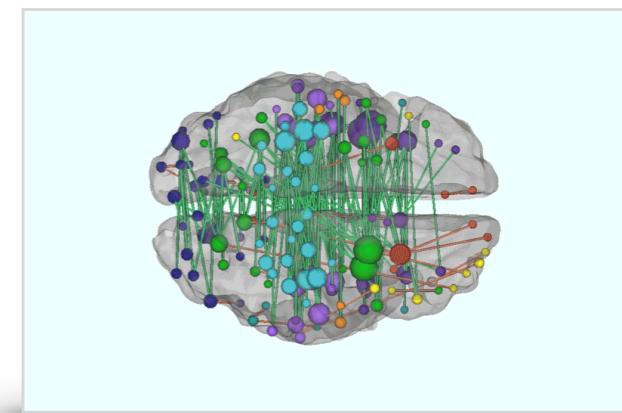
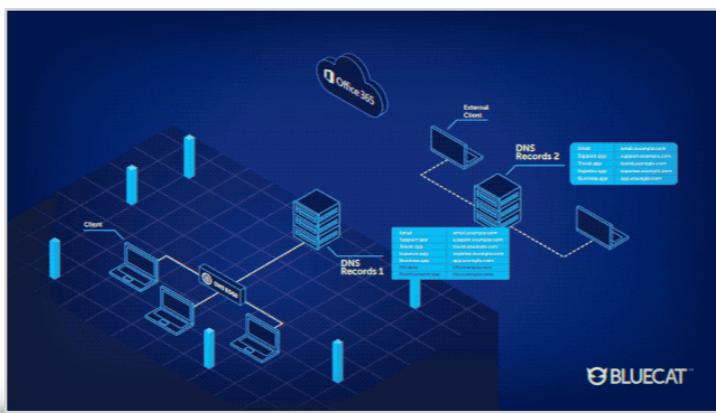
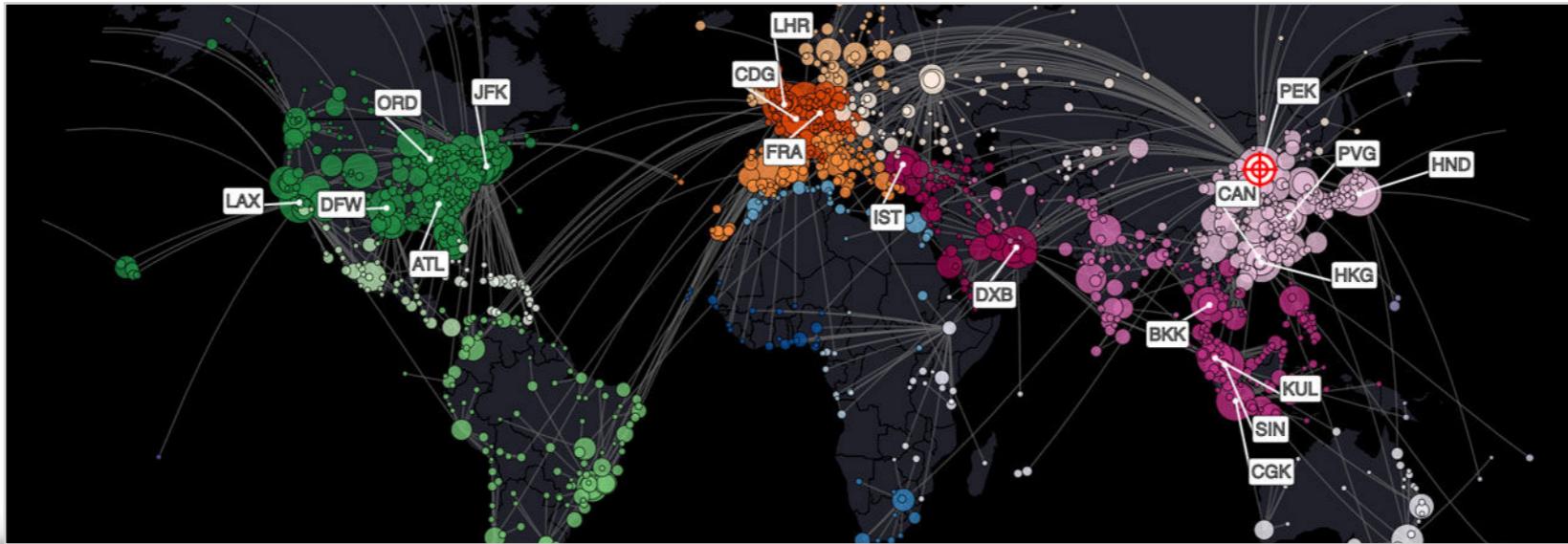
Northern Illinois
University

Zhao, Liang
Associate Professor

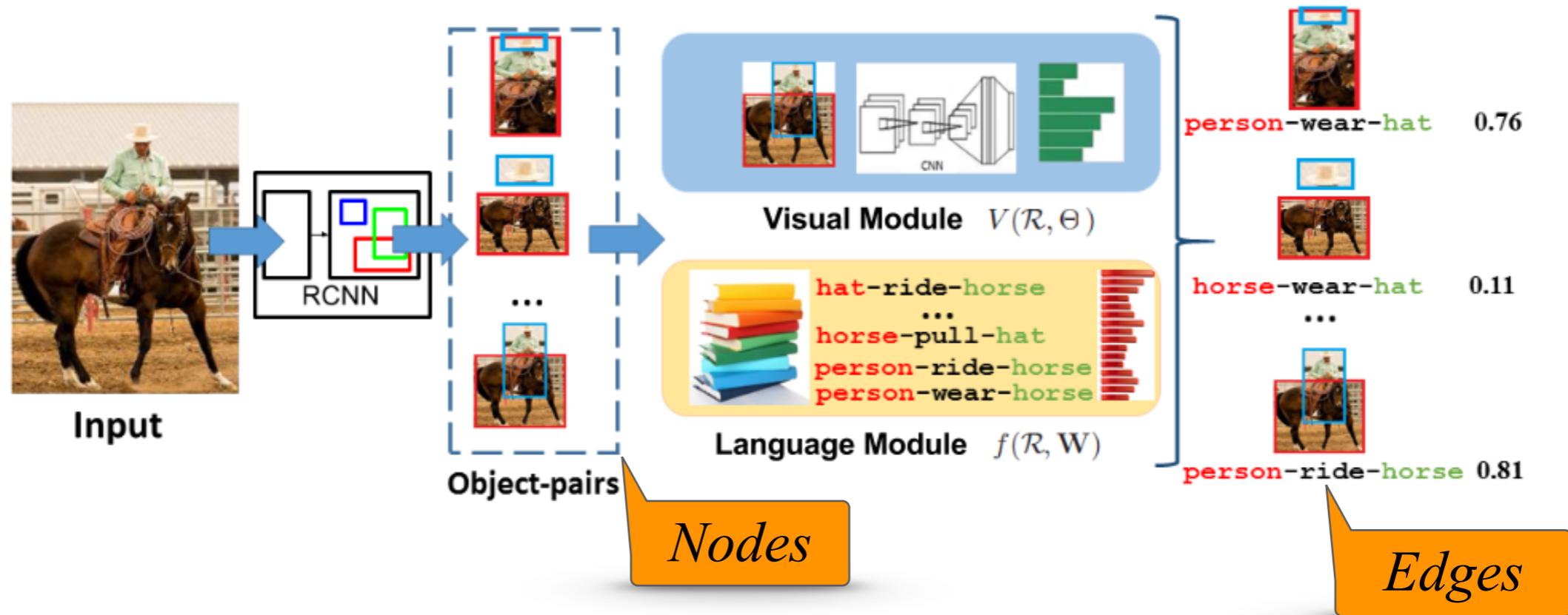
Emory University



Graph is Pervasive

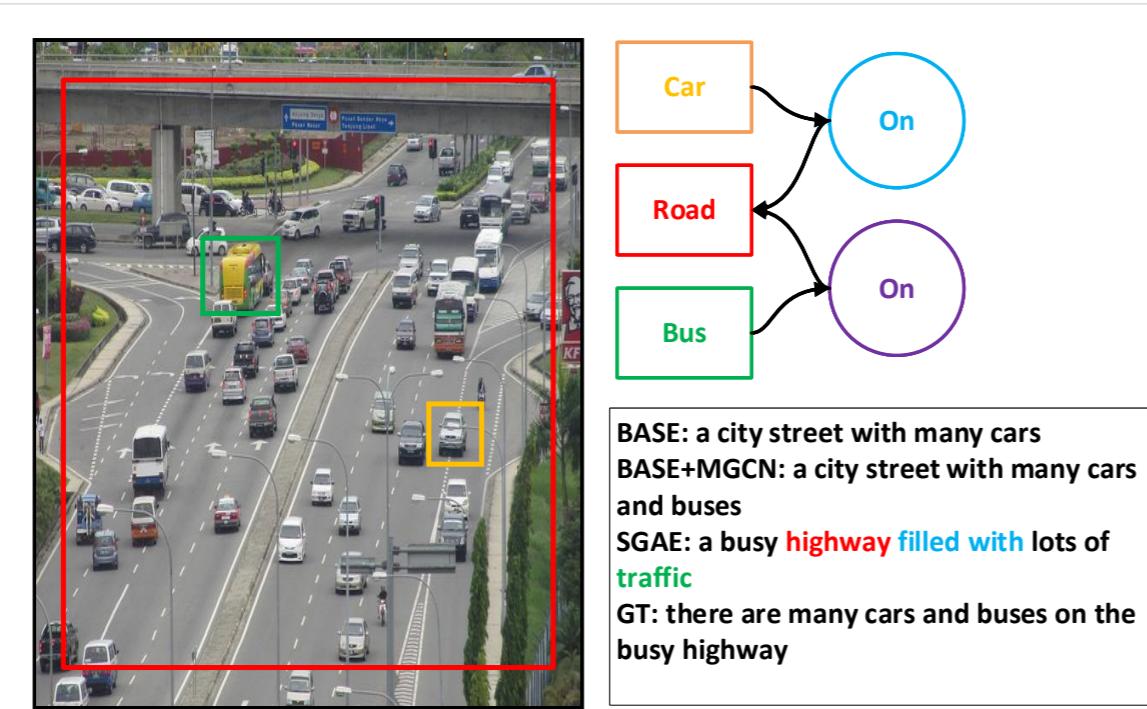


Graphs in CV



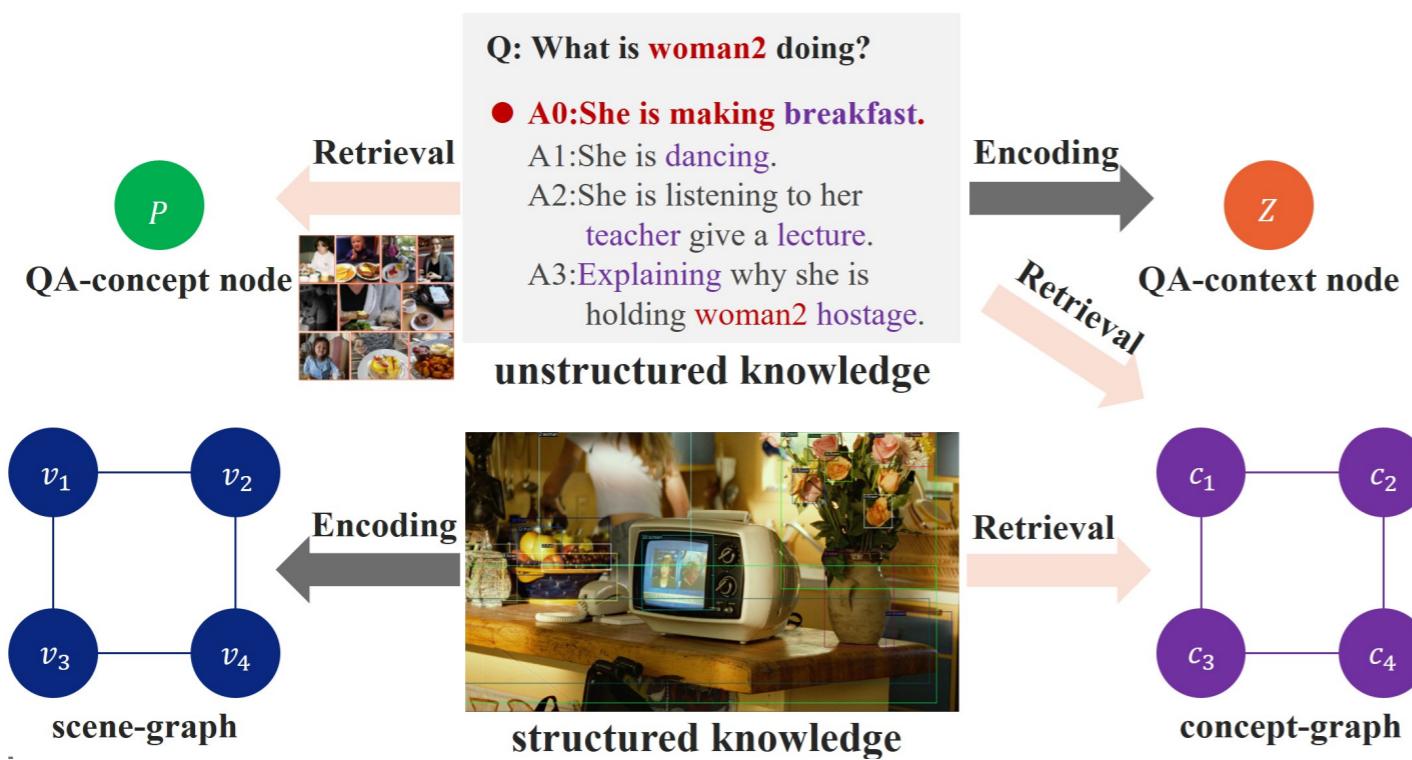
Graphs in Image Data:
Scene Graph

Graphs in CV



Image/Video Captioning

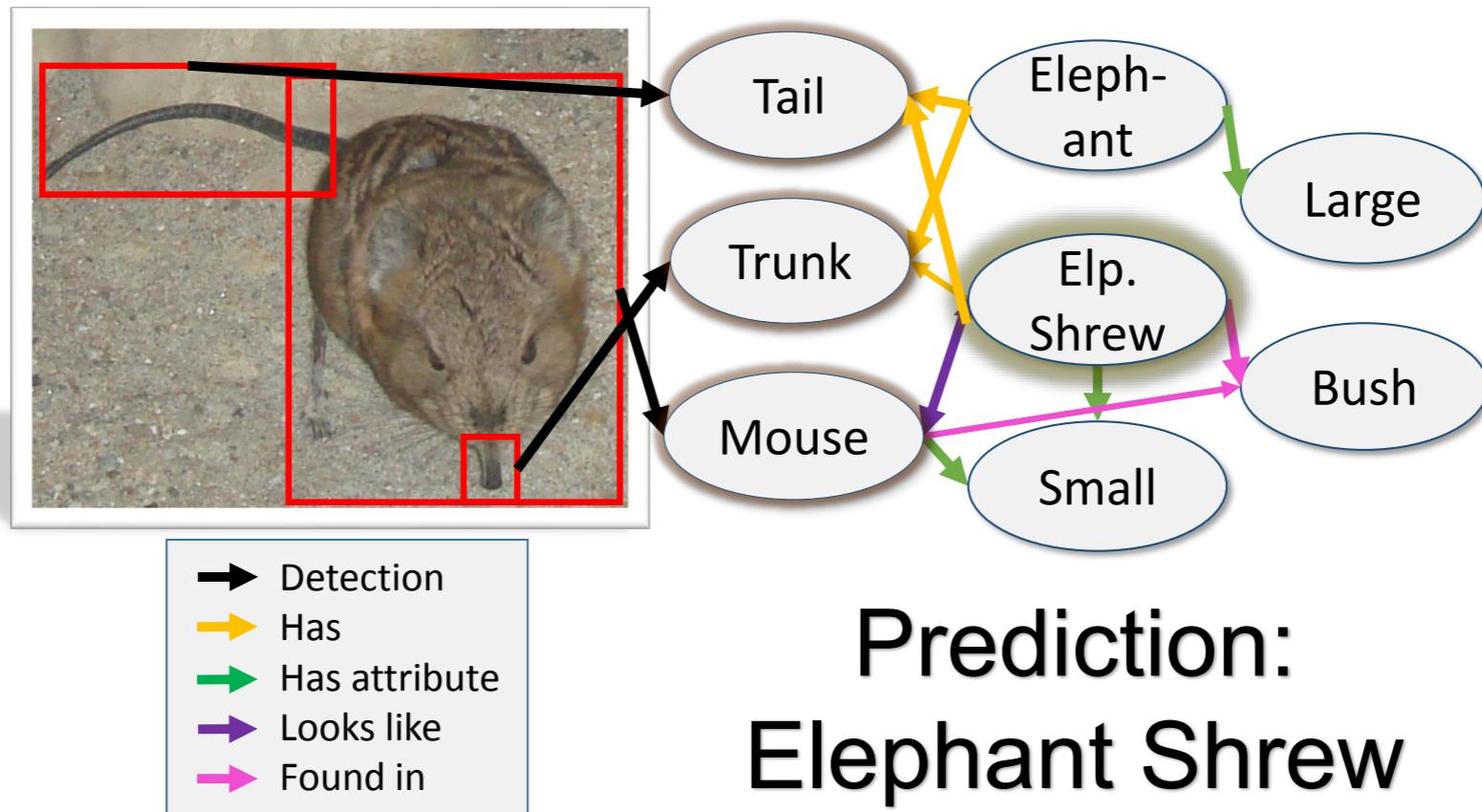
Yang, Xu, et al. "Auto-encoding scene graphs for image captioning." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.



Structured reasoning for visual Q&A

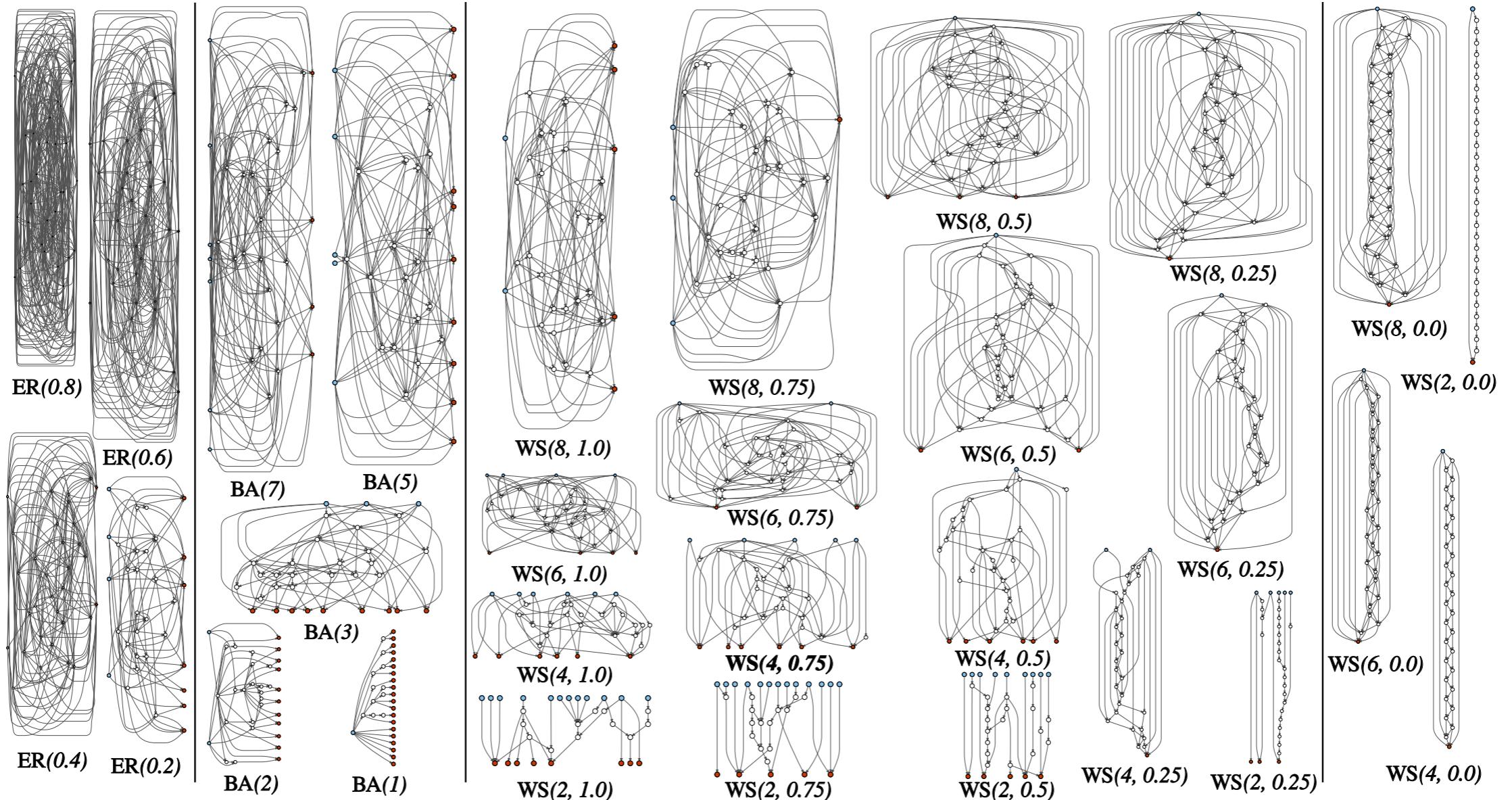
Wang, Yanan, et al. "Vqa-gnn: Reasoning with multimodal knowledge via graph neural networks for visual question answering." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

Graphs in CV



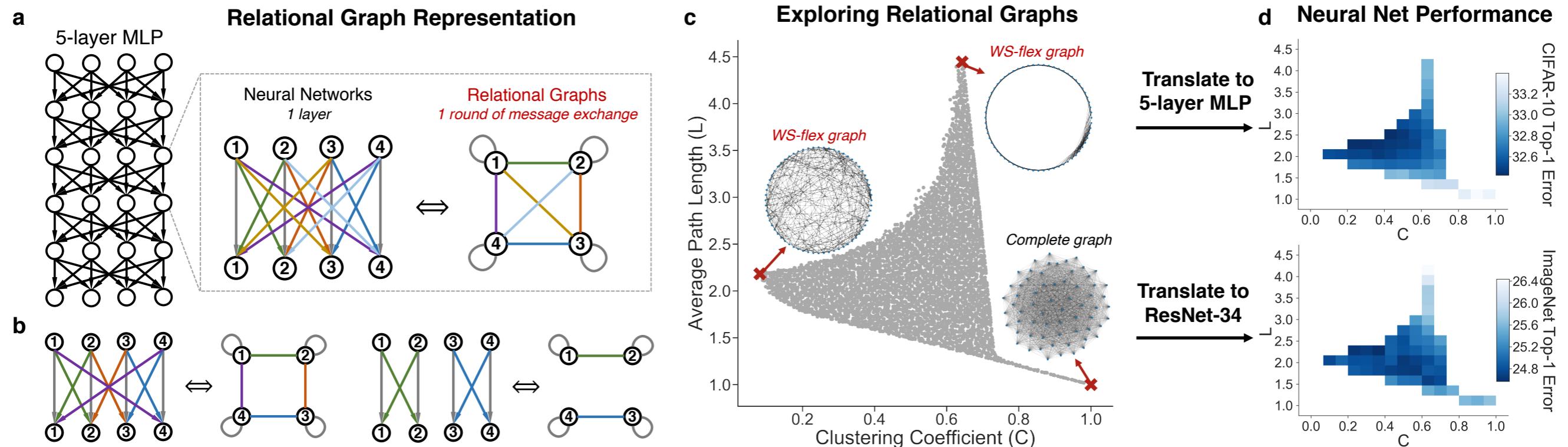
Graph as Auxiliary:
Knowledge graph for image classification

Graphs in CV



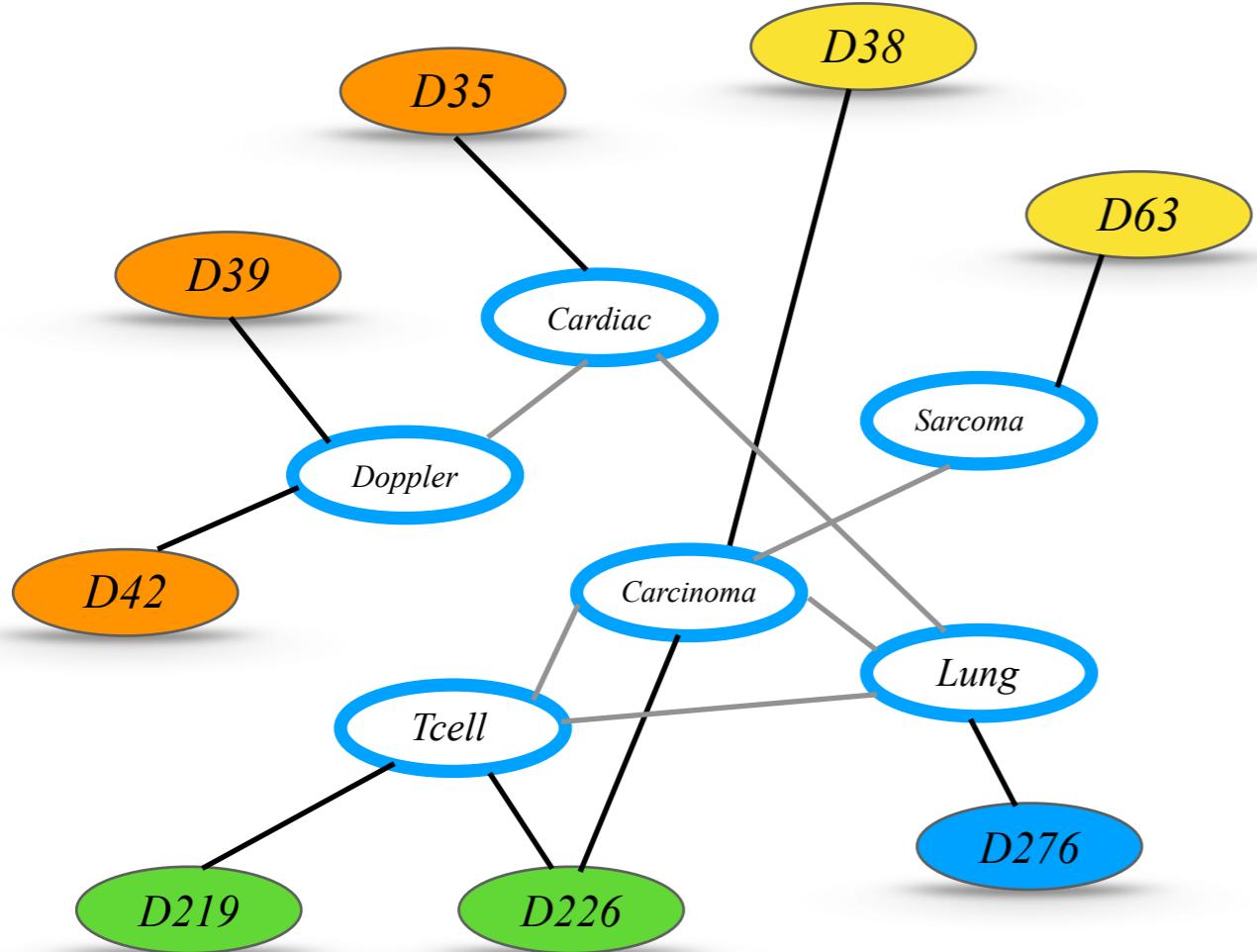
Graphs in neural network architecture

Graphs in Neural Networks Architecture



Graph topology and neural network architectures

Graphs in NLP



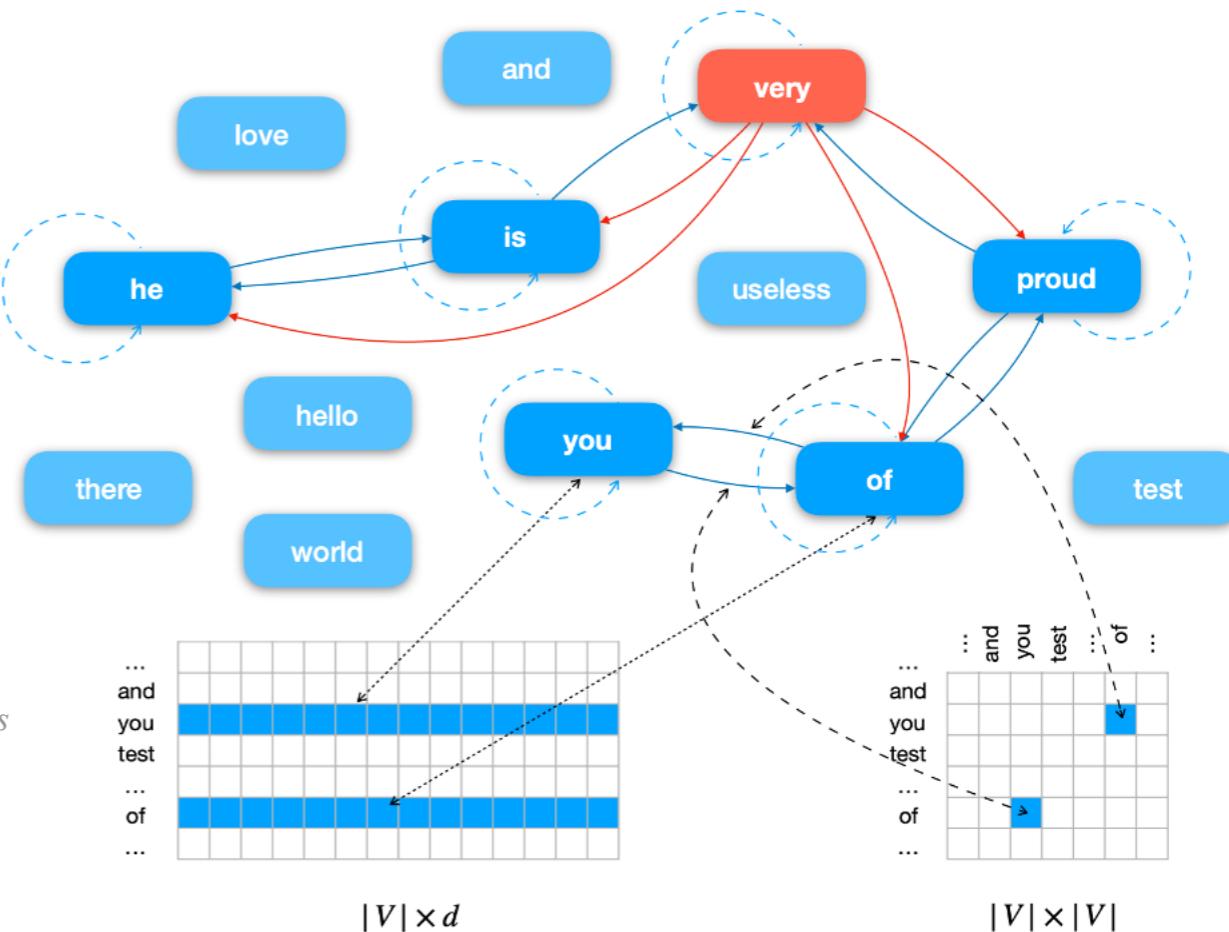
Inductive setting:
Document Graph

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.

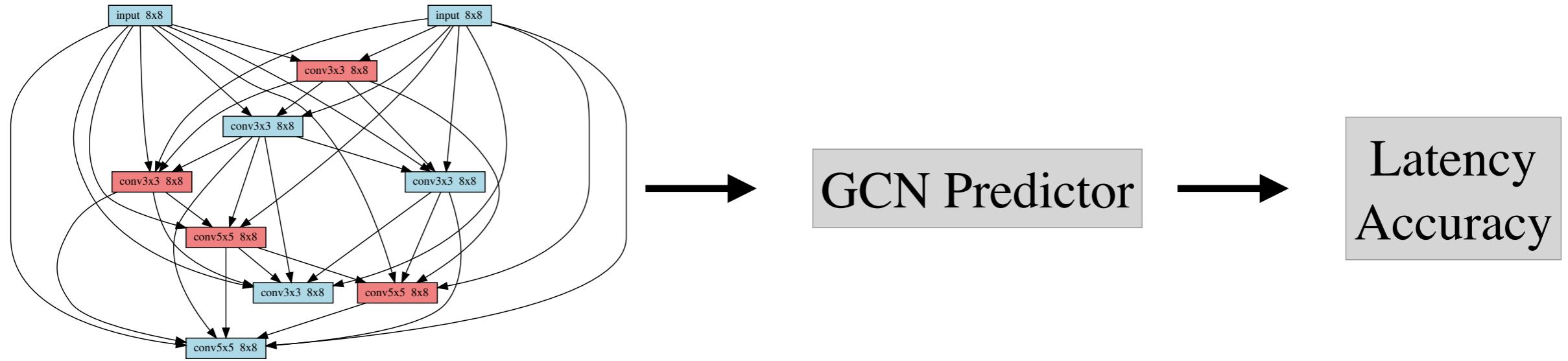
Transduction setting:
Word-Document Graph

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

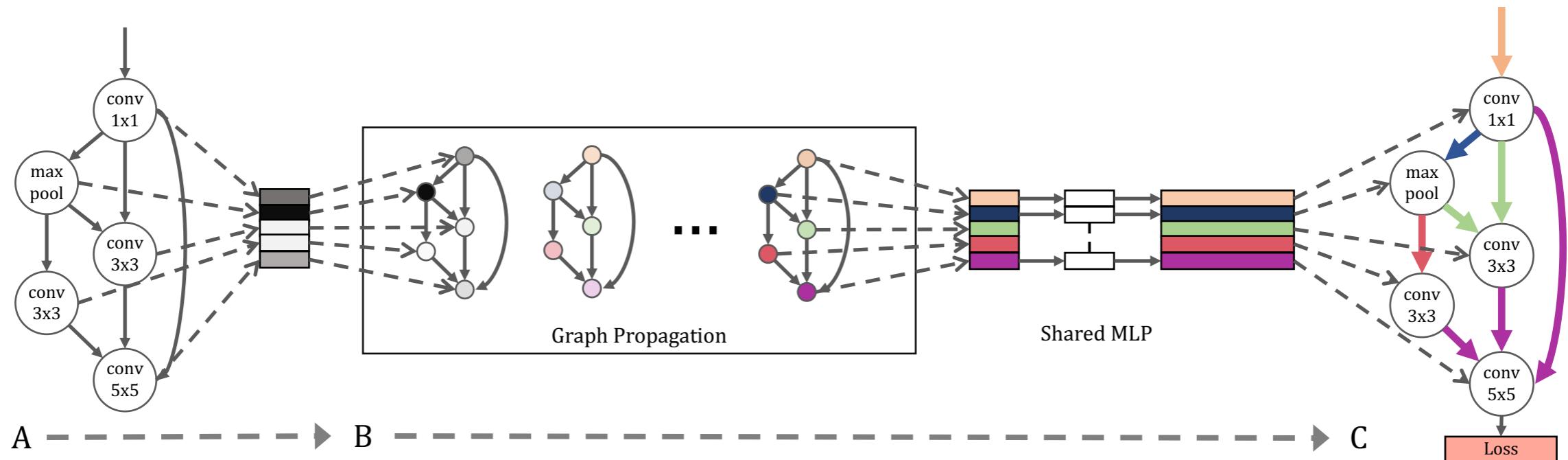
Yao, Liang, Chengsheng Mao, and Yuan Luo. "Graph convolutional networks for text classification." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.



Graphs in Neural Networks Architectures

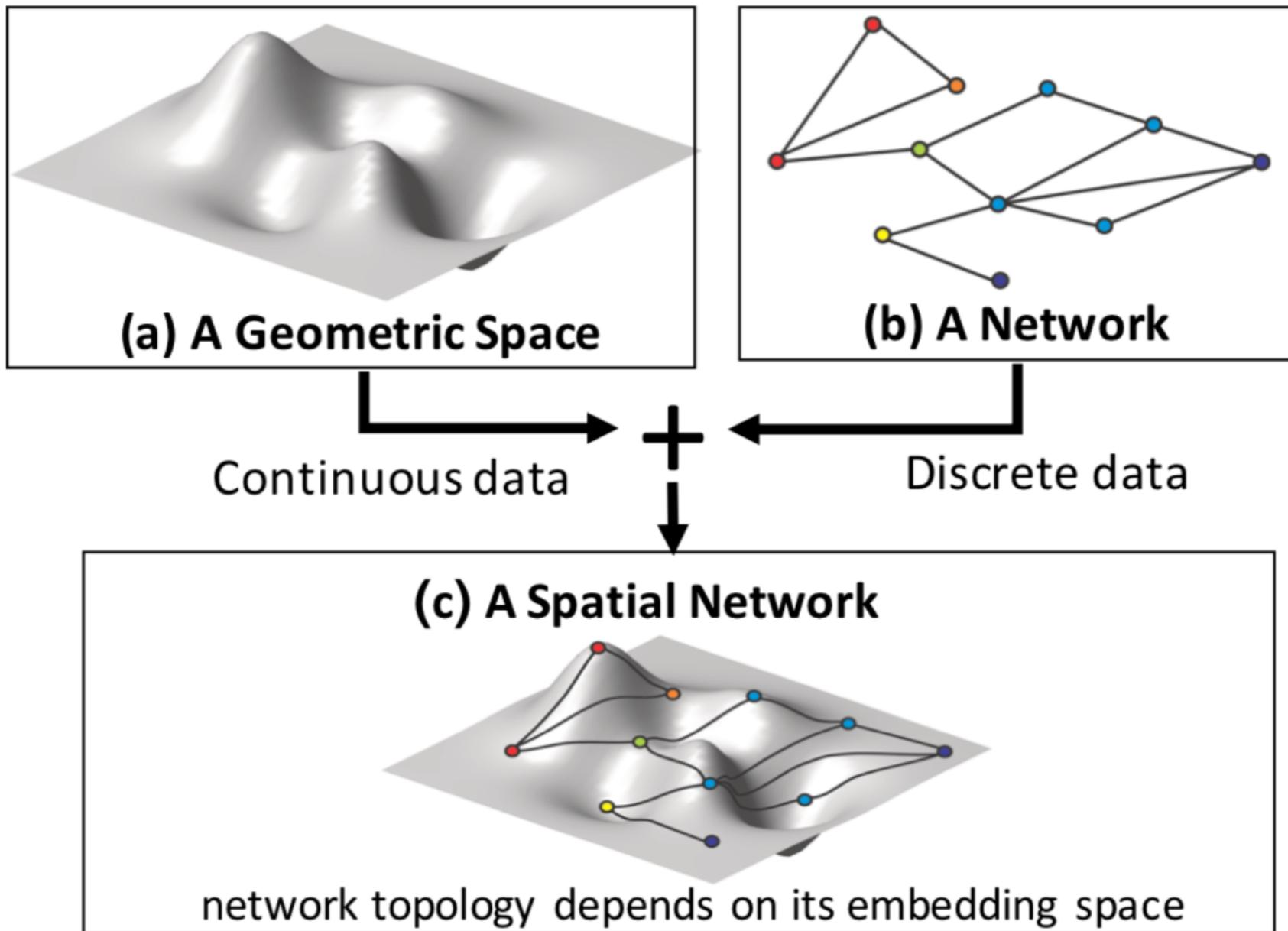


Dudziak, Lukasz, et al. "Brp-nas: Prediction-based nas using gcns." *Advances in Neural Information Processing Systems 33* (2020): 10480-10490.



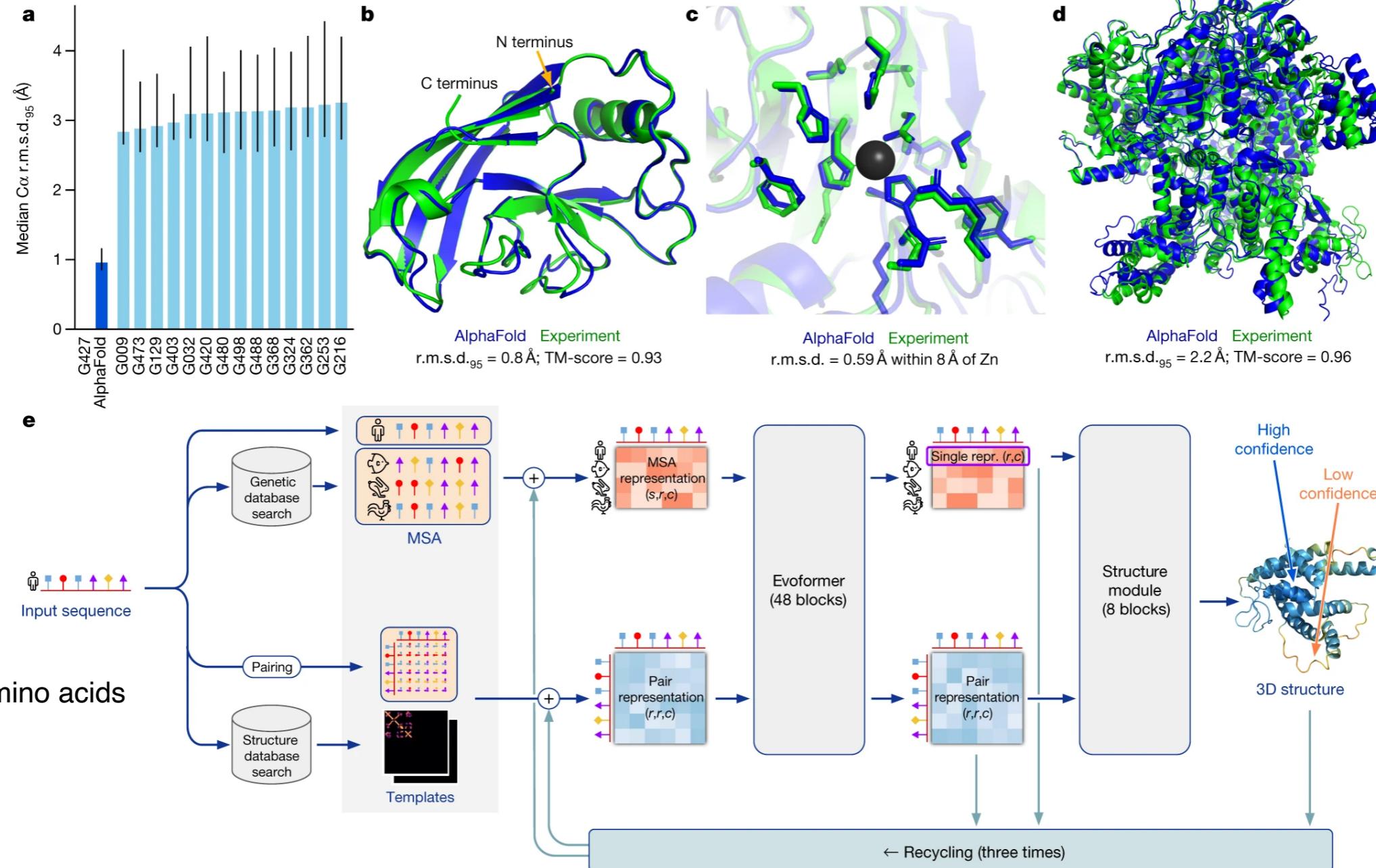
Zhang, C., et al. *Graph hypernetworks for neural architecture search*. ICLR 2019

Graphs in Spatial Data



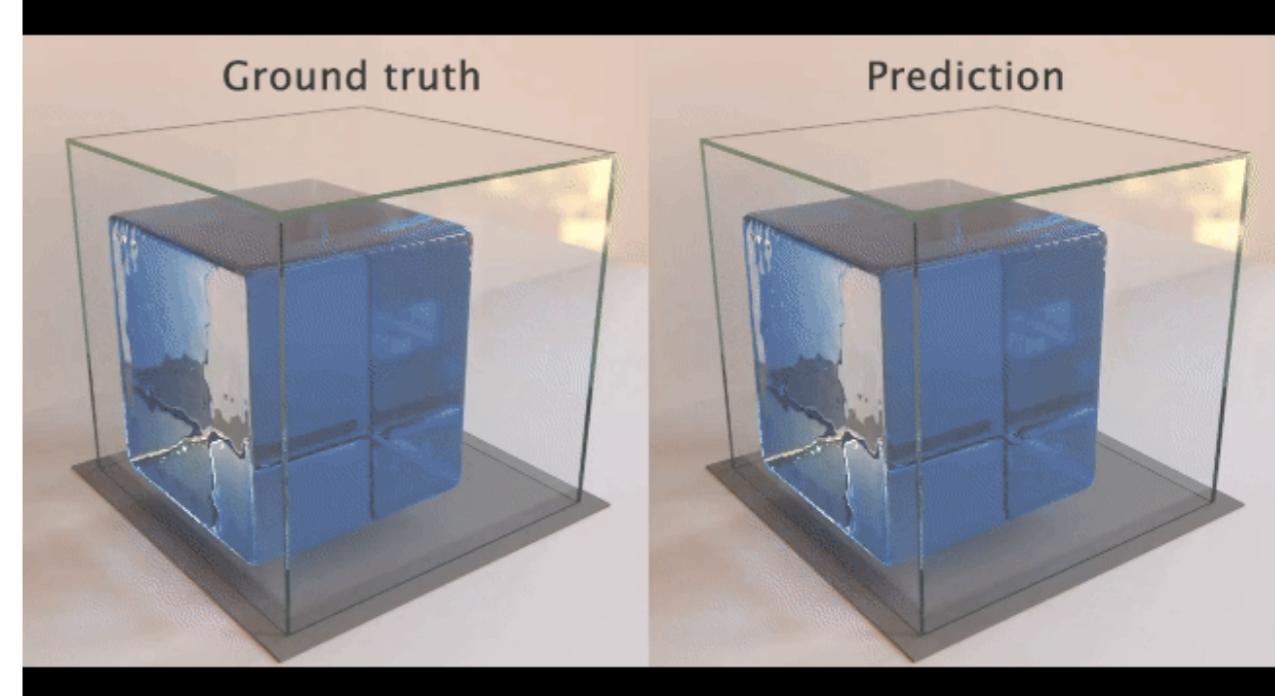
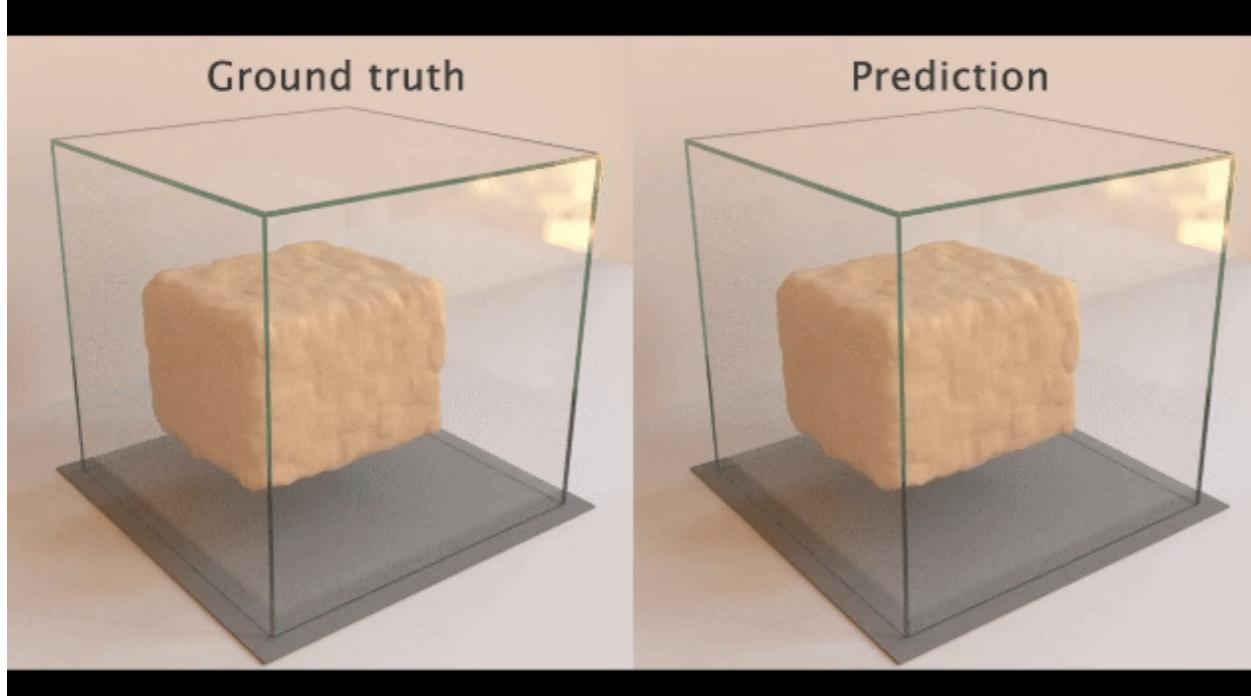
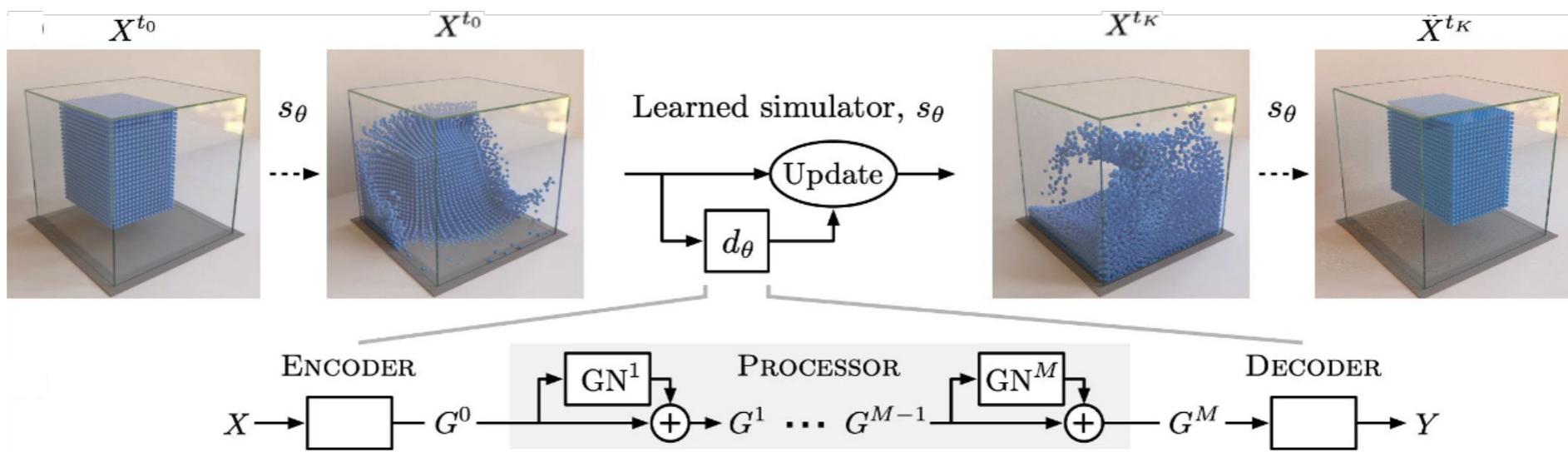
Zhang, Zheng, and Liang Zhao. "Representation learning on spatial networks." *Advances in Neural Information Processing Systems 34* (2021): 2303-2318.

Graphs in Protein Folding



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

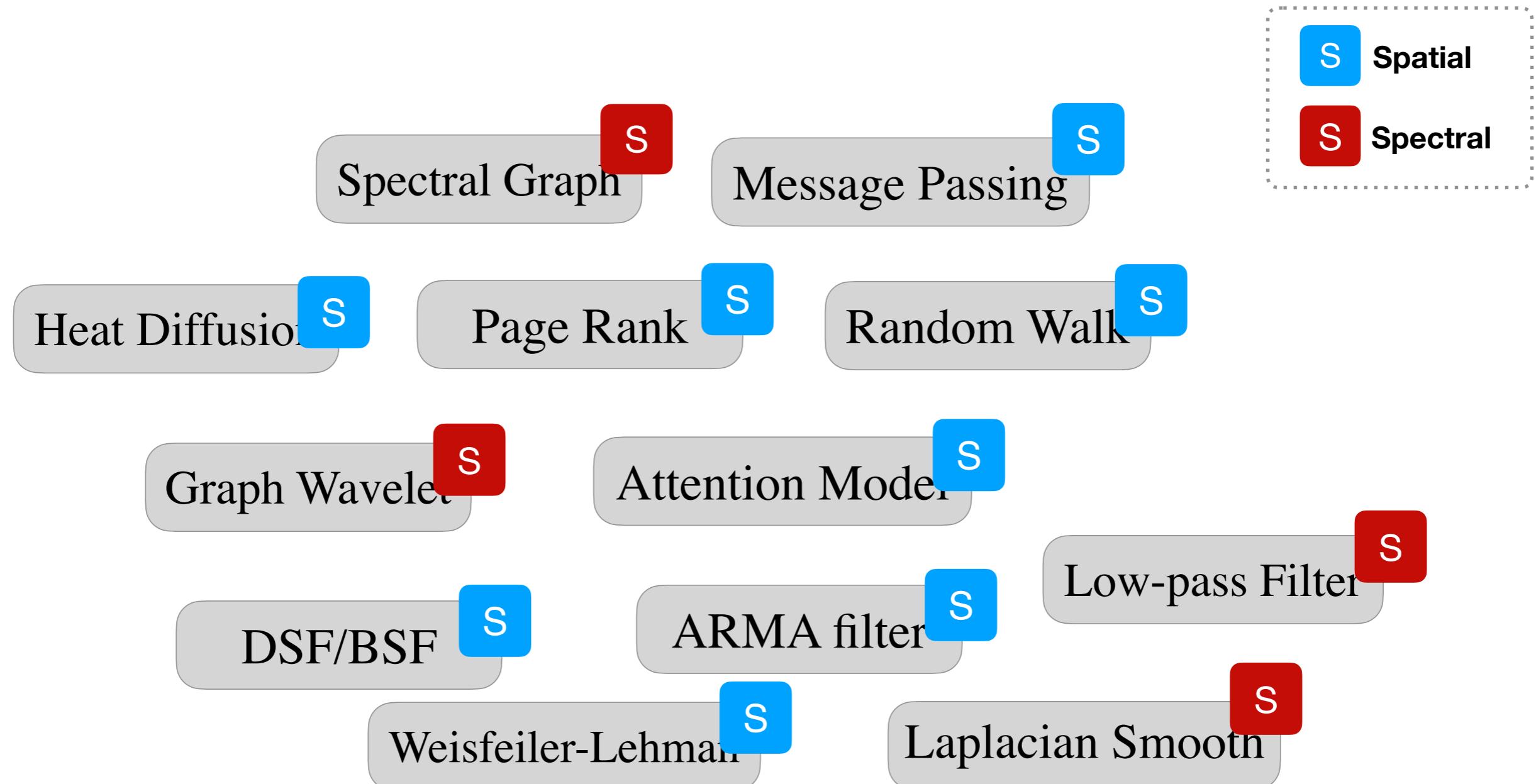
Graphs in Physics



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

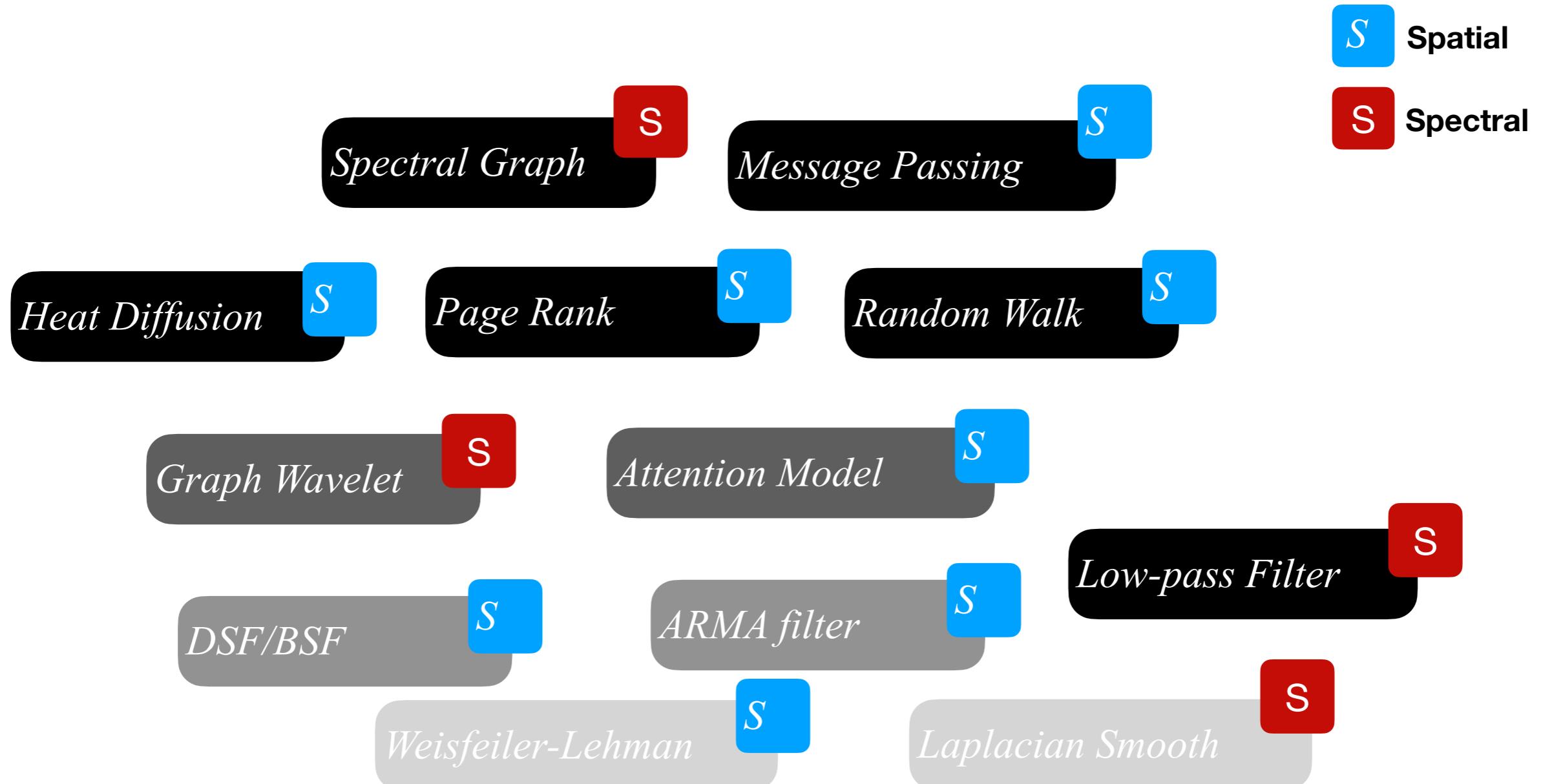
Motivation: A Unified View

A large number of graph neural networks, with different mechanisms



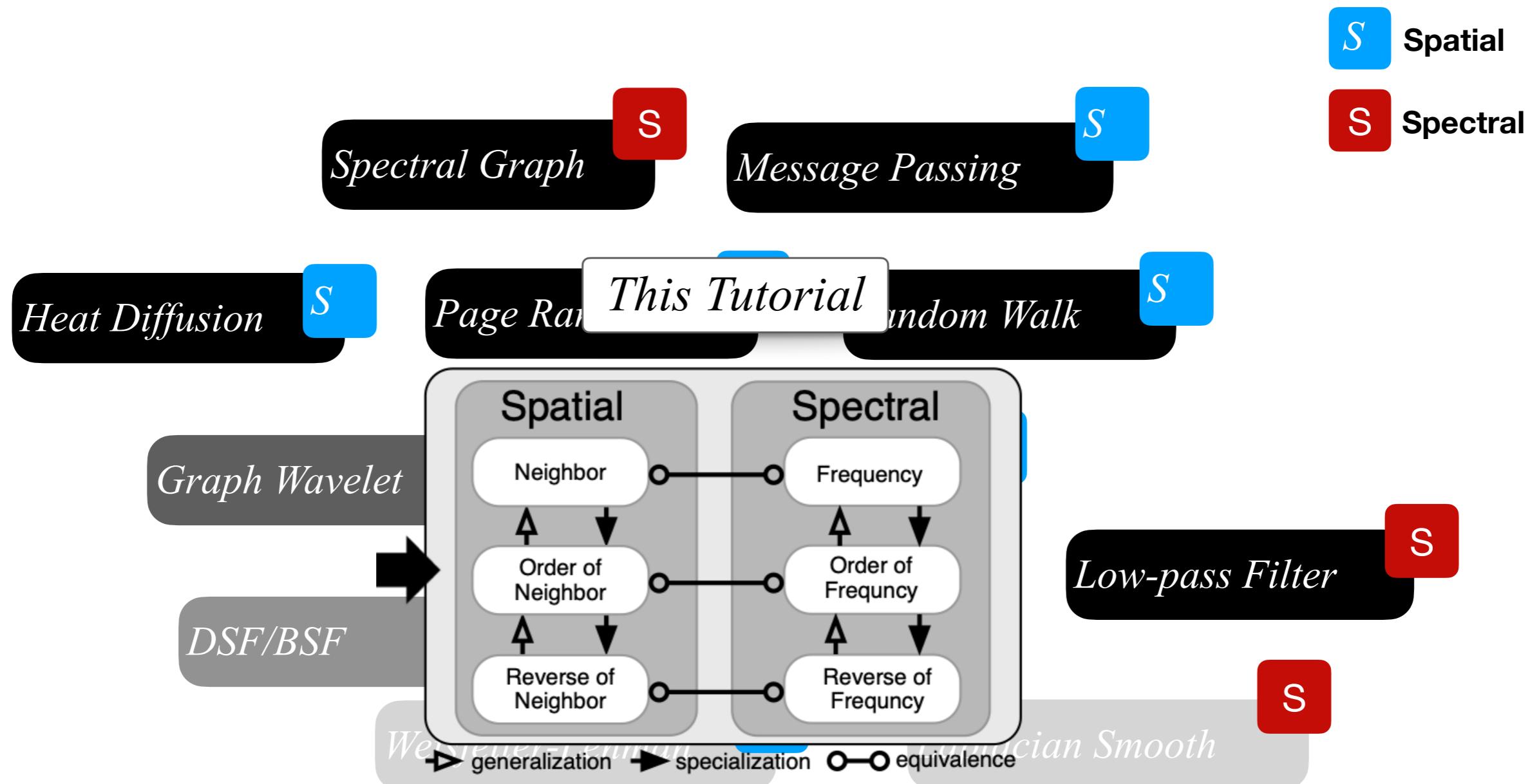
Motivation: A Unified View

A large number of graph neural networks, with different mechanisms



Motivation: A Unified View

A large number of graph neural networks, with different mechanisms



Agenda



● First Half (40 min)

- *Preliminary*: graph convolutions (25 min)
- *Analysis*: expressive power (15 min)

● Second Half (50 min)

- *Introduction*: a new unified framework (45 min)
- *Future directions* (5 min)

Agenda



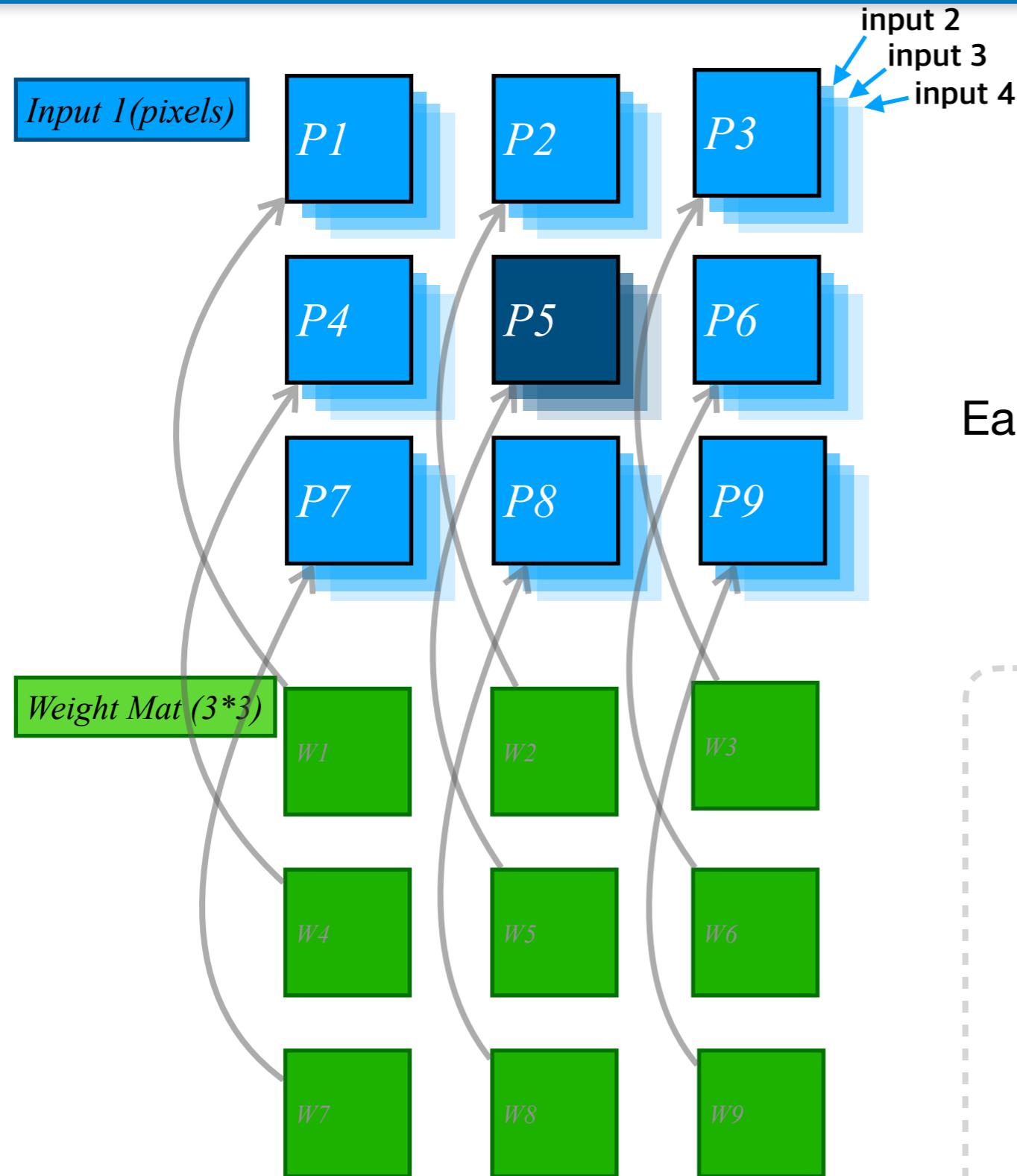
● First Half (40 min)

- *Preliminary: graph convolutions* (25 min)
- *Analysis: expressive power* (15 min)

● Second Half (50 min)

- *Introduction: a new unified framework* (45 min)
- *Future directions* (5 min)

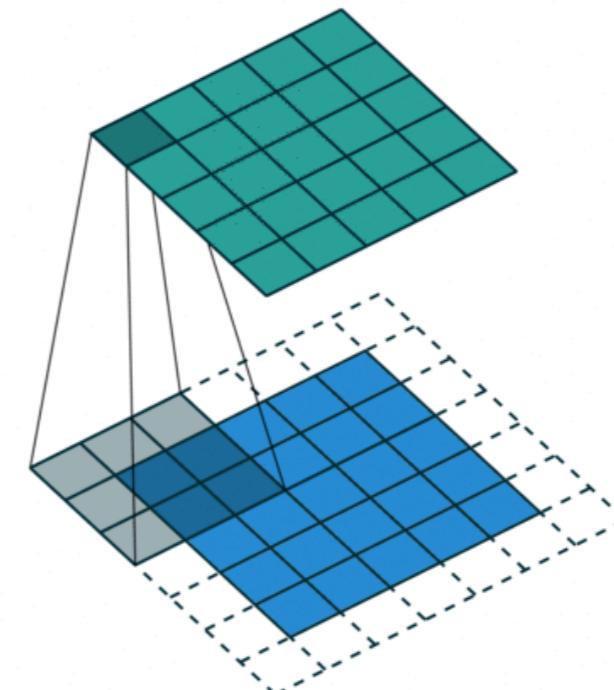
ConvNet



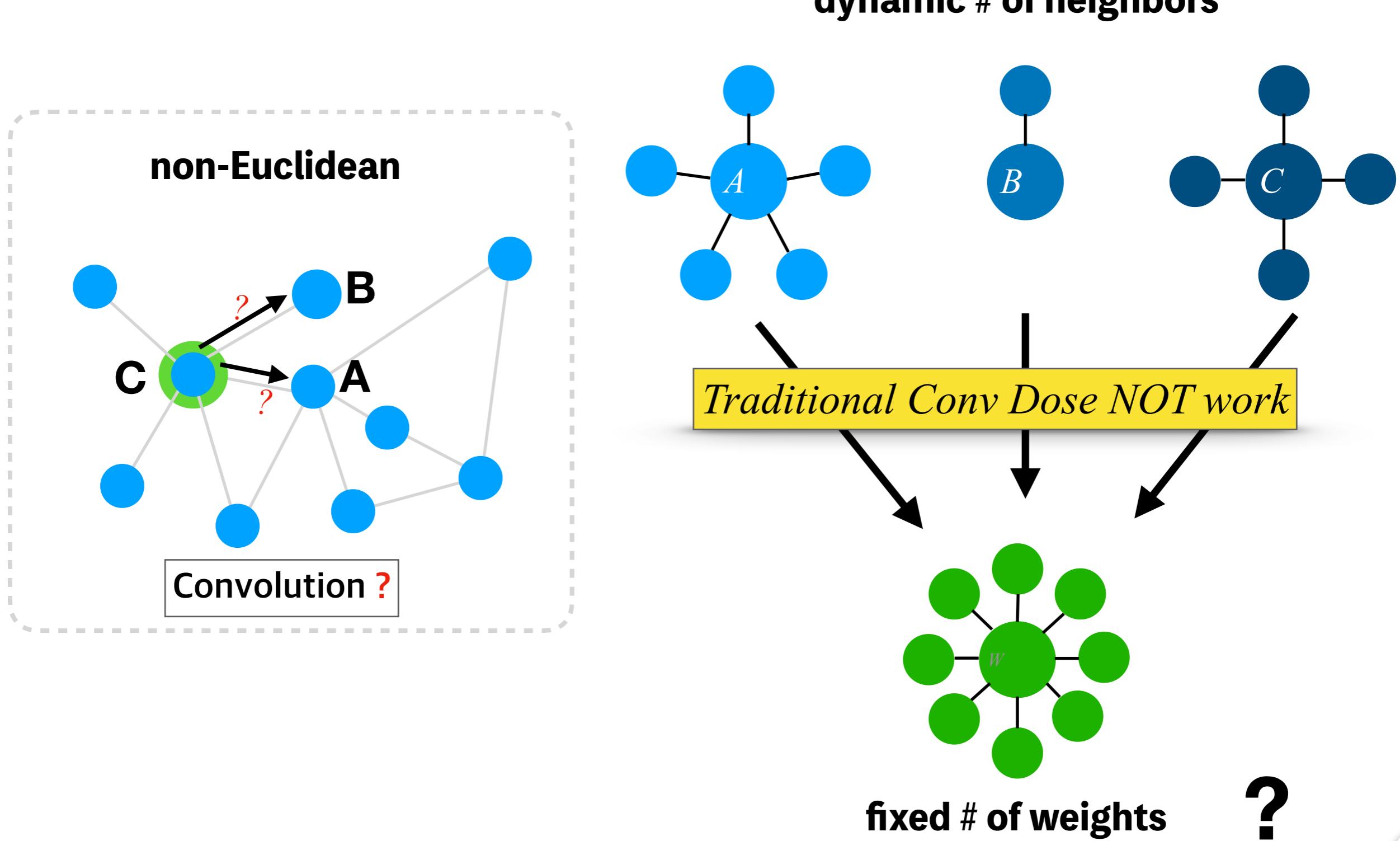
$$\hat{P}_5 = \sum_{i=1}^9 P_i \cdot W_i$$

Each pixel has **fixed** number of neighbors

Convolution



Challenge for ConvNet on Graphs



Workaround

○ Convolution Theorem

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

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

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

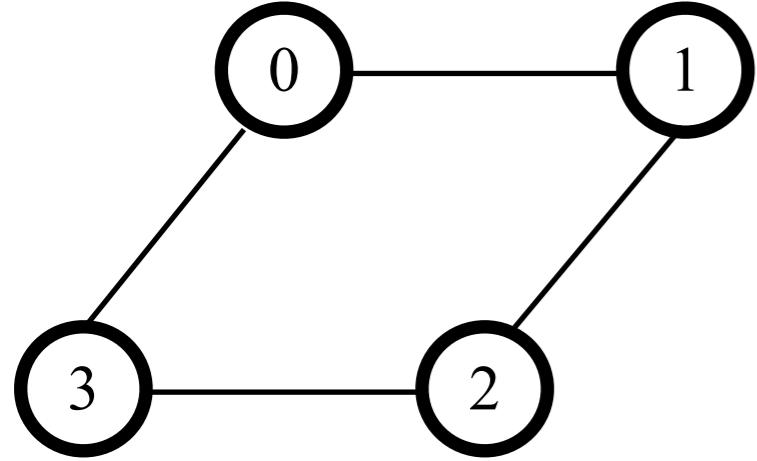
Convolution theorem: $f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$



What is the Fourier Transform on Graphs?

What is the Inverse Fourier Transform on Graphs?

Graph Fourier Transform



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

Graph Laplacian

$$L = D - A$$

Normalized Laplacian

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

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

positive semidefinite

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

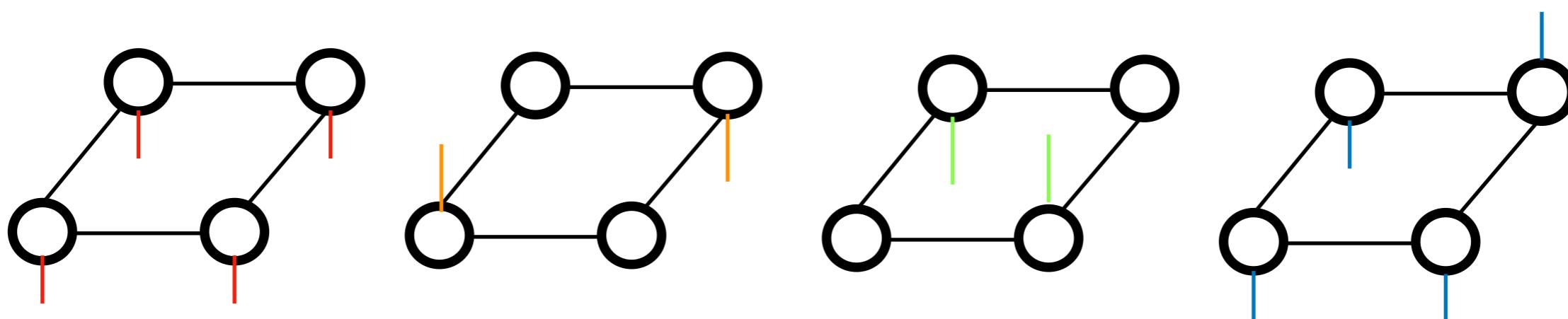
Eigenvector $U \leftrightarrow$ Frequency Component

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

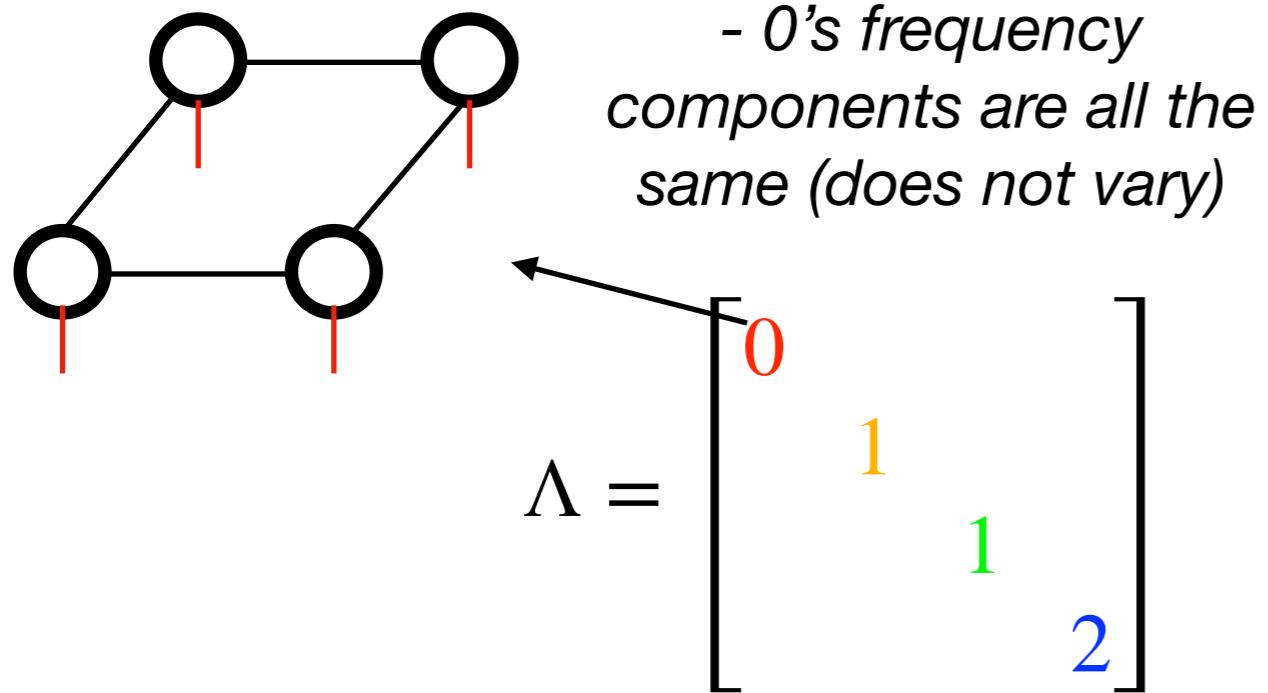
They form an **Orthonormal basis** for the vector space of graph signals

- unit vectors
- orthogonal to each other

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

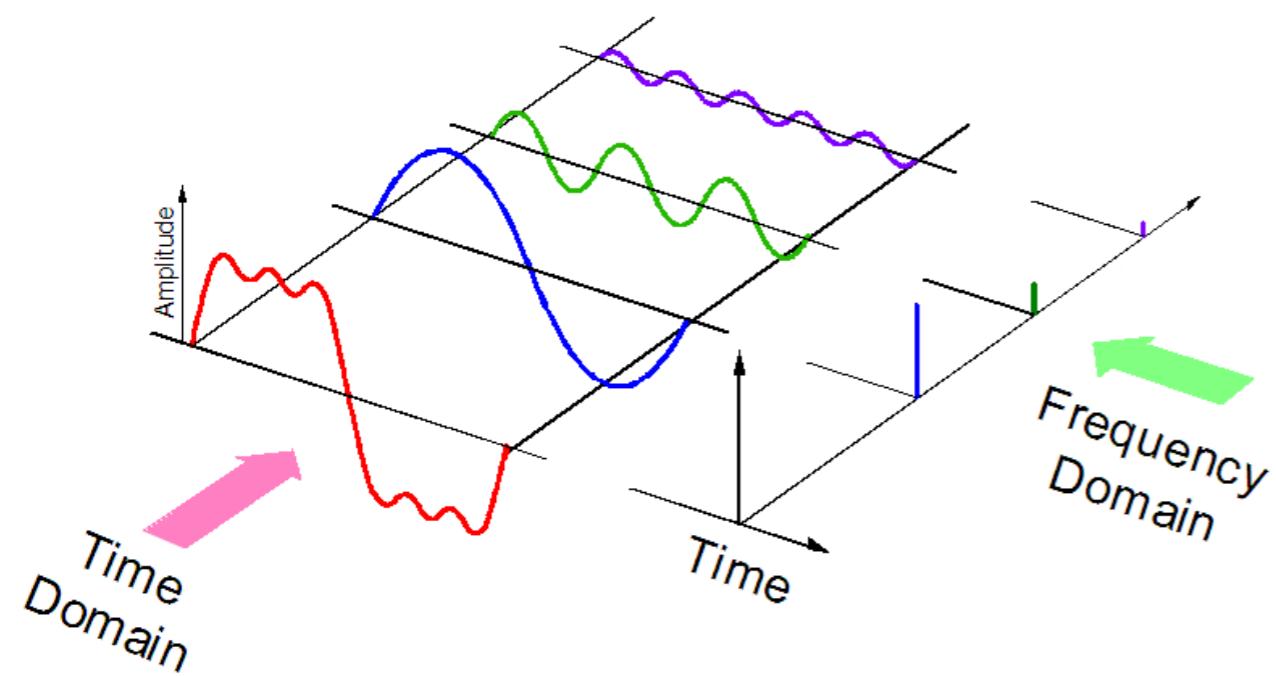


Frequencies of a Graph



Eigenvalue $\Lambda \leftrightarrow$ Frequency

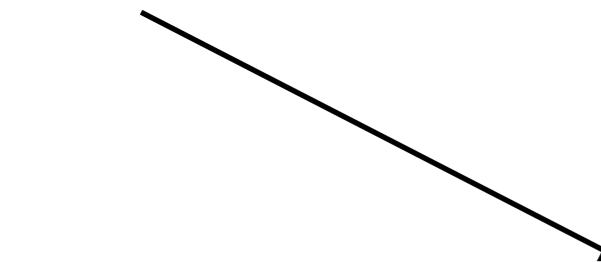
For graph,
low frequency \leftrightarrow smooth signal
(signal on nodes does not vary too much between neighboring nodes)



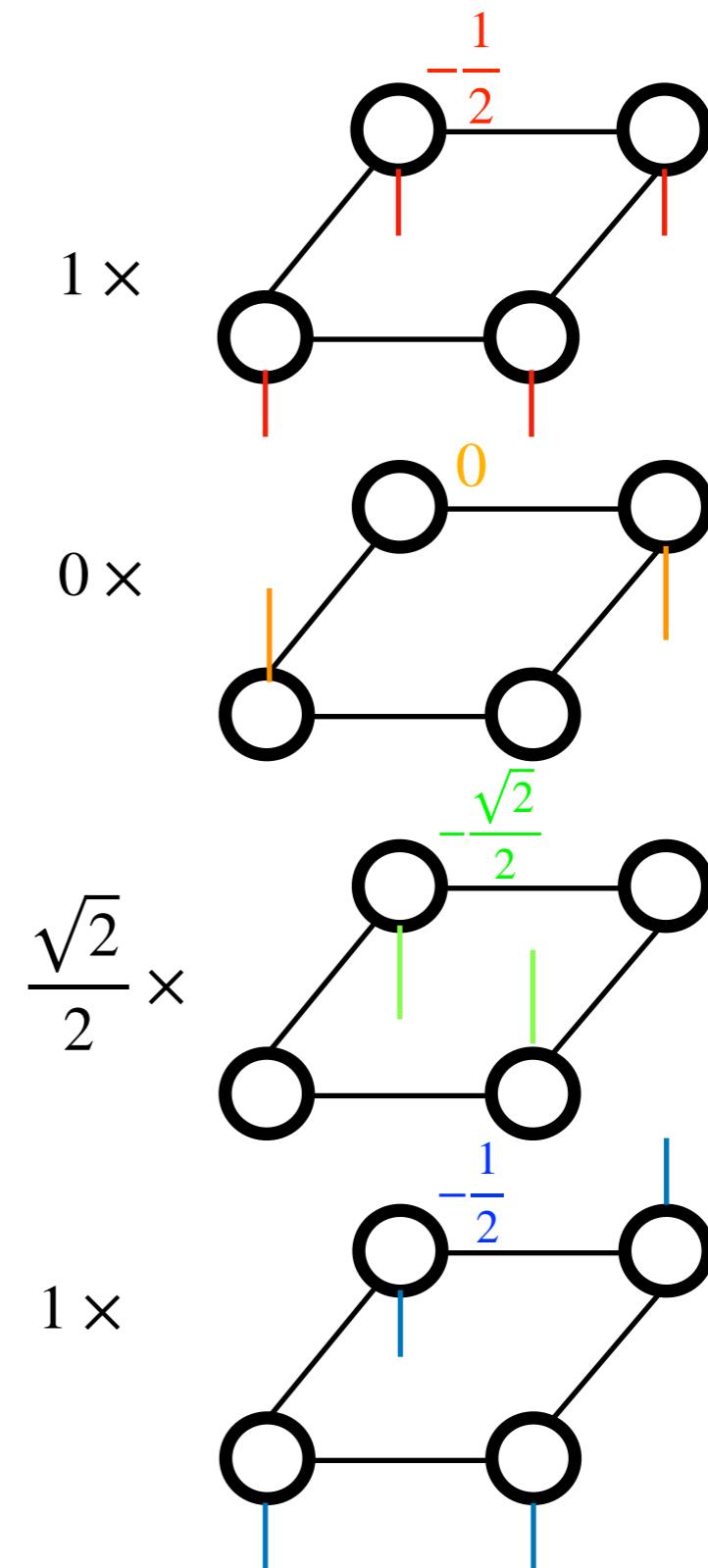
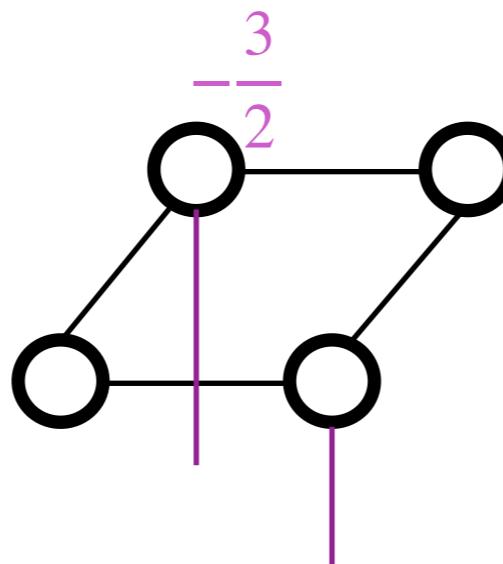
For sound waves,
low frequency \leftrightarrow slowly varying signal
Frequency \leftrightarrow hertz (Hz)

Graph Fourier Transform

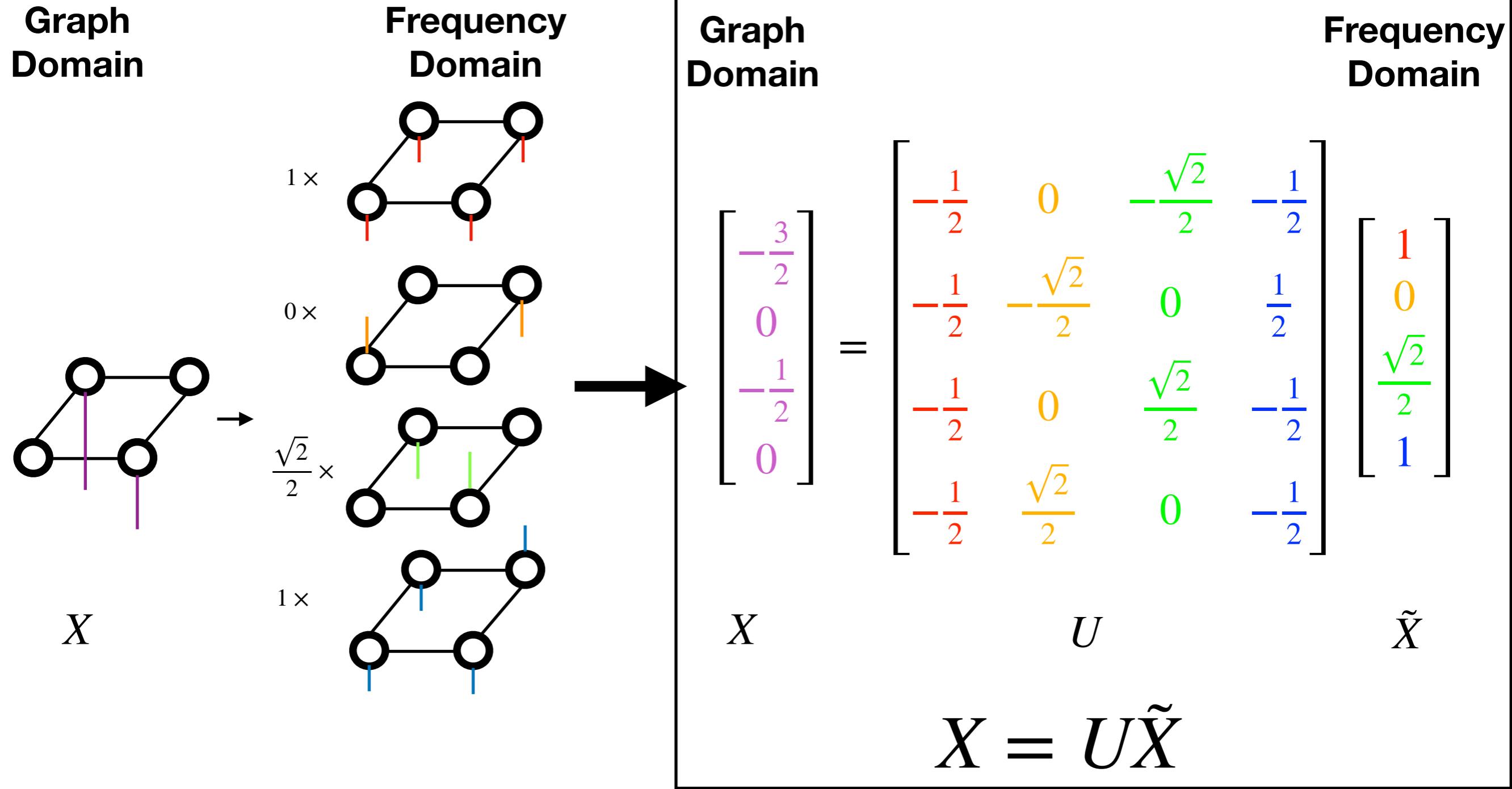
$$-\frac{3}{2} = (1 \times -\frac{1}{2}) + (0 \times 0) + (\frac{\sqrt{2}}{2} \times -\frac{\sqrt{2}}{2}) + (1 \times -\frac{1}{2})$$



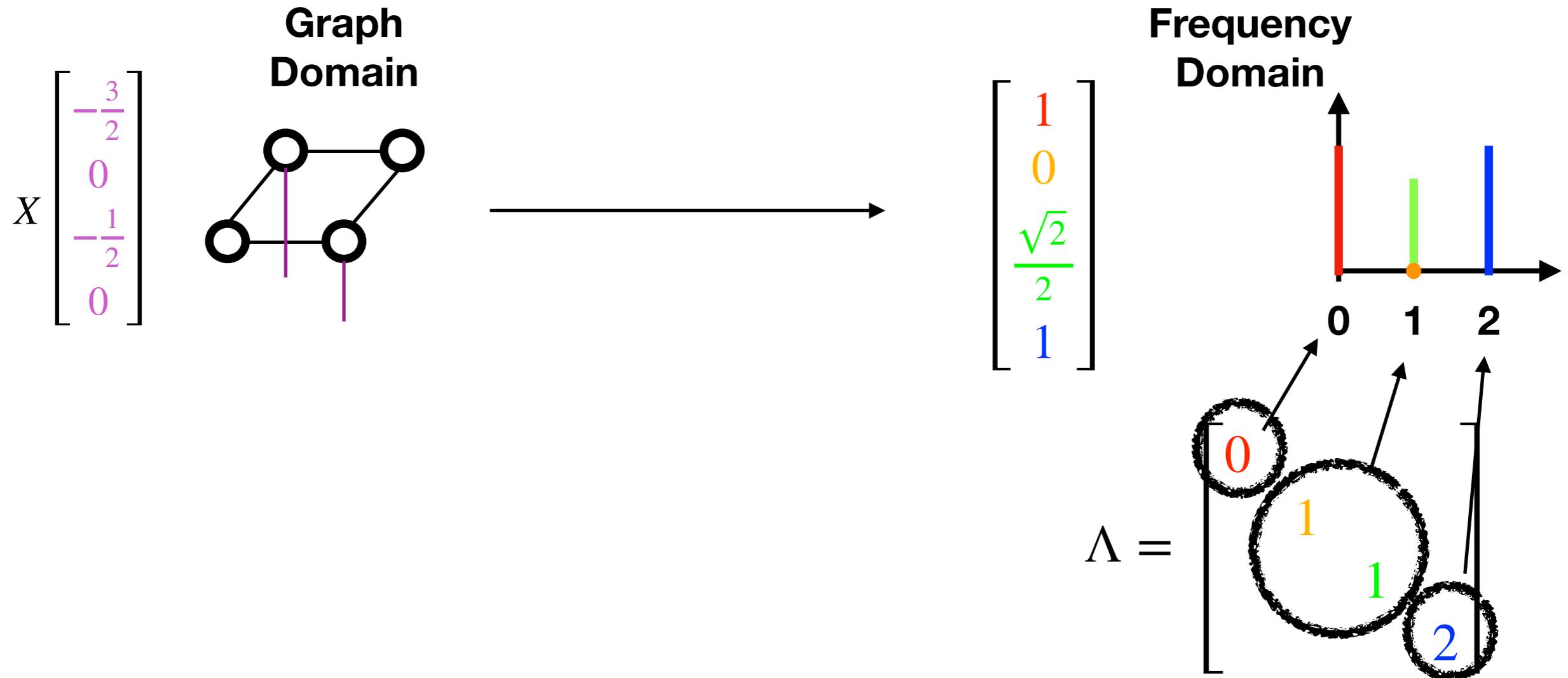
$$X = \begin{bmatrix} -\frac{3}{2} \\ 0 \\ -\frac{1}{2} \\ 0 \end{bmatrix}$$



Graph Fourier Transform



Graph Fourier Transform



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

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

Convolution on Graph Data

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

Convolution on Graph Data

$$g_\theta * x = U g_\theta(\Lambda) U^\top x \quad L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$$

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

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

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

Graph Convolutional Neural Network

Ver 2.0

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



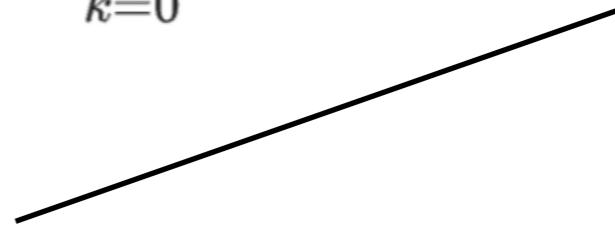
$$g_{\theta}(\Lambda) \approx \sum_{k=0}^{K-1} \theta_k \Lambda^k$$

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

Graph Convolutional Neural Network

Ver 2.0

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



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

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

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

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

ChebNet

Chebyshev polynomial

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

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

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

The “GCN”

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

When k is 2, and

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

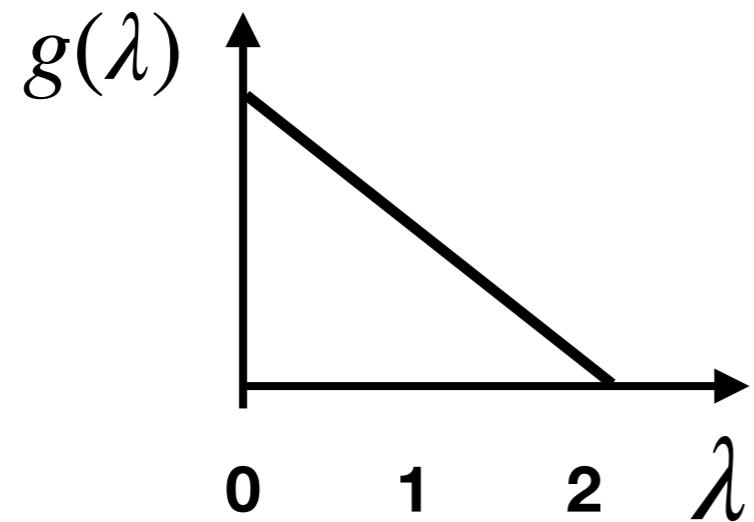
With only two free parameters. How about fewer?

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

The “GCN”

Spectral View of GCN

$$g_\theta * f = \textcolor{red}{U g_\theta U^\top} f = \textcolor{red}{g_\theta(L)} \cdot f$$
$$g_\lambda = g(\lambda) = 2 - \lambda$$



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

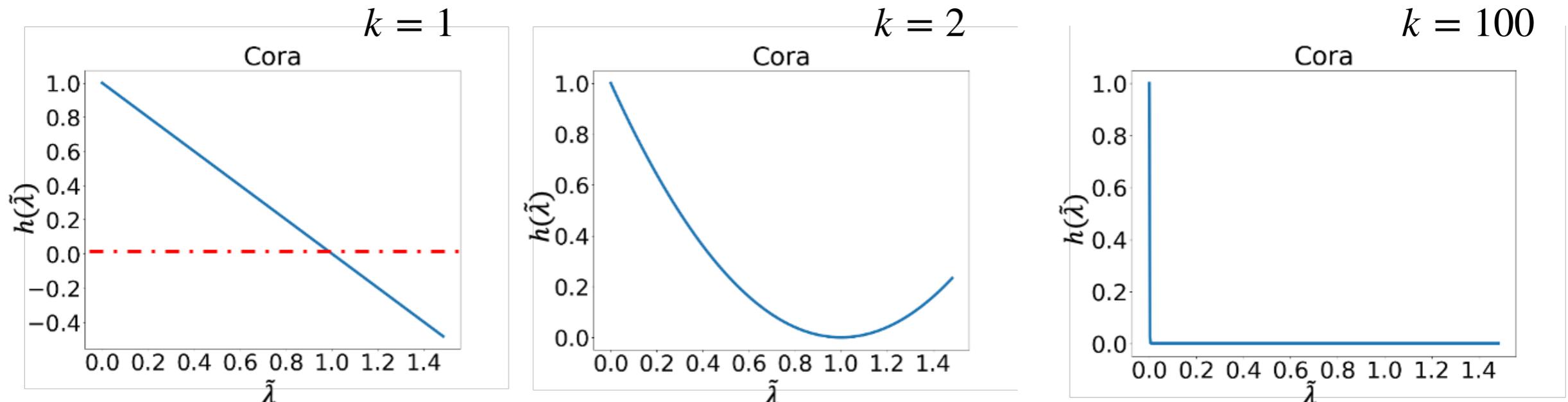
$$0 < \lambda_0 \leq \dots \leq \lambda_{n-1} \leq 2$$

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.

Spatial View of GCN

If constraint g_θ is polynomial

$$g_\theta * f = \mathbf{U} g_\theta \mathbf{U}^\top f \xrightarrow{\text{curve}} = g_\theta(L) \cdot f$$

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

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

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

1-order polynomial without weights

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

A and L are normalized

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

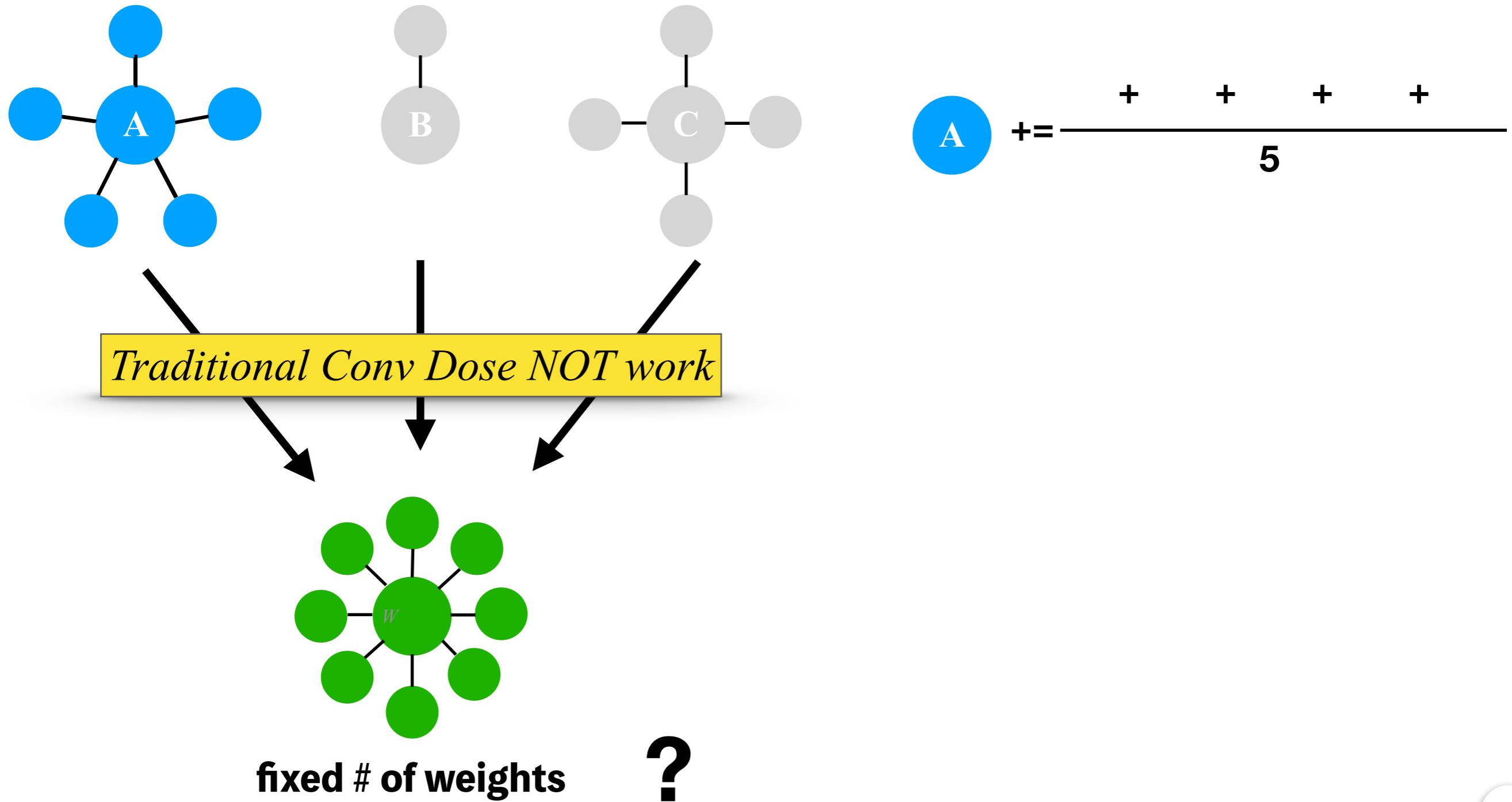
$$= \boxed{f} + \boxed{A \cdot f}$$

Self Average of neighbors

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

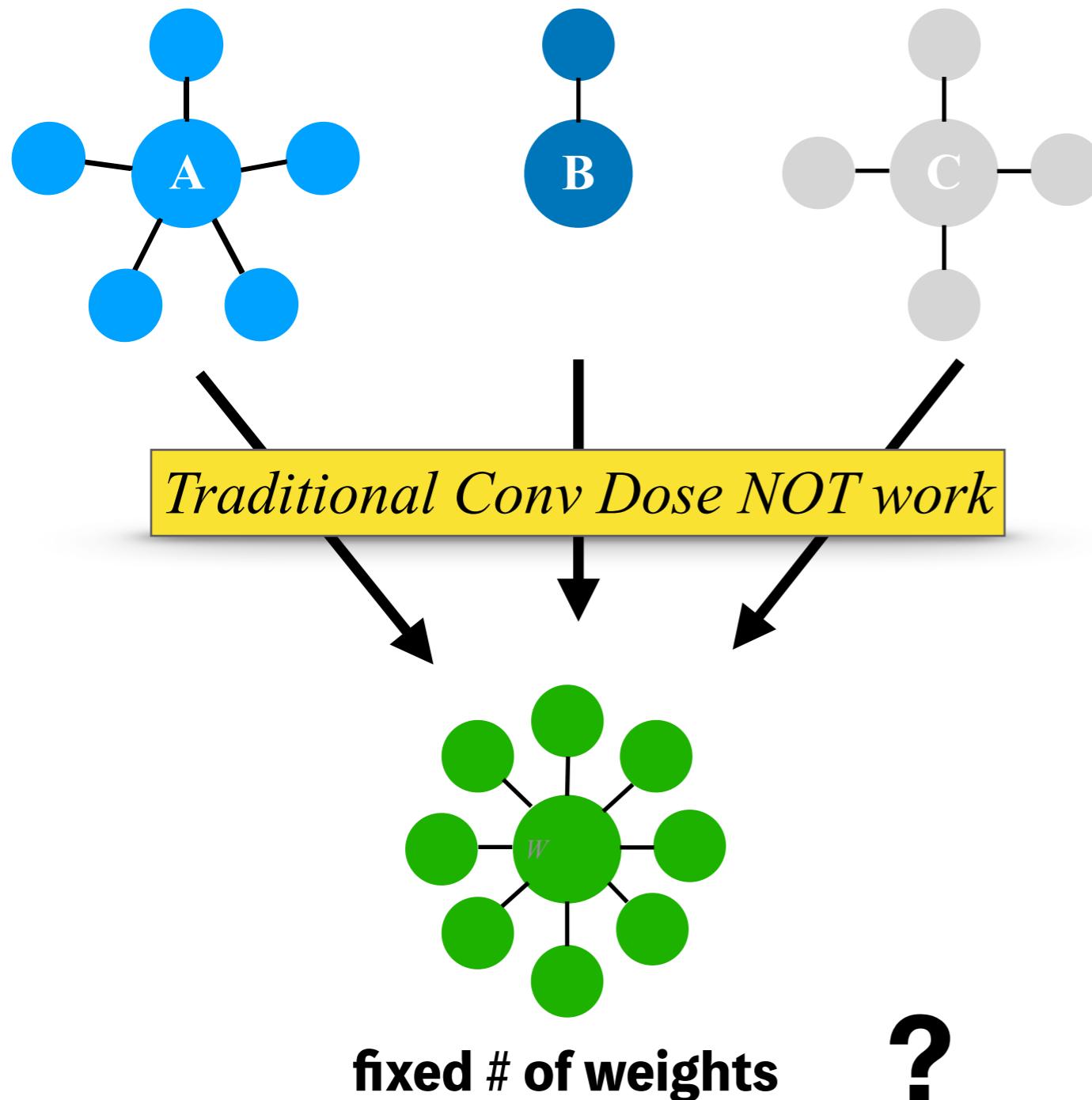
Convolution on Graph Data

dynamic # of neighbors



Convolution on Graph Data

dynamic # of neighbors



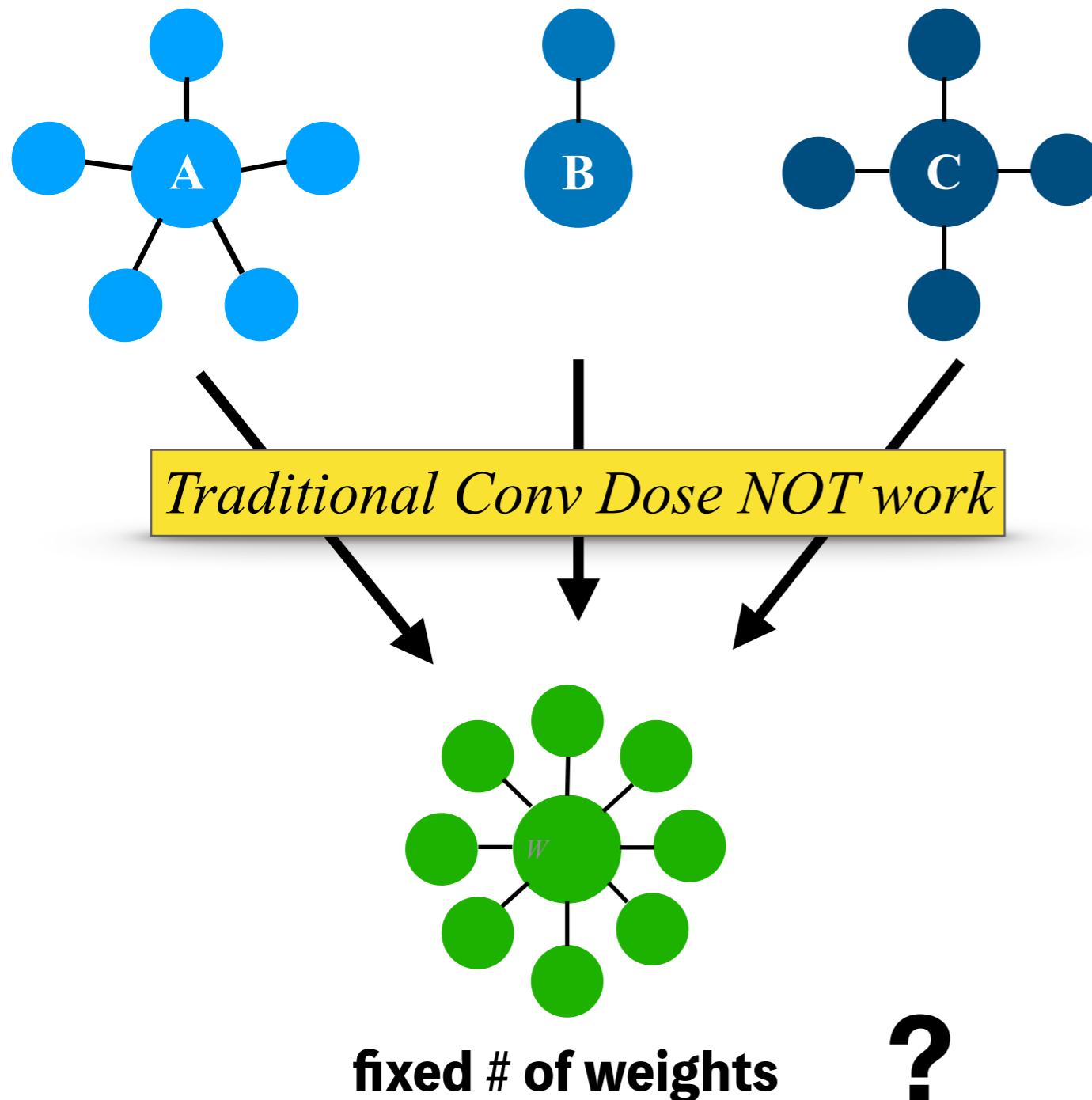
$$\begin{array}{rcl} A & + = & \text{---} \\ & & \text{---} \\ B & = & \text{---} \\ & & \text{---} \end{array}$$

5

The diagram illustrates a convolution operation. Node A's value is updated by summing its own value and the values of its five neighbors. Node B's value is updated by summing its own value and the values of its one neighbor.

Convolution on Graph Data

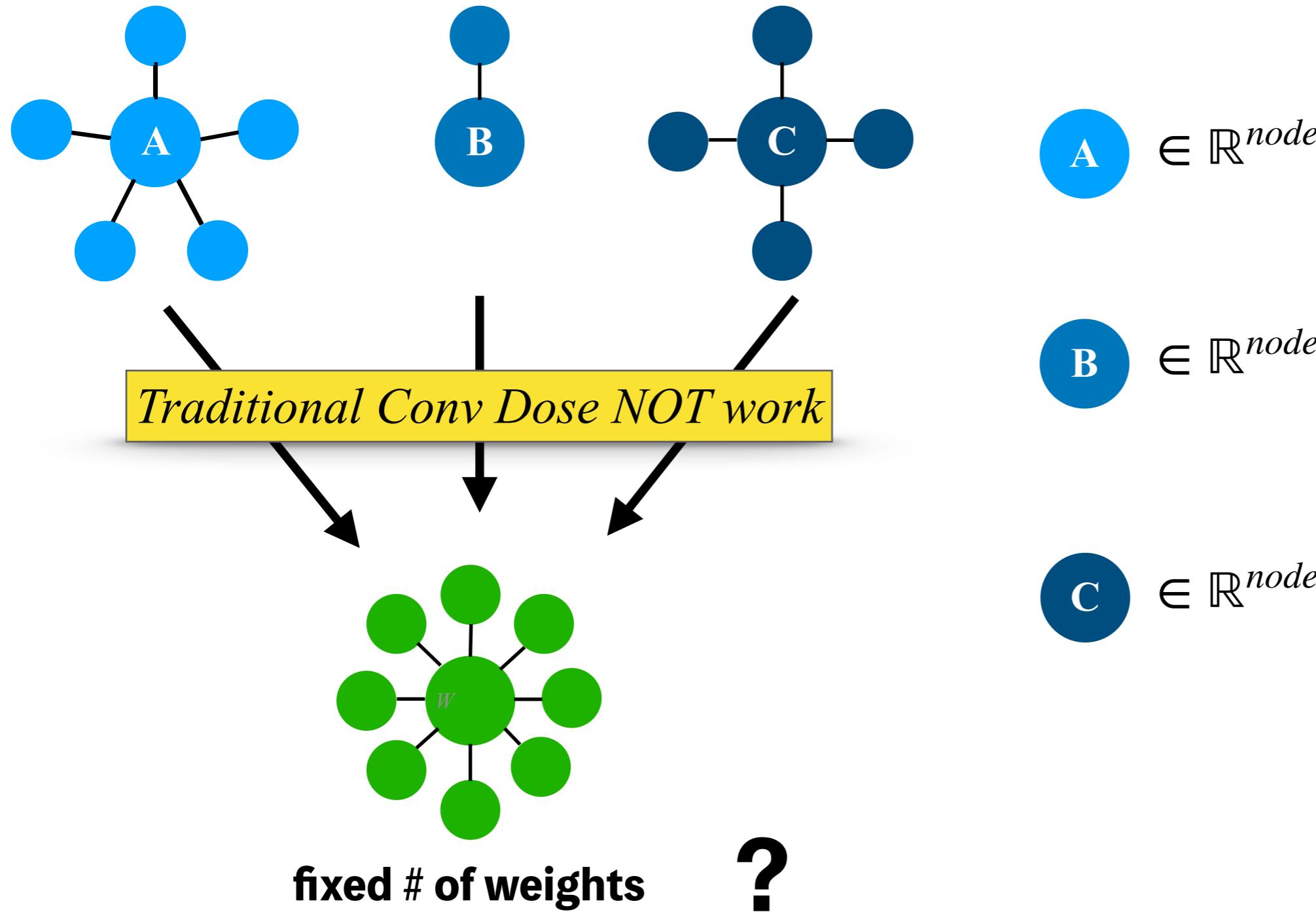
dynamic # of neighbors



$$\begin{array}{rcl} A & += & \textcolor{blue}{\bullet} + \textcolor{blue}{\bullet} + \textcolor{blue}{\bullet} + \textcolor{blue}{\bullet} + \textcolor{blue}{\bullet} \\ & & \hline & & 5 \\ B & += & \textcolor{blue}{\bullet} \\ & & \hline & & 1 \\ C & += & \textcolor{darkblue}{\bullet} + \textcolor{darkblue}{\bullet} + \textcolor{darkblue}{\bullet} + \textcolor{darkblue}{\bullet} \\ & & \hline & & 4 \end{array}$$

Convolution on Graph Data

dynamic # of neighbors



Convolution on Graph Data

If constraint g_θ is polynomial, rational or exp function

$$g_\theta * f = U g_\theta U^\top f = \boxed{f} + \boxed{A \cdot f}$$

Self Average of neighbors

$$\text{A} += \frac{\text{blue circle} + \text{blue circle} + \text{blue circle} + \text{blue circle} + \text{blue circle}}{5}$$

$$\text{B} += \frac{\text{blue circle}}{1}$$

$$\text{C} += \frac{\text{dark blue circle} + \text{dark blue circle} + \text{dark blue circle} + \text{dark blue circle}}{4}$$

Convolution on Graph Data

If constraint g_θ is polynomial, rational or exp function

$$g_\theta * f = U g_\theta U^\top f = \boxed{f} + \boxed{A \cdot f}$$

Self Average of neighbors

A $\quad + = \frac{\bullet + \bullet + \bullet + \bullet + \bullet}{5}$

B $\quad + = \frac{\bullet}{1}$

Convolution theorem

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

Space convolution = frequency multiplication

Agenda



● First Half (40 min)

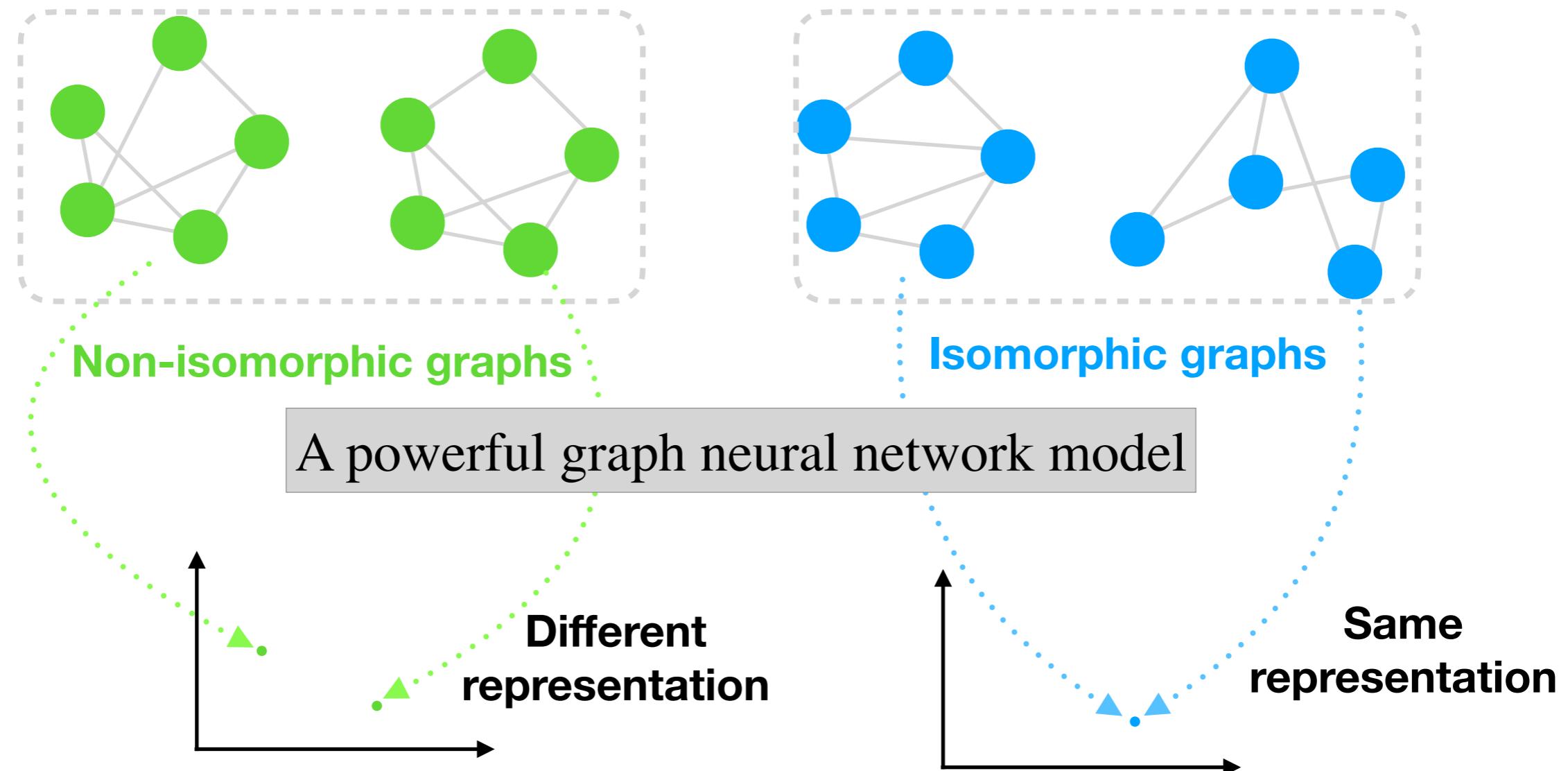
- *Preliminary: graph convolutions (25 min)*
- *Analysis: expressive power (15 min)*

● Second Half (50 min)

- *Introduction: a new unified framework (45 min)*
- *Future directions (5 min)*

Attempts to Unify GNNs

Expressive Power of Spatial 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?

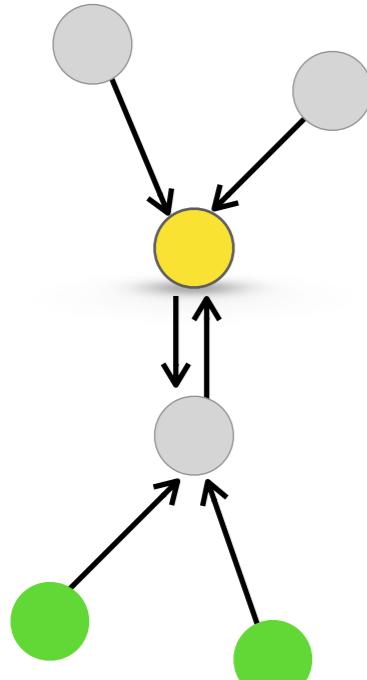
For MLP: A feedforward neural network (such as an MLP) with at least one hidden layer and a sufficient number of neurons can approximate any continuous function on a compact subset of R^n , to any desired degree of accuracy, provided the activation function is non-linear (e.g., sigmoid, ReLU).

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

$$f_\theta \rightarrow f^*$$

How about in graphs?

Expressive Power of Spatial GNNs



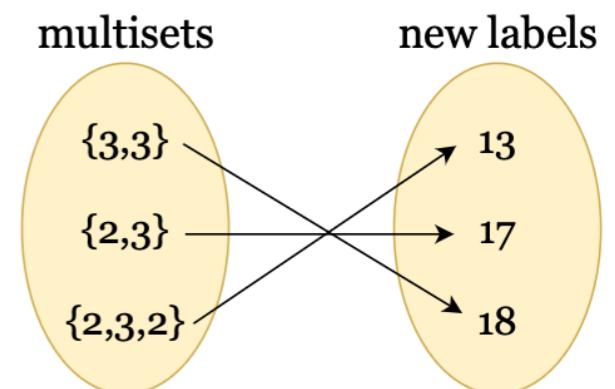
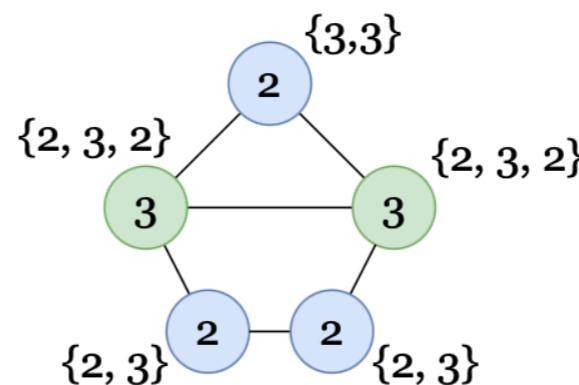
GNNs are defined as a composition of

- **AGGREGATE** functions and
- **READOUT** functions

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

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

- *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

Expressive Power of Spectral GNNs

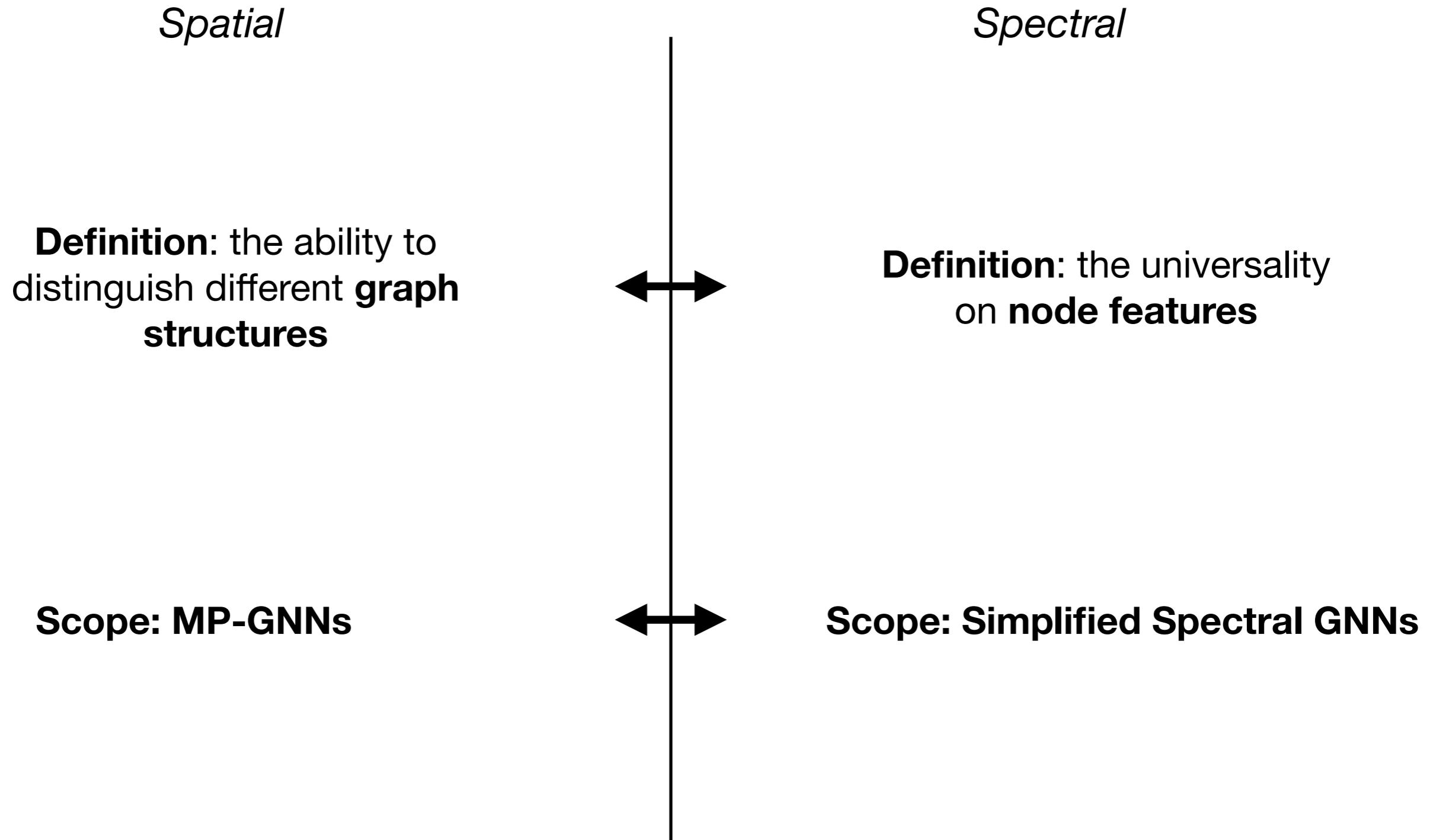
GNNs are defined as a simplified polynomial Spectral GNN:

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

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*

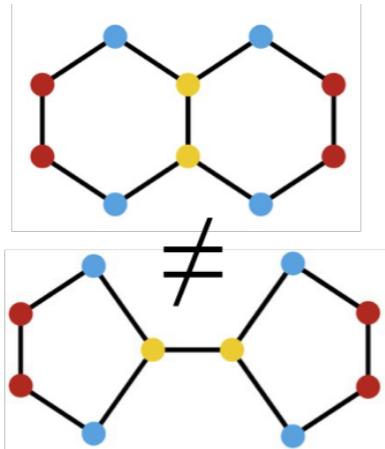
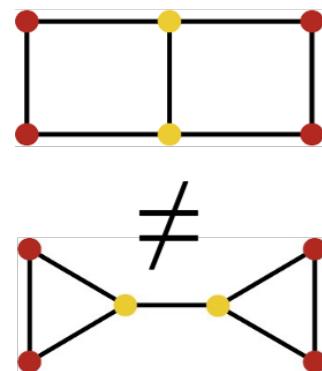
Expressive Power



Expressive Power

Spatial

- What 1-WL cannot distinguish:



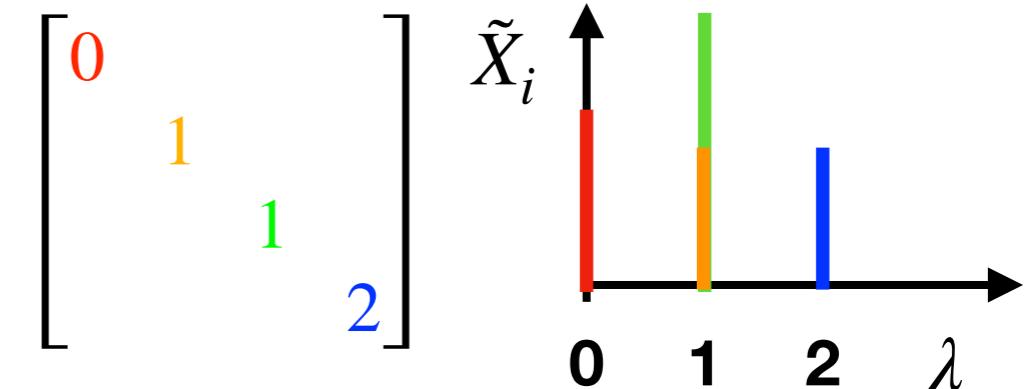
- Does not consider node features

Compatible

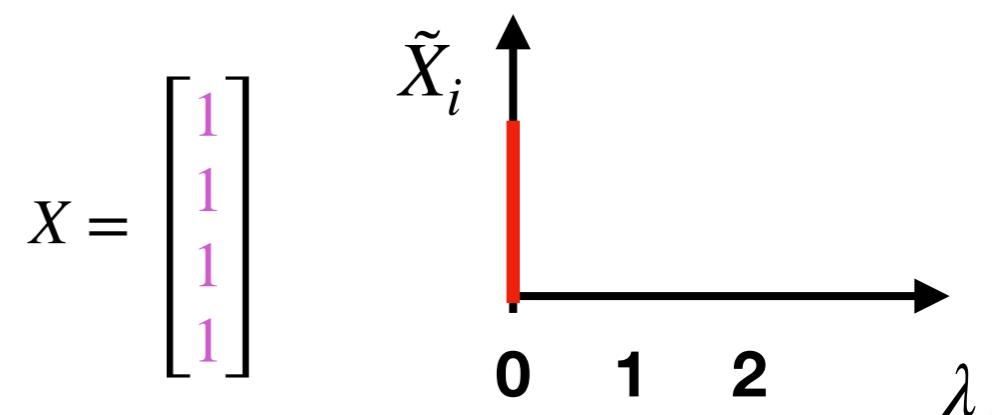
Difference

Spectral

- No multiple eigenvalue
 - A Graph has low symmetry
 - Order of graph's automorphism < 3



- No missing component
 - Nodes have rich features



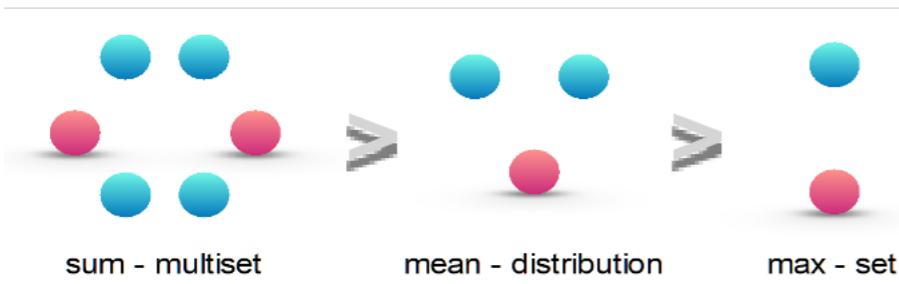
Expressive Power

Spatial

- Upper bound: 1-WL test

- Sum Aggregation

Ranking by discriminative power:



Spectral

- Universal approximator
 - With constraints

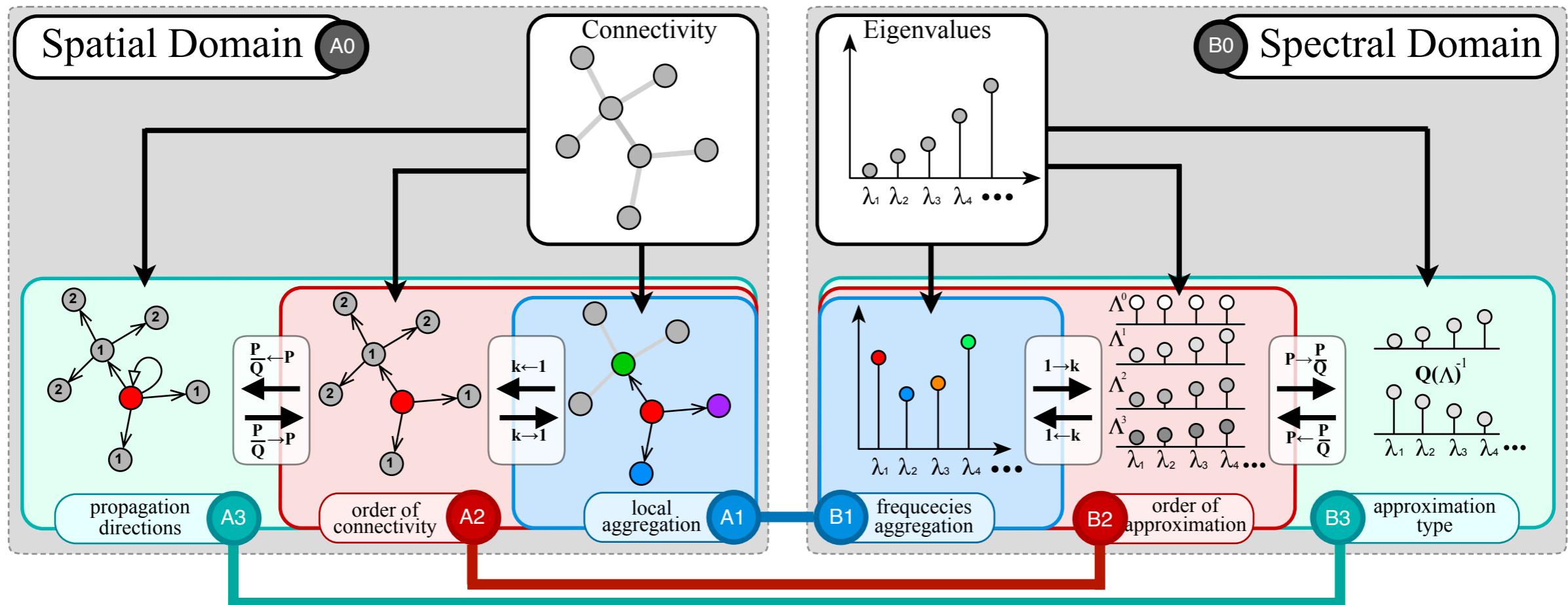
- Normalization makes more sense

Progressive
↔

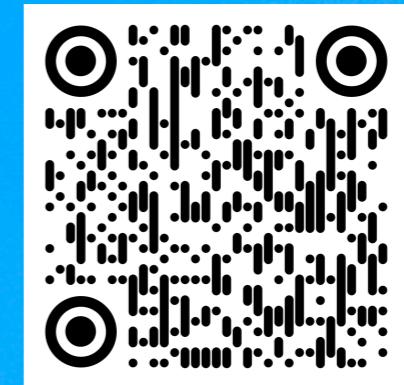
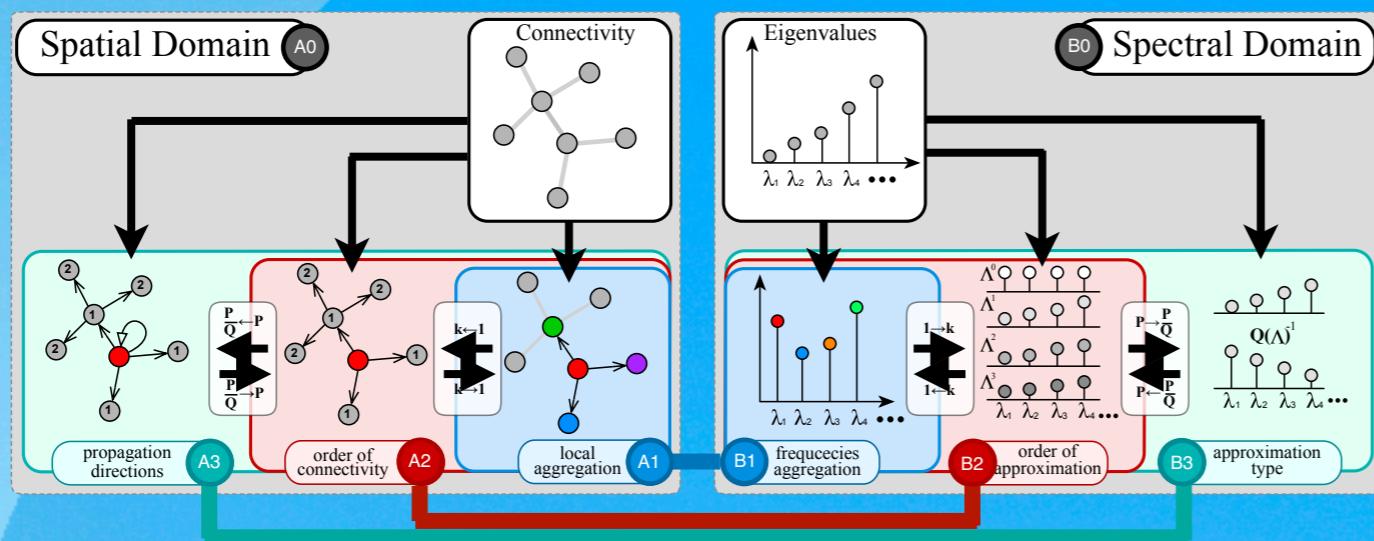
Conflicts
↔

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



Half Time Break



Paper Collection

Agenda



- First Half (40 min)
 - *Background: unified frameworks for GNN* (25 min)
 - *Preliminary: graph convolutions* (15 min)
- Second Half (50 min)
 - ***Introduction: a new unified framework*** (45 min)
 - *Future directions* (5 min)

Agenda



- **Formal Definitions of Spectral/Spatial method**
- **Normalization**
- **The framework**
 - 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

$$g(\Lambda, X) = U g_\theta(\Lambda) U^T X$$

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

Normalization and its Reasons

- Spatial reason

- Suppose a two-cluster partitioning for A and B

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

Unnormalized

- Use **# node** to cluster graph

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

Normalized

- Use **# link** to cluster graph

Normalization and its Reasons

● Spectral reason

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

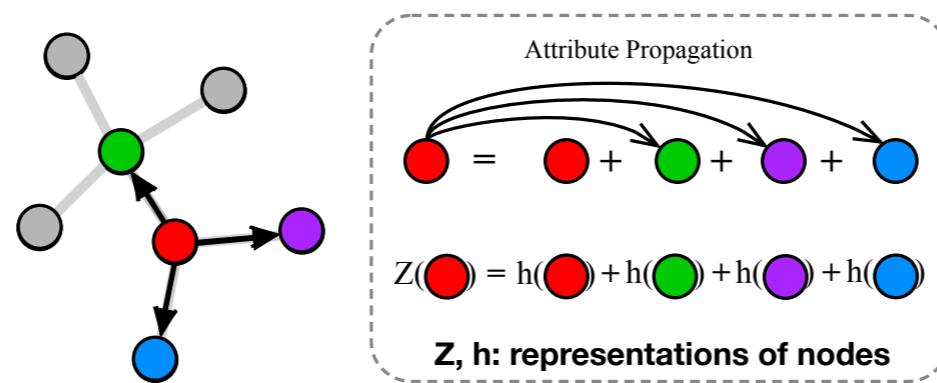
Unnormalized

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

Normalized

Case Study 1: GCN

The “GCN” Thomas N. Kipf et al. (2017)



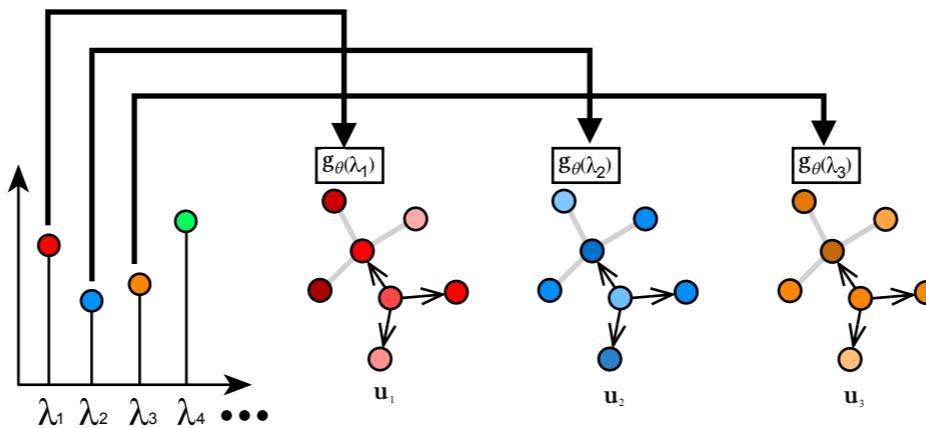
$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}} \hat{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}} \mathbf{X} = \mathbf{D}^{-\frac{1}{2}} (\mathbf{I} + \mathbf{A}) \mathbf{D}^{-\frac{1}{2}} \mathbf{X} = (\mathbf{I} + \tilde{\mathbf{A}}) \mathbf{X}$$



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)*

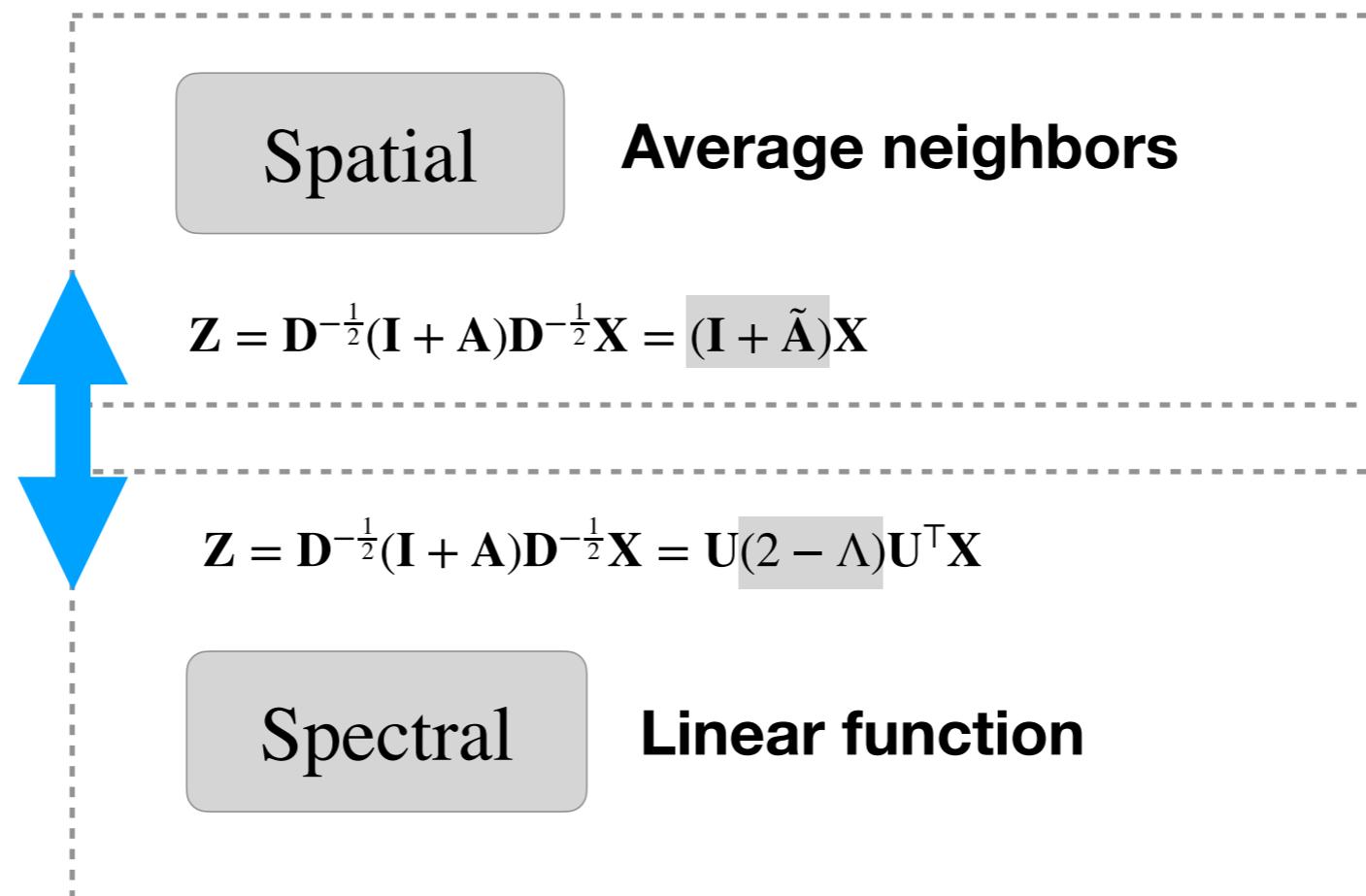


$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{L} + \mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{U}(2 - \boldsymbol{\Lambda})\mathbf{U}^\top \mathbf{X}$$

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

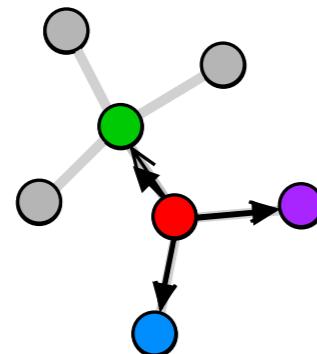
Case Study 1: GCN

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

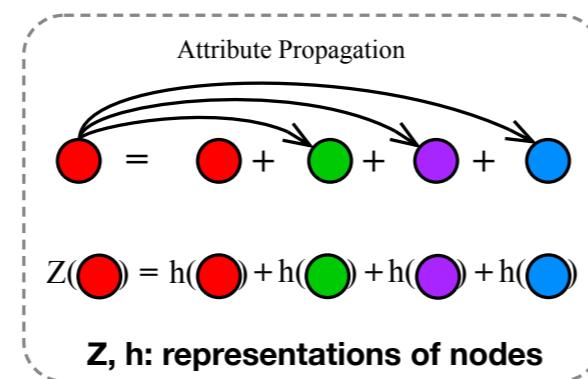


Spatial-based GNN: Linear

Linear



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



GCN Thomas N. Kipf et al. (2016)

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

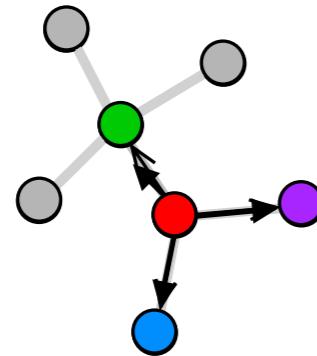
GraphSAGE

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

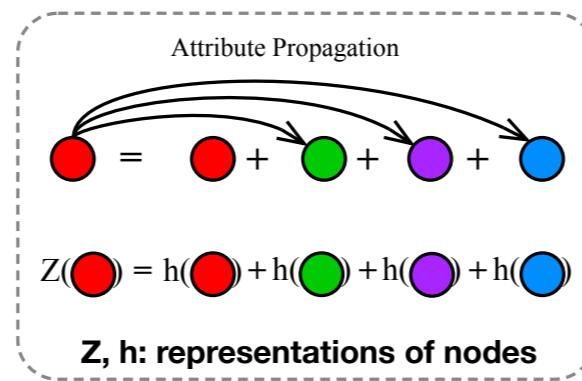
w/ mean aggregator

Spatial-based GNN: Linear

Linear



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



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)

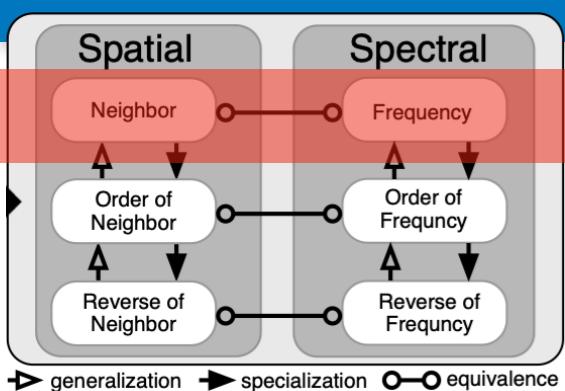
$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}} (\mathbf{I} + \mathbf{A}) \mathbf{D}^{-\frac{1}{2}} \mathbf{X} = (\mathbf{I} + \tilde{\mathbf{A}}) \mathbf{X}$$

GIN

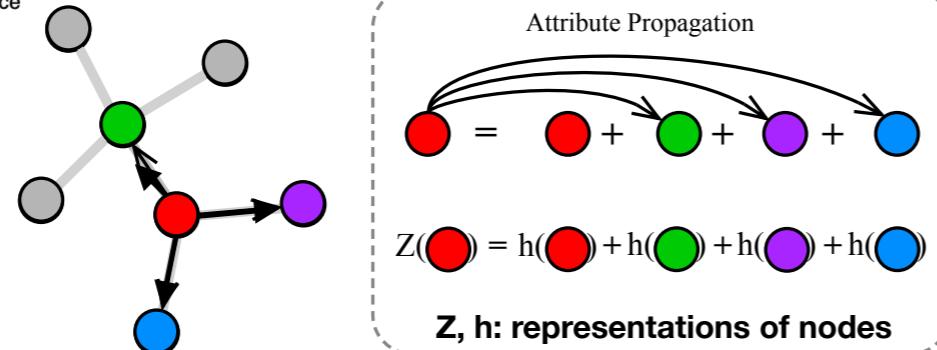
$$\mathbf{Z} = (1 + \epsilon) \cdot \mathbf{h}(v) + \sum_{u_j \in \mathcal{N}(v_i)} \mathbf{h}_{(u_j)} = [(1 + \epsilon) \mathbf{I} + \mathbf{A}] \mathbf{X}$$

Control hyperparameter

Spatial-based GNN: Linear



Linear



Function of graph matrix

$$g(A)X$$

GCN Thomas N. Kipf et al. (2016)

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

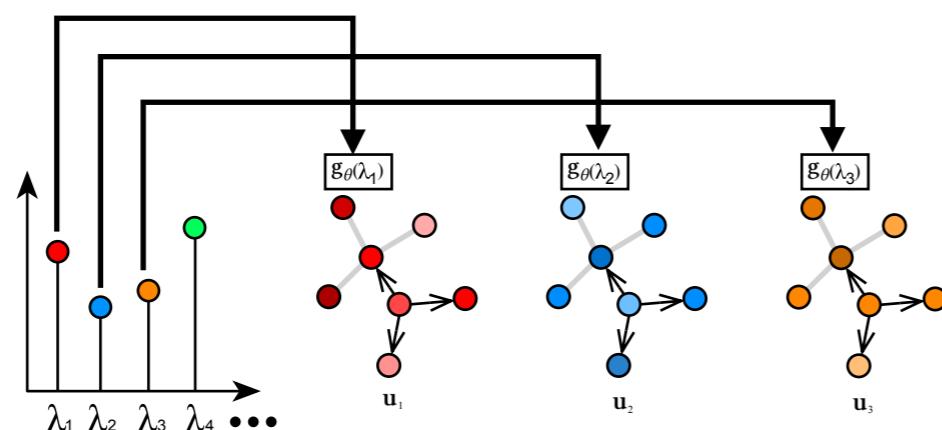
GraphSAGE Will Hamilton et al. (2017)

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

GIN

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

Linear



Function of eigenvalue

$$U g_\theta(\Lambda) U^T X$$

GCN Thomas N. Kipf et al. (2016)

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

GraphSAGE Will Hamilton et al. (2017)

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

GIN Xukeyu Lu et al. (2019)

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

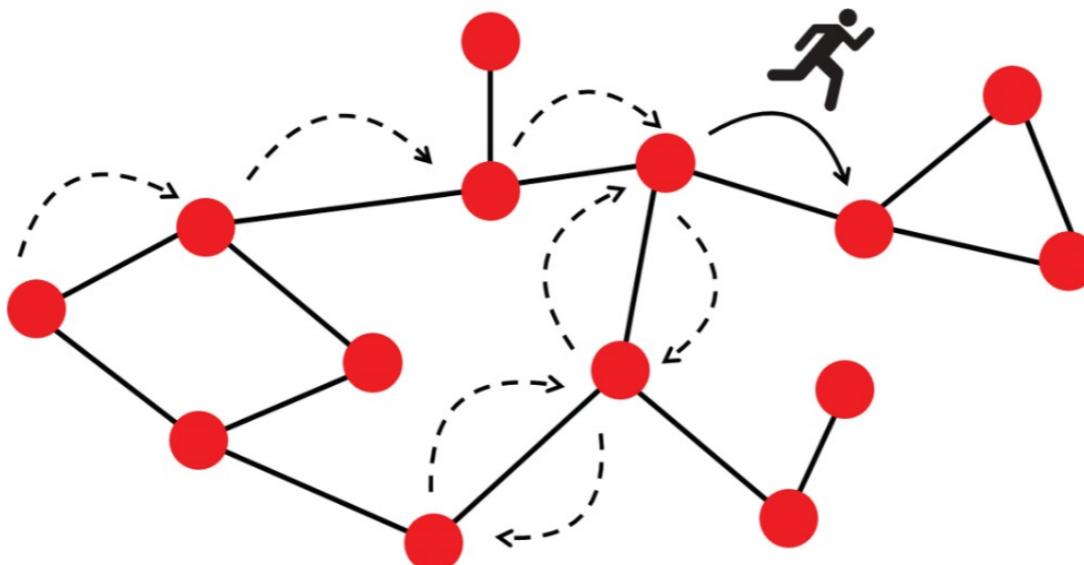
Case Study 2: DeepWalk

- 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 $2t+1$ and the current node is the $(t+1)$ -th

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



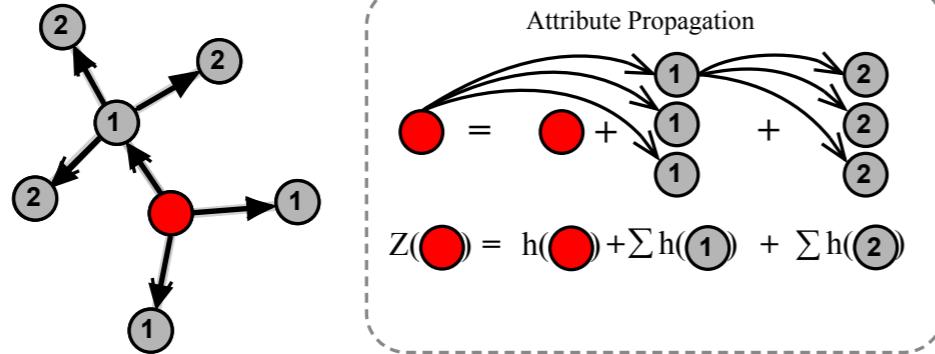
Img credit: DOI: (10.1002/sim.9346)

Spectral-based GNN: Polynomial

Function of **graph matrix**

$$g(A)X$$

Polynomial



DeepWalk *Bryan Perozzi et al. (2014)*

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

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

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



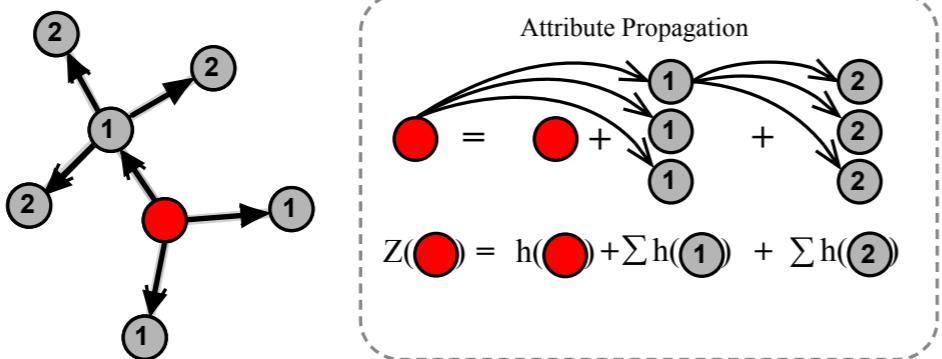
Chebyshev polynomial (1st kind) of L

Spectral-based GNN: Polynomial

Function of **graph matrix**

$$g(\tilde{\mathbf{A}})\mathbf{X}$$

Polynomial



DeepWalk *Bryan Perozzi et al. (2014)*

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

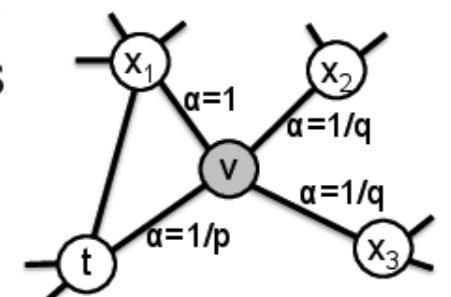
ChebyNet *Defferrard, Michael et al. (2016)*

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

Node2Vec *Aditya Grover et al. (2016)*

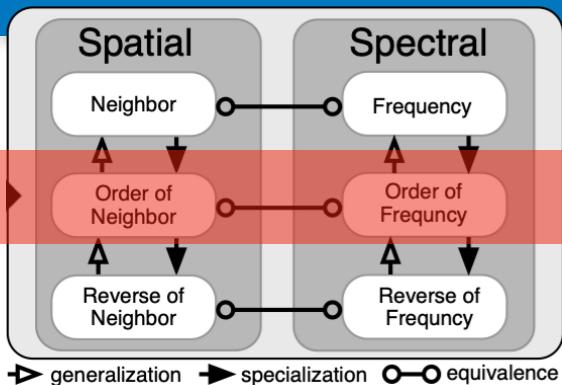
$$\mathbf{Z} = \left(\frac{1}{p} \cdot \underbrace{\mathbf{I}}_{\text{source}} + \underbrace{\tilde{\mathbf{A}}}_{\text{BFS}} + \frac{1}{q} \underbrace{(\tilde{\mathbf{A}}^2 - \tilde{\mathbf{A}})}_{\text{DFS}} \right) \mathbf{X}$$

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

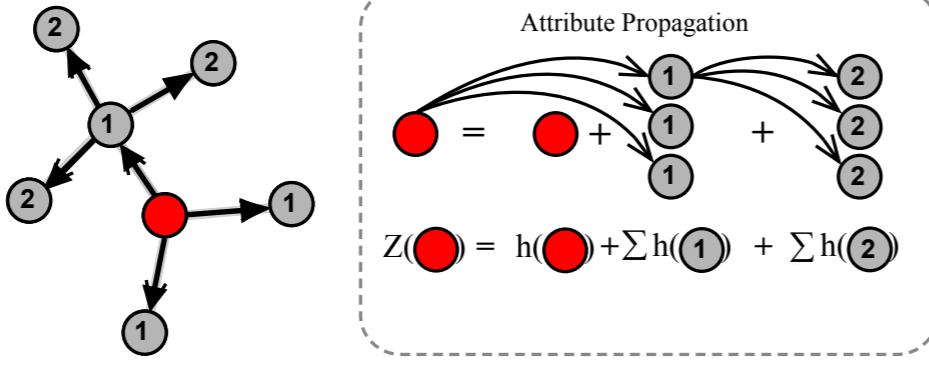


$$\mathbf{Z} = \left(\frac{1}{p} \cdot \mathbf{I} + \tilde{\mathbf{A}} + \frac{1}{q} (\tilde{\mathbf{A}}^2 - \tilde{\mathbf{A}}) \right) \mathbf{X} = \left[\frac{1}{p} \mathbf{I} + \left(1 - \frac{1}{q} \right) \tilde{\mathbf{A}} + \frac{1}{q} \tilde{\mathbf{A}}^2 \right] \mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})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

Spectral-based GNN: Polynomial



Polynomial



Function of graph matrix $g(A)X$

DeepWalk Bryan Perozzi et al. (2014)

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

ChebyNet Defferrard, Michael et al. (2016)

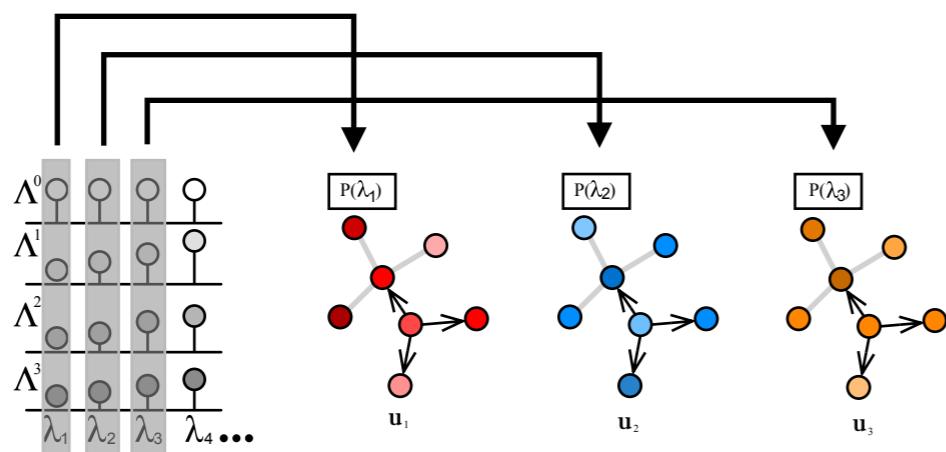
$$\mathbf{Z} = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathbf{L}}) \mathbf{X} = [\tilde{\theta}_0 \mathbf{I} + \tilde{\theta}_1 (\mathbf{I} - \tilde{\mathbf{A}}) + \tilde{\theta}_2 (\mathbf{I} - \tilde{\mathbf{A}})^2 + \dots] \mathbf{X} = (\phi \mathbf{I} + \sum_{i=1}^k \psi_i \tilde{\mathbf{A}}^i) \mathbf{X}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

Node2Vec Aditya Grover et al. (2016)

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

Function of eigenvalue $U g_\theta(\Lambda) U^T X$

Polynomial



DeepWalk Bryan Perozzi et al. (2014)

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

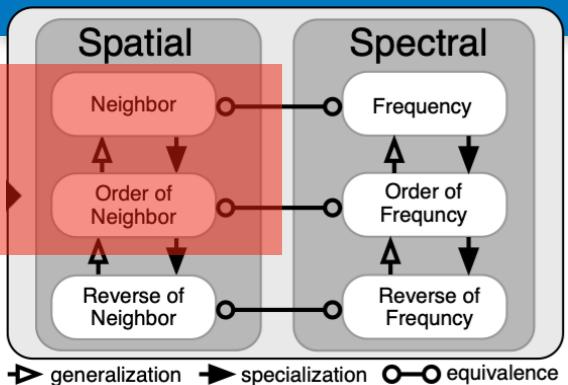
ChebyNet Defferrard, Michael et al. (2016)

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

Node2Vec Aditya Grover et al. (2016)

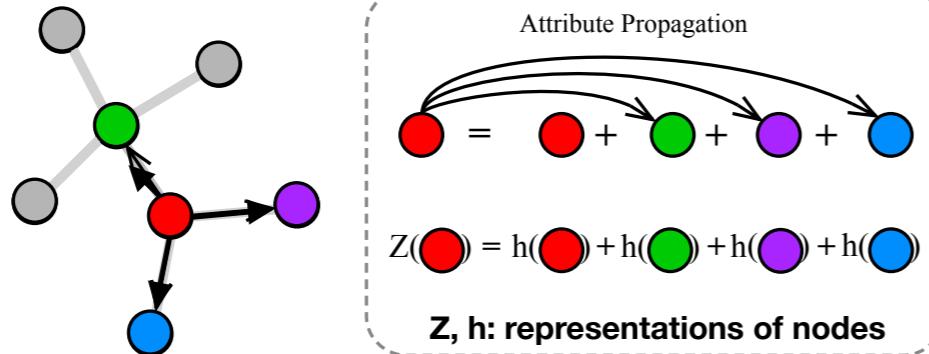
$$\mathbf{Z} = \left[\left(1 + \frac{1}{p} \right) \mathbf{I} - \left(1 + \frac{1}{q} \right) \tilde{\mathbf{L}} + \frac{1}{q} \tilde{\mathbf{L}}^2 \right] \mathbf{X} = \mathbf{U} \left[\left(1 + \frac{1}{p} \right) - \left(1 + \frac{1}{q} \right) \tilde{\Lambda} + \frac{1}{q} \tilde{\Lambda}^2 \right]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

Linear v.s. Polynomial

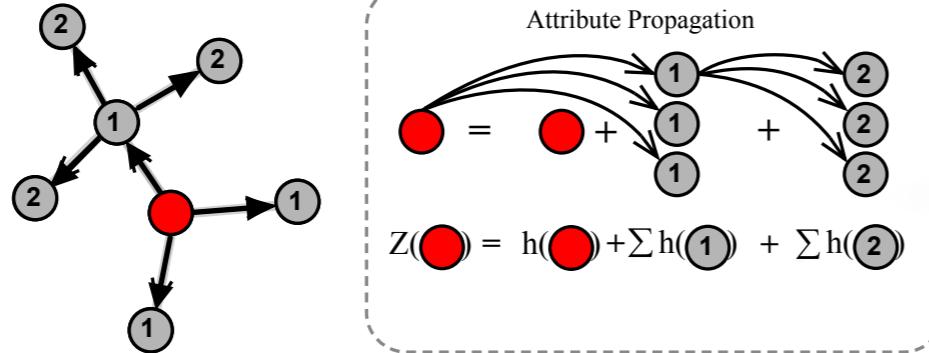


Function of **graph matrix** $g(A)X$

Linear



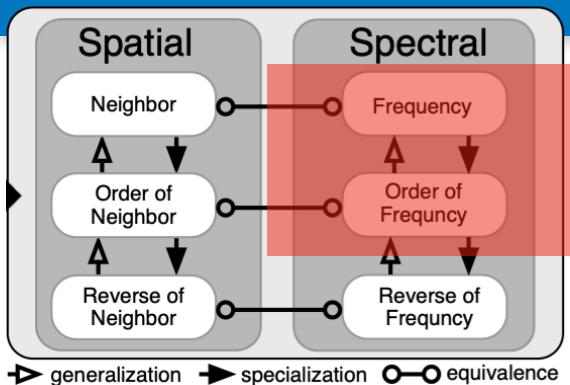
Polynomial



only consider the direct neighbors

consider the higher-order neighbors

Linear v.s. Polynomial

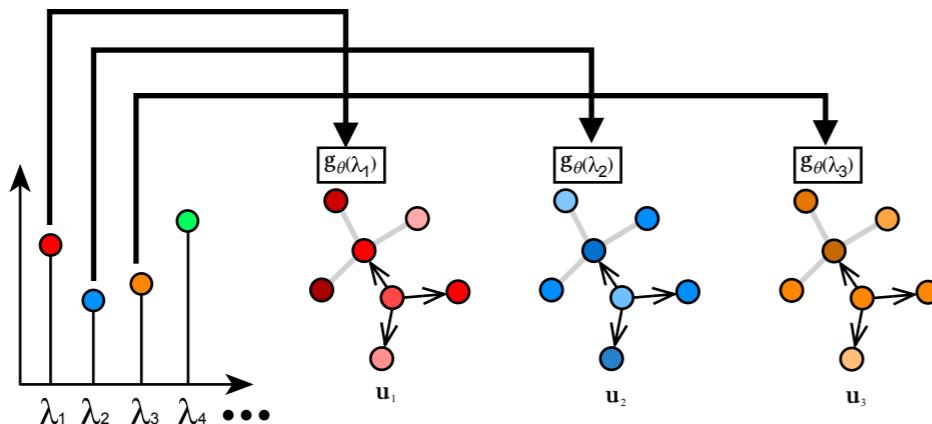


Function of eigenvalue

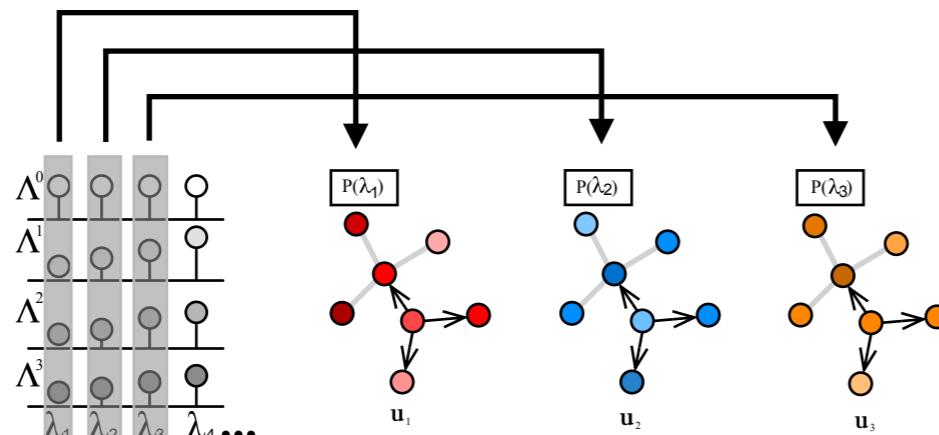
“spectral response functions”

$$Ug_{\theta}(\Lambda)U^T X$$

Linear



Polynomial



GCN Thomas N. Kipf et al. (2016)

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

GraphSAGE Will Hamilton et al. (2017)

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

GIN Xukeyu Lu et al. (2019)

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

DeepWalk Bryan Perozzi et al. (2014)

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

ChebyNet Defferrard, Michael et al. (2016)

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

Node2Vec Aditya Grover et al. (2016)

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

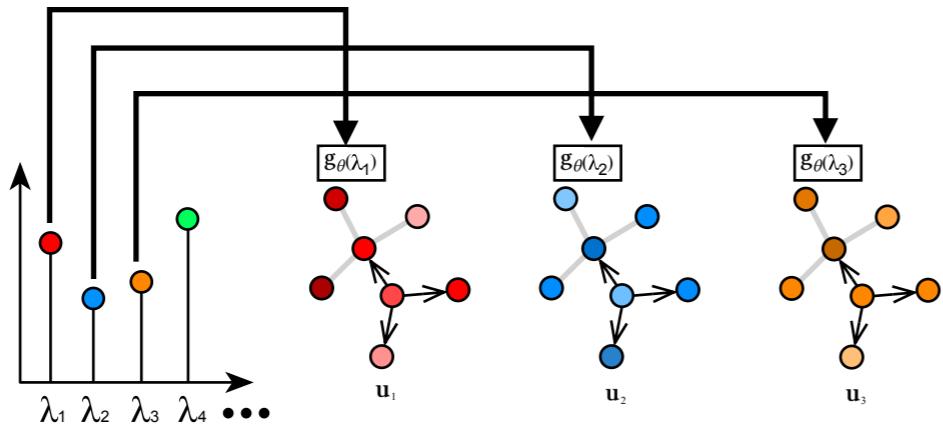
Linear v.s. Polynomial

“spectral response functions”

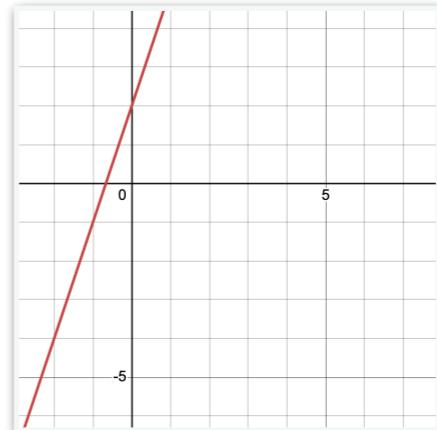
Function of eigenvalue

$$Ug_{\theta}(\Lambda)U^T X$$

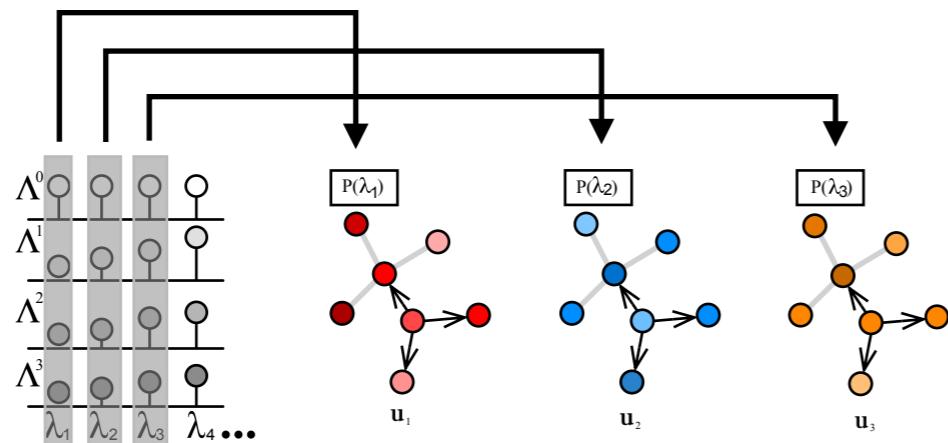
Linear



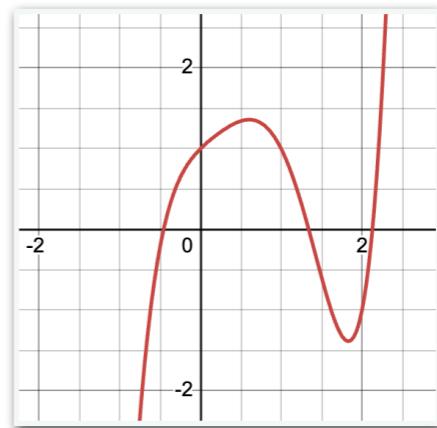
$$y = 3x + 2$$



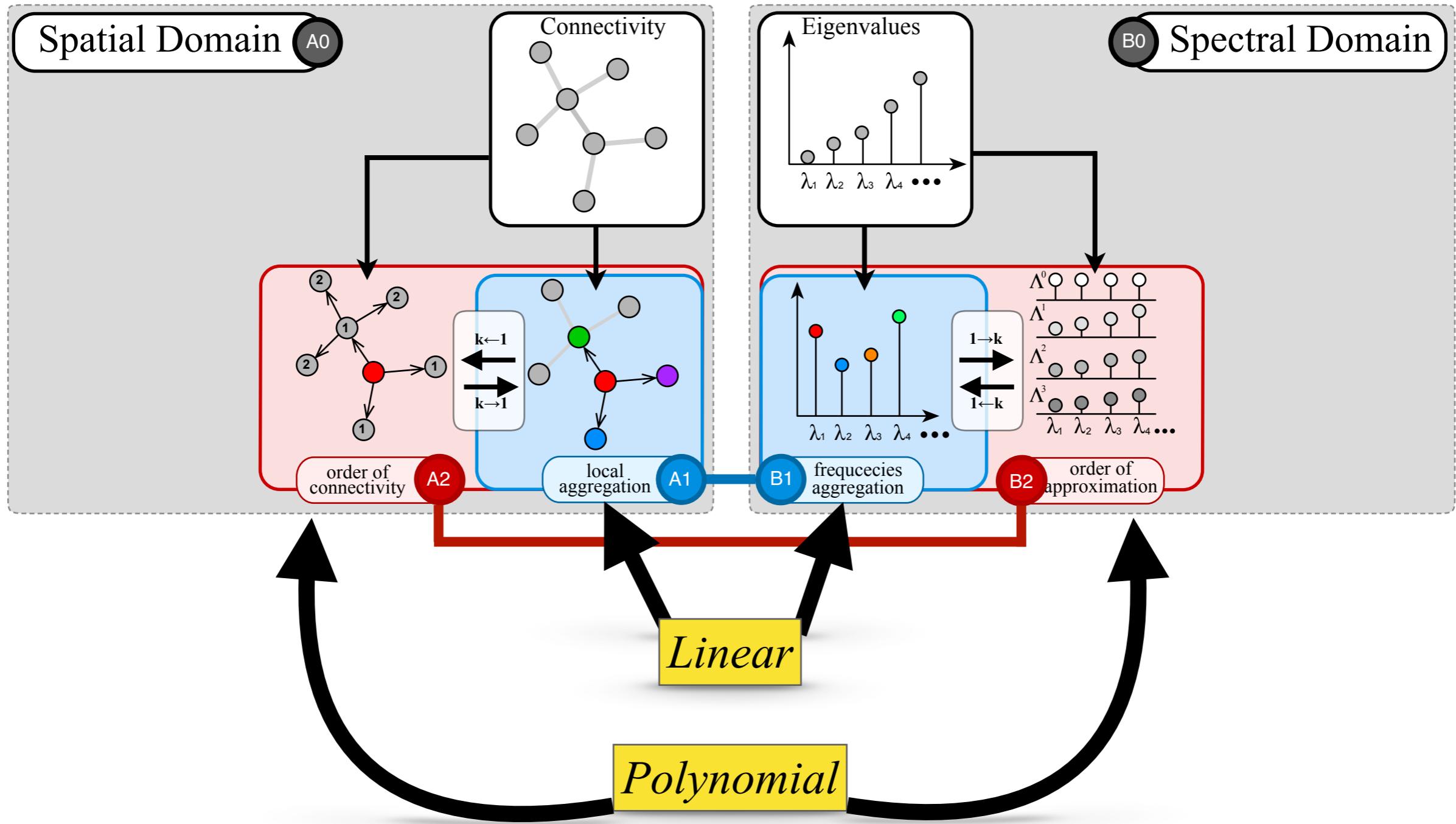
Polynomial



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



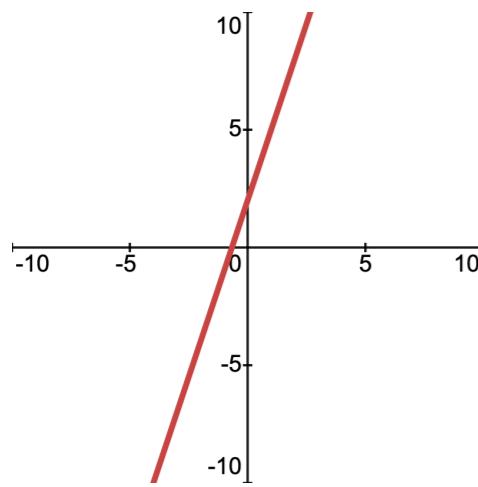
Linear v.s. Polynomial



Beyond Polynomial: Rational Model

GCN

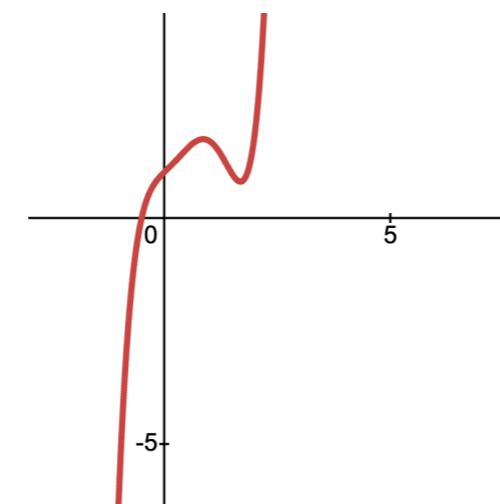
$$y = 3x + 2$$



Linear

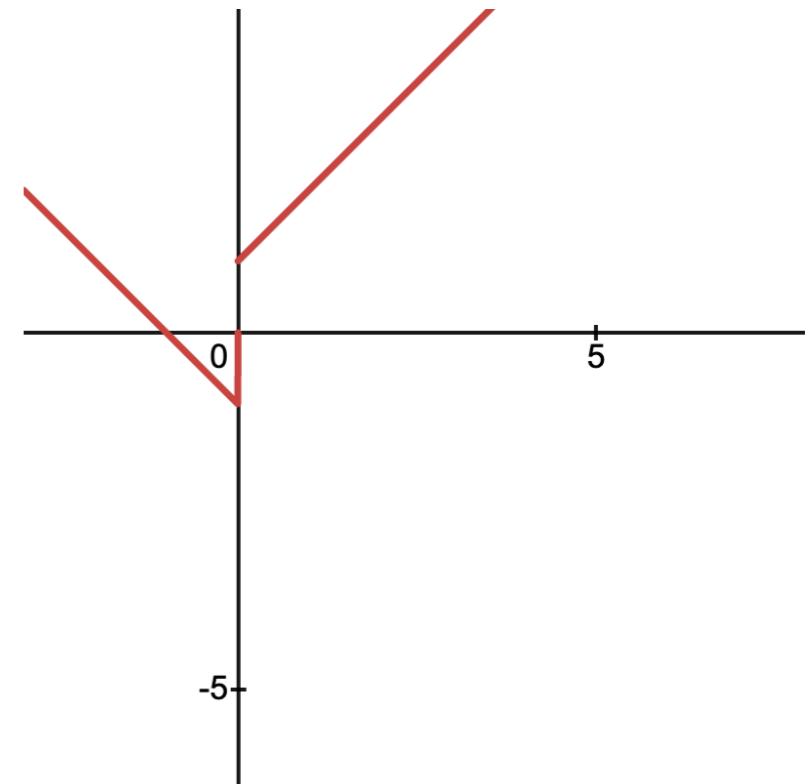
ChebNet

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



Polynomial

What if non-smooth function



Beyond Polynomial: Rational Model

Polynomial approximation

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

Cheaper
Less accurate

Rational approximation

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

More expensive
More 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



moderately simple form, not well-known properties

moderately **easy** to handle computationally



excellent for oscillations between exact-fit value

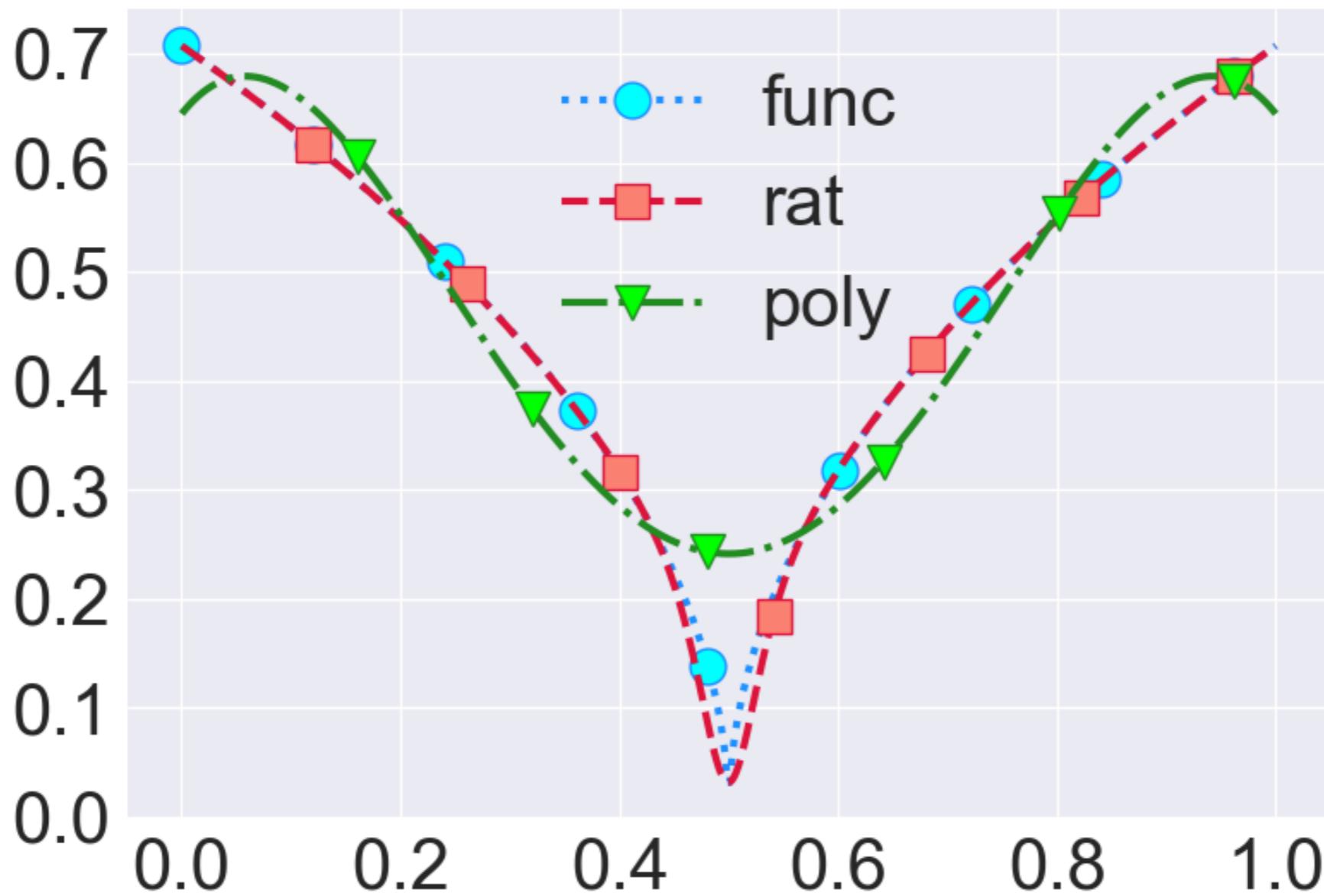
model **complicated** structure with a fairly low degree

excellent interpolatory/extrapolatory/asymptotic properties

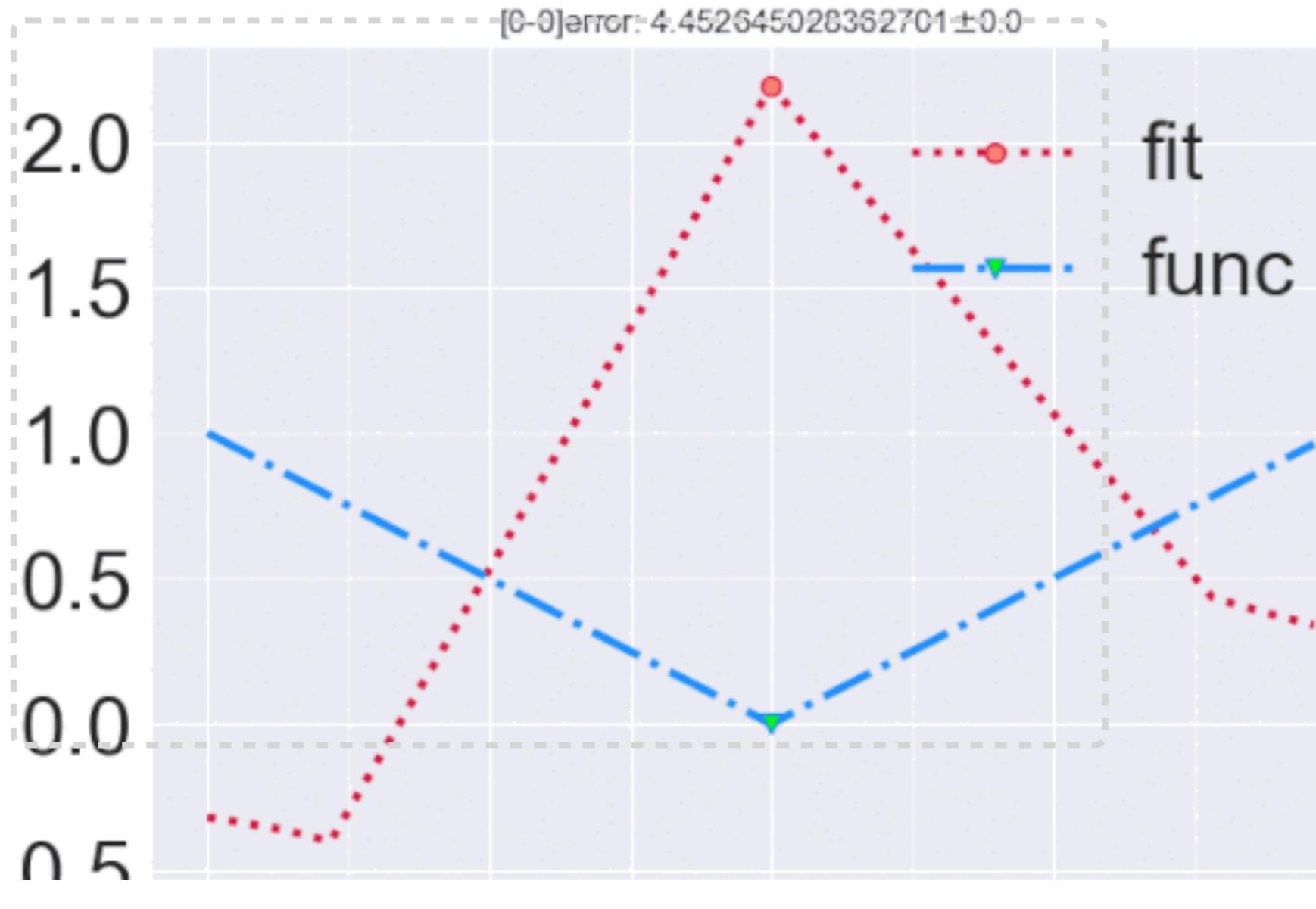


Beyond Polynomial: Rational Model

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



Beyond Polynomial: Rational Model



Rational Neural Network: iteratively close to the target

Polynomial v.s. Rational

Polynomial

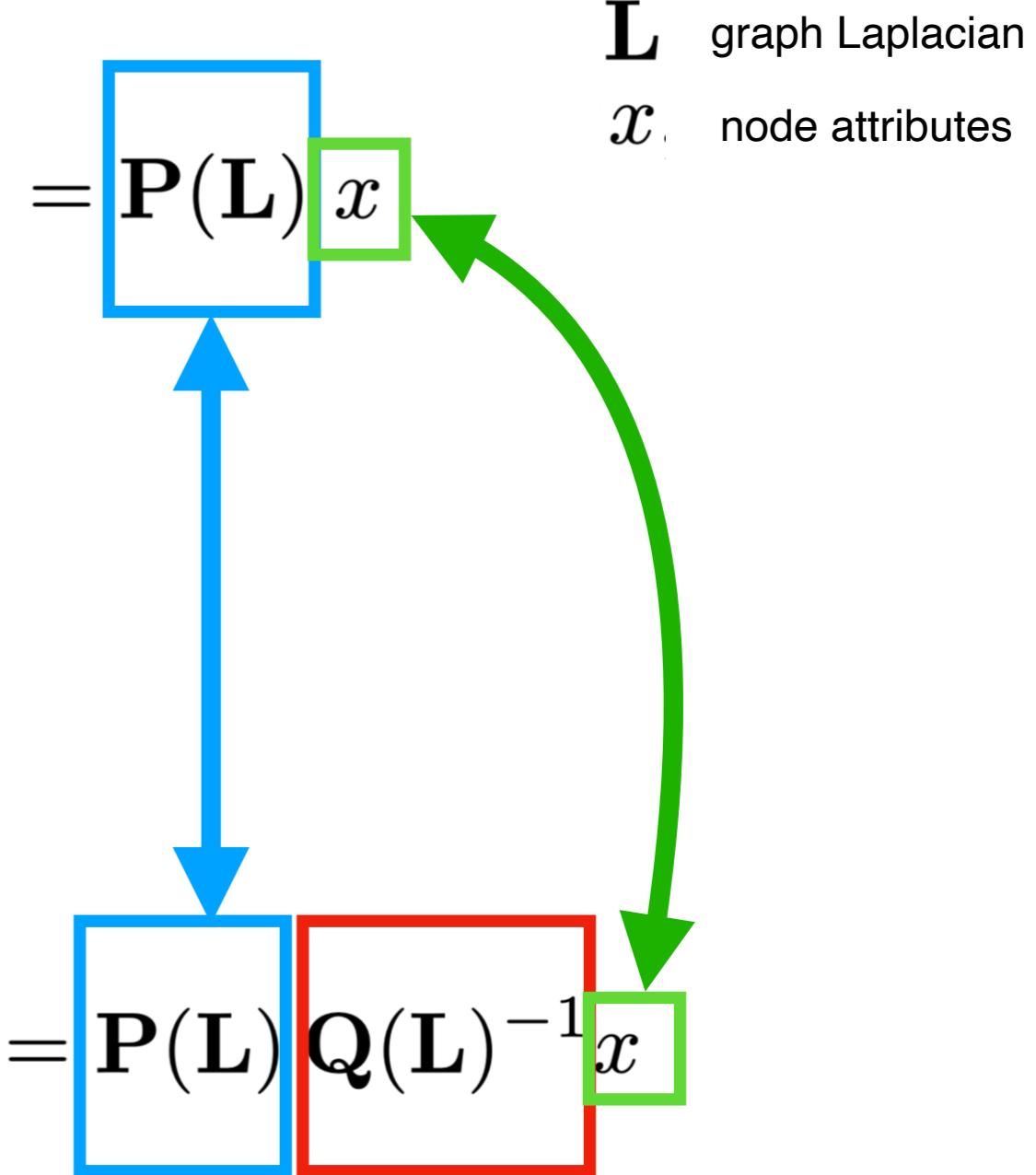
Thomas N. Kipf et al. (2016)

$$\begin{aligned}
 g * x &= \mathbf{U} g(\Lambda) \mathbf{U}^\top x \\
 &\approx \mathbf{U} \sum_k \theta_k T_k(\tilde{\Lambda}) \mathbf{U}^\top x \quad (\tilde{\Lambda} = \frac{2}{\lambda_{max}} \Lambda - \mathbf{I}_N) \\
 &= \sum_k \theta_k T_k(\tilde{\Lambda}) x \quad (\mathbf{U} \Lambda^k \mathbf{U}^\top = (\mathbf{U} \Lambda \mathbf{U}^\top)^k) \\
 &= \mathbf{P}(\mathbf{L}) x
 \end{aligned}$$

Rational

Z. Chen et al. (2018)

$$\begin{aligned}
 g_\theta * x &= \mathbf{U} g_\theta \mathbf{U}^\top x \\
 &\approx \mathbf{U} \frac{\sum_{i=0}^m \psi_i \tilde{\Lambda}^i}{1 + \sum_{j=1}^n \phi_j \tilde{\Lambda}^j} \mathbf{U}^\top x \quad (\text{convolution theorem}) \\
 &= \mathbf{U} \frac{\mathbf{P}(\Lambda)}{\mathbf{Q}(\Lambda)} \mathbf{U}^\top x, \quad (\text{define P and Q}) \\
 &= \mathbf{P}(\mathbf{L}) \mathbf{Q}(\mathbf{L})^{-1} x
 \end{aligned}$$



Polynomial v.s. Rational

Polynomial

$$P(L) \quad x$$

Label propagation

Rational

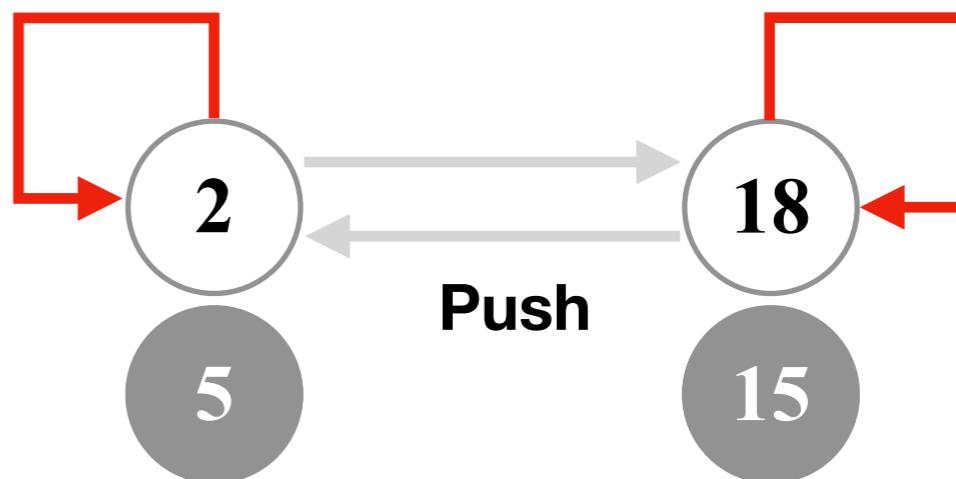
$$P(L) \quad Q(L)^{-1} \quad x$$

Label propagation

Reverse Label propagation

Over-smooth issue

Pull



Polynomial v.s. Rational

Polynomial

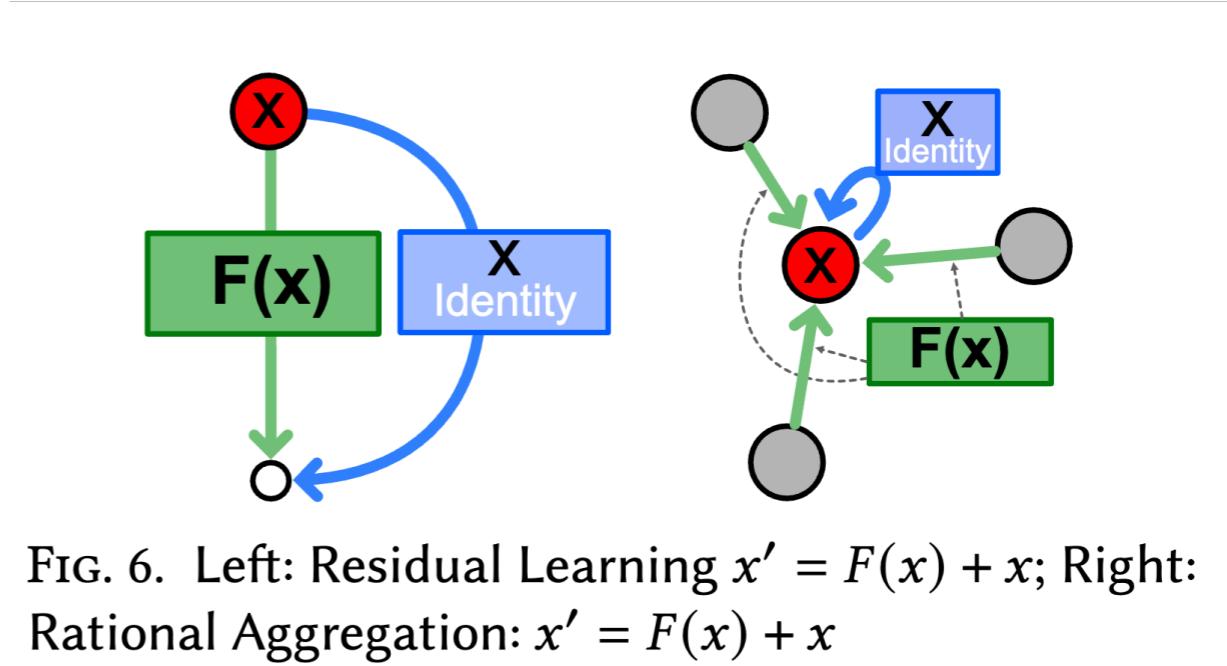
$$\mathbf{P}(\mathbf{L}) \quad x$$

Label propagation

Rational

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

Reverse Label propagation



Polynomial v.s. Rational

Polynomial

$$P(L) \quad x$$

Label propagation

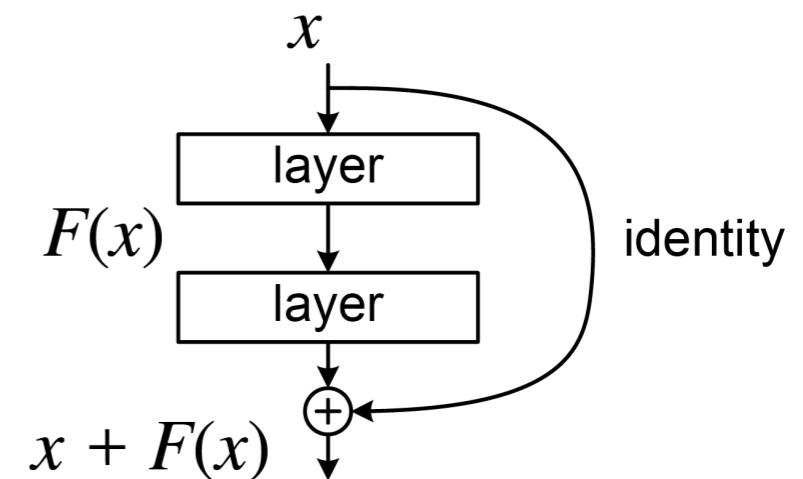
Rational

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

Reverse Label propagation



Difference?



Multi-layer Graph Convolution

Polynomial v.s. Rational

Polynomial

$$P(L) \quad x$$

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

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

Reverse Label propagation

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)

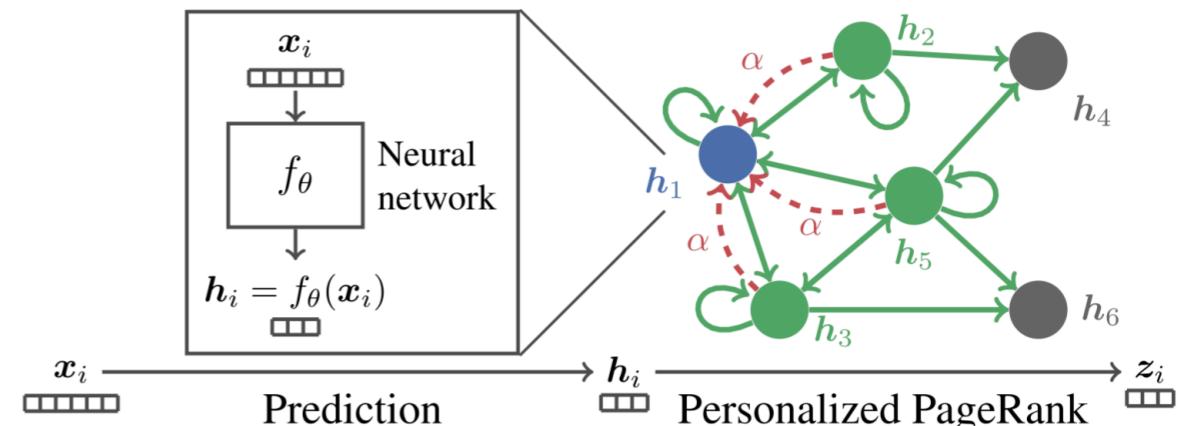
Personalized Page Rank (information retrieval)

$$\pi_{\text{ppr}}(i_x) = (1 - \alpha)\hat{\tilde{A}}\pi_{\text{ppr}}(i_x) + \alpha i_x$$

$(1 - \alpha) \qquad \qquad \qquad \alpha$

PPNP

Personalized Propagation of Neural Predictions



Use personalized PageRank matrix Π_{ppr} to propagate further while retaining information about root node, adjust via teleport probability α :

$$\Pi_{\text{ppr}} = \alpha \left(I_n - (1 - \alpha)\hat{\tilde{A}} \right)^{-1}$$

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

Beyond Polynomial: Rational Model

Johannes Klicpera et al. (2018)

**Personalized Page Rank
(information retrieval)**

$$\pi_{\text{ppr}}(i_x) = (1 - \alpha)\hat{\tilde{A}}\pi_{\text{ppr}}(i_x) + \alpha i_x$$

$(1 - \alpha)$

α

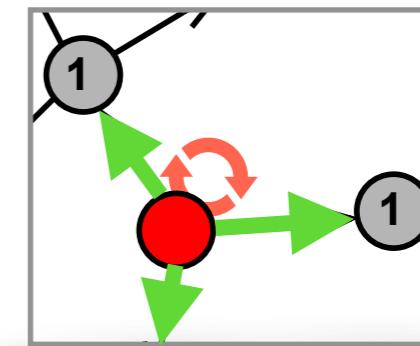
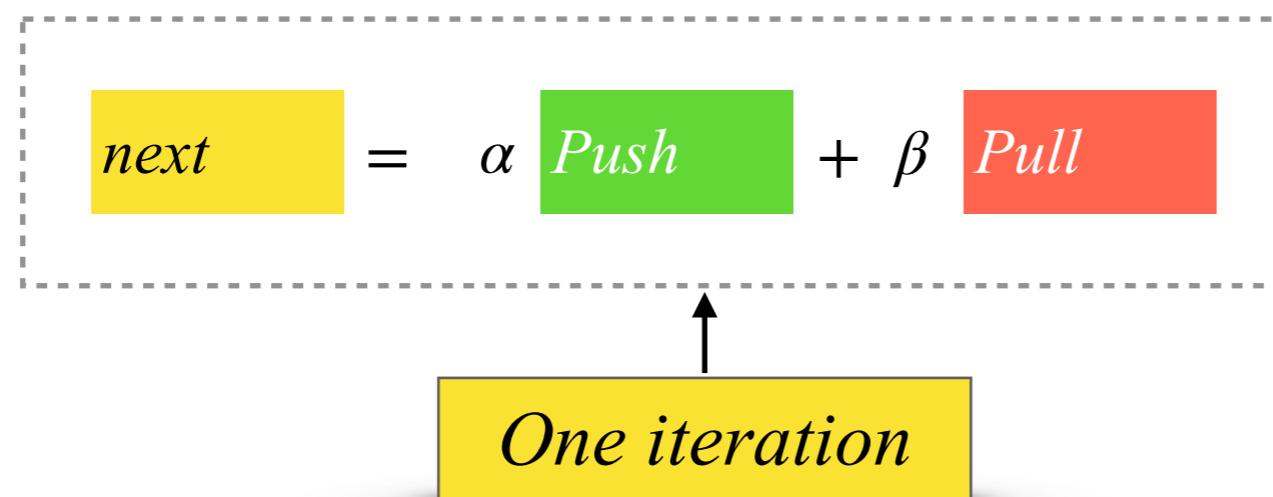
Filippo Maria Bianchi et al. (2018)

**ARMA
(time series)**

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

a

b



K iterations \rightarrow K -order rational function

Why Rational, and Why Not?

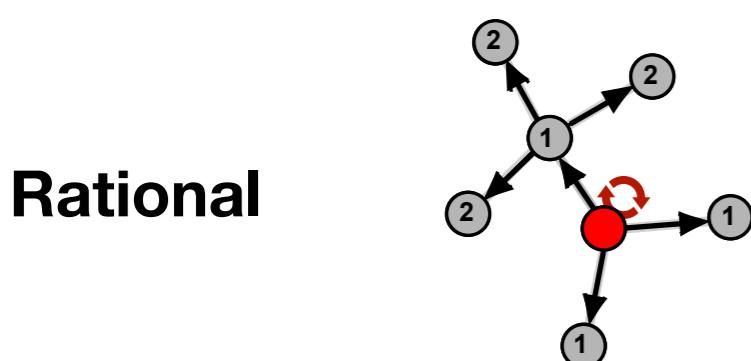
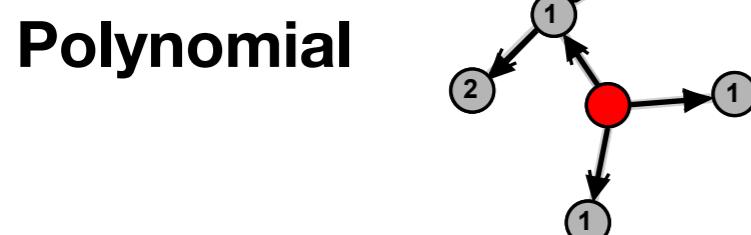
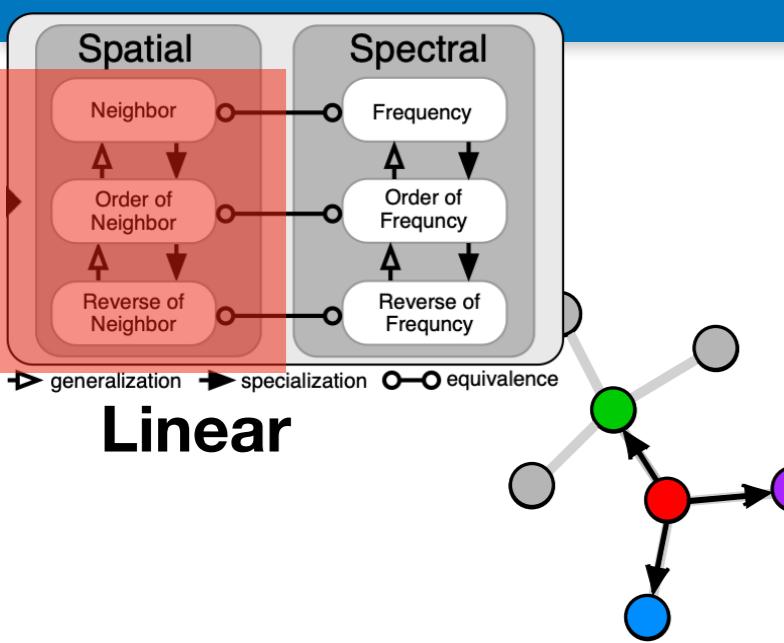
○ Yes

- 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

$$g(A)X$$

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)

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

DeepWalk Bryan Perozzi et al. (2014)

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

ChebyNet Defferrard, Michael et al. (2016)

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

Node2Vec Aditya Grover et al. (2016)

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

Personalized PageRank Johannes Klicpera et al. (2018)

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

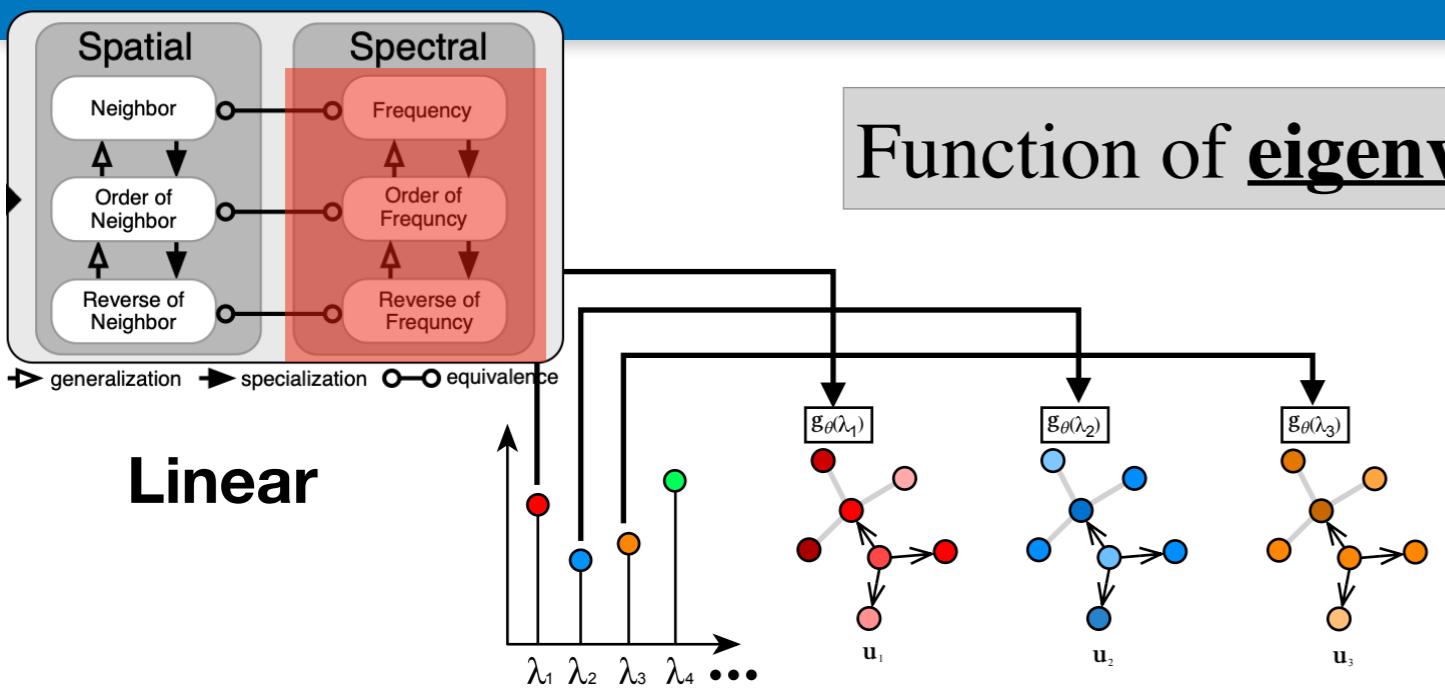
ARMA Filter Filippo Maria Bianchi et al. (2018)

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

Auto Regressive Filter Qimai Li et al. (2019)

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

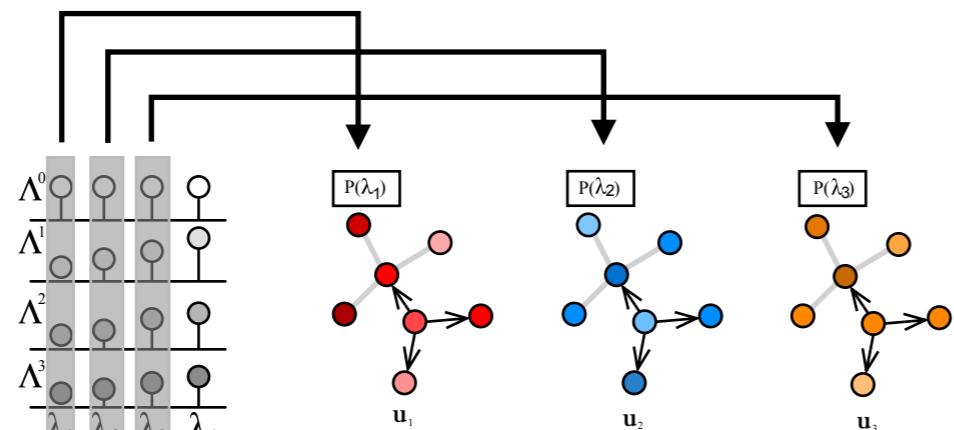
Spectral-based GNN



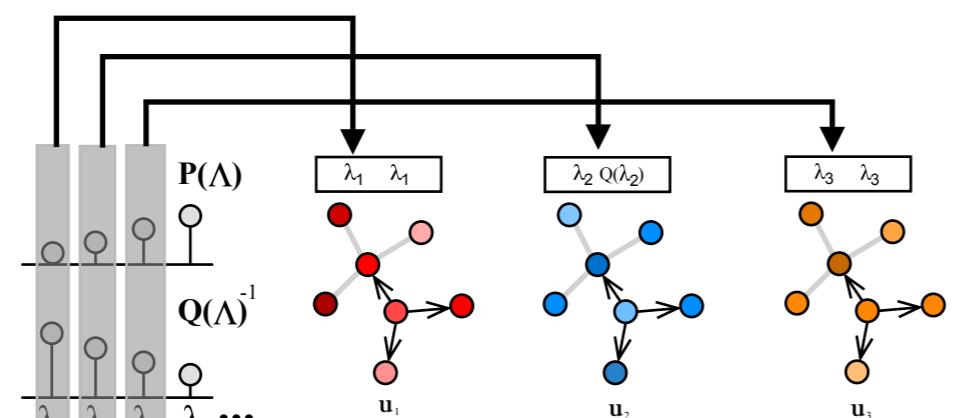
Function of eigenvalue

$$Ug_\theta(\Lambda)U^T X$$

Polynomial



Rational



GCN Thomas N. Kipf et al. (2016)

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

GraphSAGE Will Hamilton et al. (2017)

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

GIN Xukeyu Lu et al. (2019)

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

DeepWalk Bryan Perozzi et al. (2014)

$$Z = \frac{1}{t+1} (I + (I - \tilde{L}) + (I - \tilde{L})^2 + \dots + (I - \tilde{L})^t) X = U (\theta_0 + \theta_1 \Lambda + \theta_2 \Lambda^2 + \dots + \theta_t \Lambda^t) U^TX$$

ChebyNet Defferrard, Michael et al. (2016)

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

Node2Vec Aditya Grover et al. (2016)

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

Personalized PageRank Johannes Klicpera et al. (2018)

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

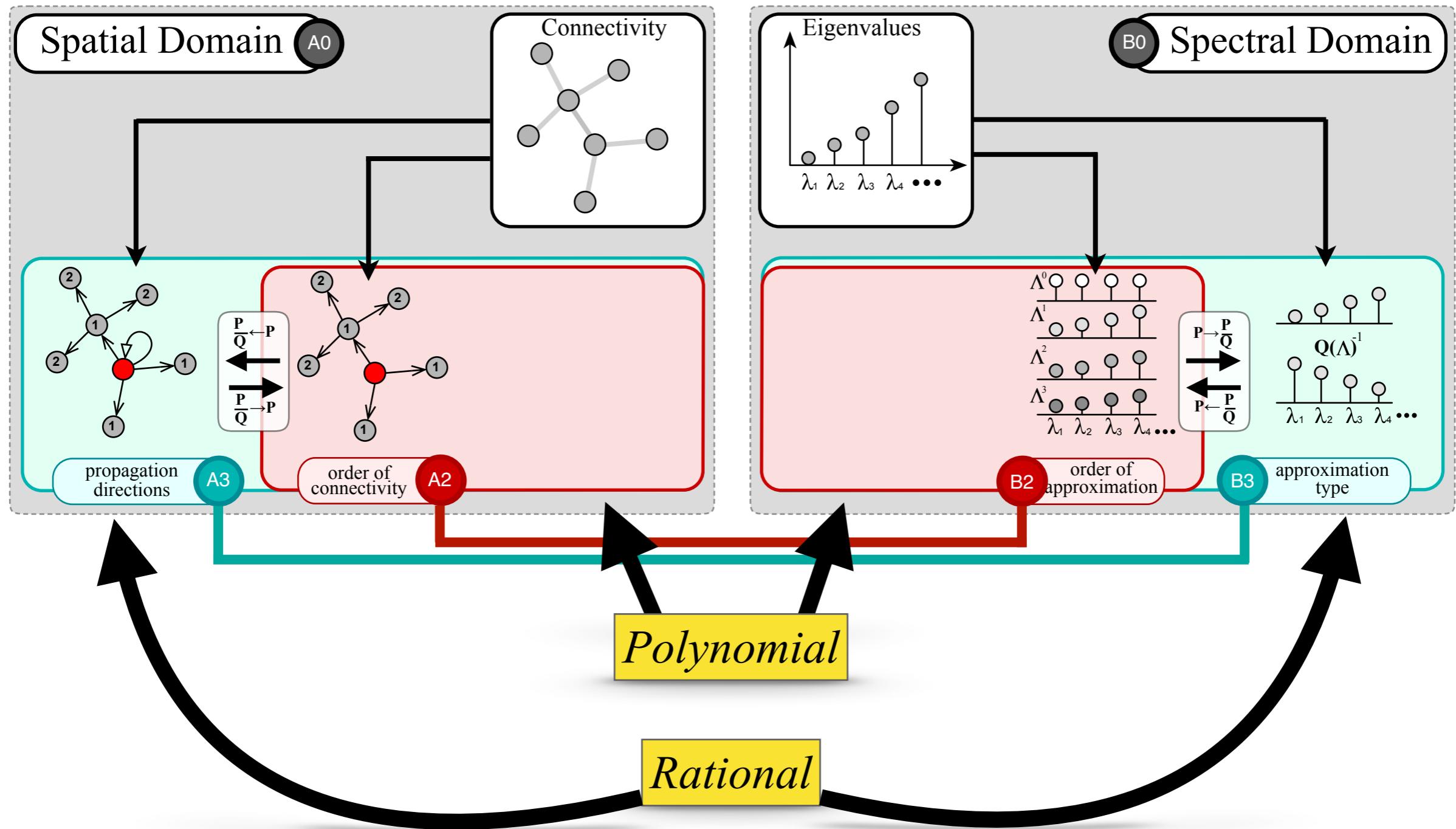
ARMA Filter Filippo Maria Bianchi et al. (2018)

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

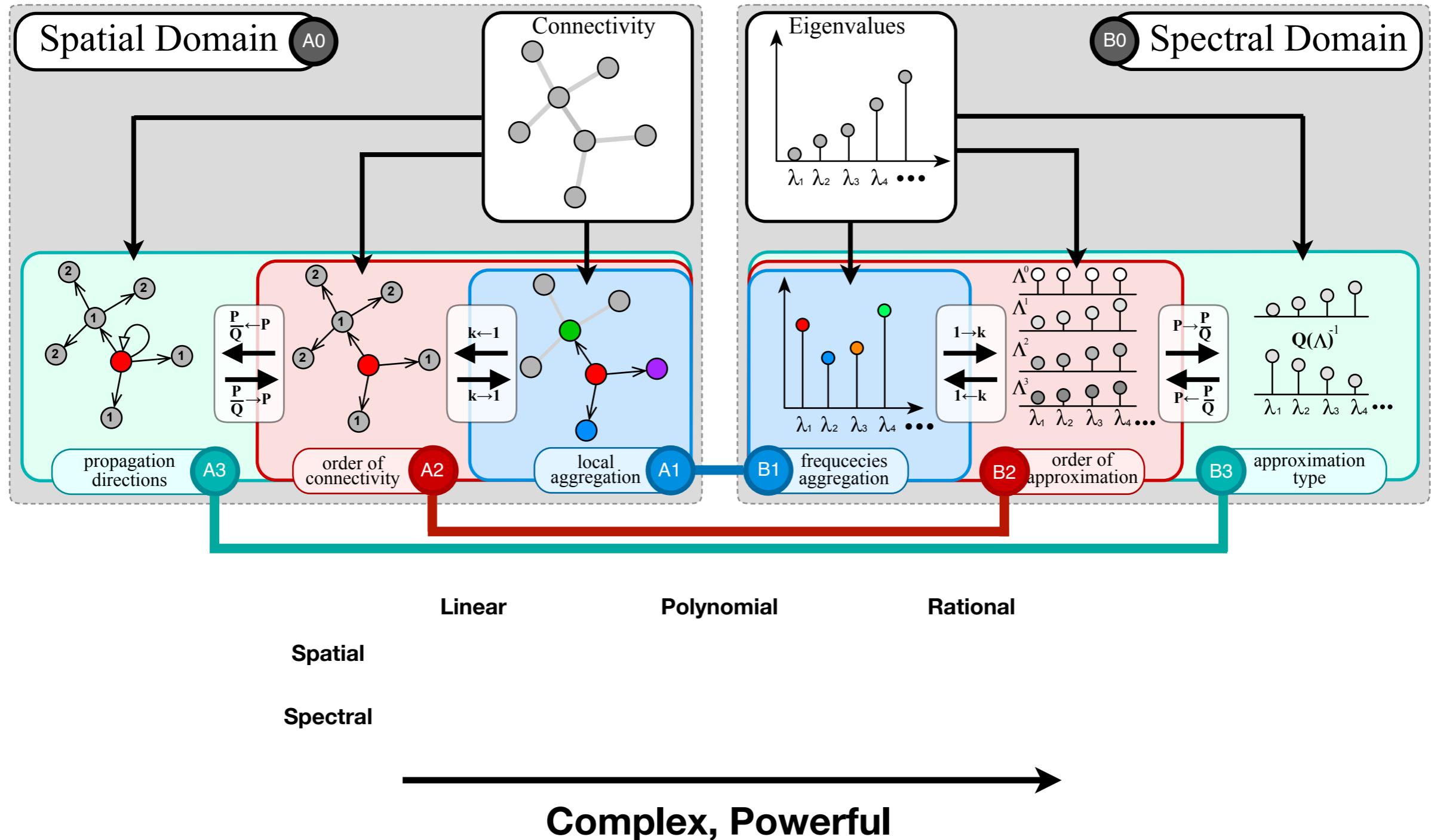
Auto Regressive Filter Qimai Li et al. (2019)

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

Rational v.s. Polynomial



The Unified Framework

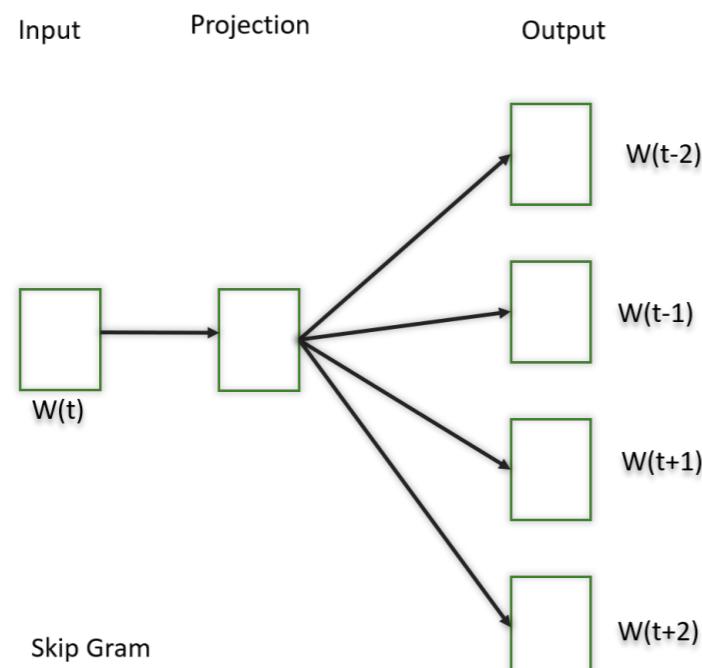


Spatial v.s. Spectral

Spatial, local \longleftrightarrow Spectral, global

Word2Vec

Tomas Mikolov 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

Spatial v.s. Spectral

Spatial, local



Spectral, global

SpectralNet

Uri Shaham et al. (2018)

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

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

Spectral Clustering

$$\mathbf{A} = \begin{bmatrix} & & \\ \vdash & \vdash & \vdash \\ & & \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}^{-1}$$

Eigen vectors of \mathbf{A} Eigen values of \mathbf{A} Eigen vectors of \mathbf{A}

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

Spatial v.s. Spectral

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

Agenda



- First Half (40 min)
 - *Preliminary: graph convolutions (25 min)*
 - *Analysis: expressive power (15 min)*
- Second Half (50 min)
 - *Introduction: a new unified framework (45 min)*
 - ***Future directions (5 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 $t \rightarrow +\infty$?	Energy is constant so does not decay	Decays to zero (if ϕ integrable)
(vii) Information	Transported	Lost gradually

Rational

Polynomial

Future Directions

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

Future Directions

○ Hodge Laplacian

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

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

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

L₁ is normal graph Laplacian

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

Function of graph matrix

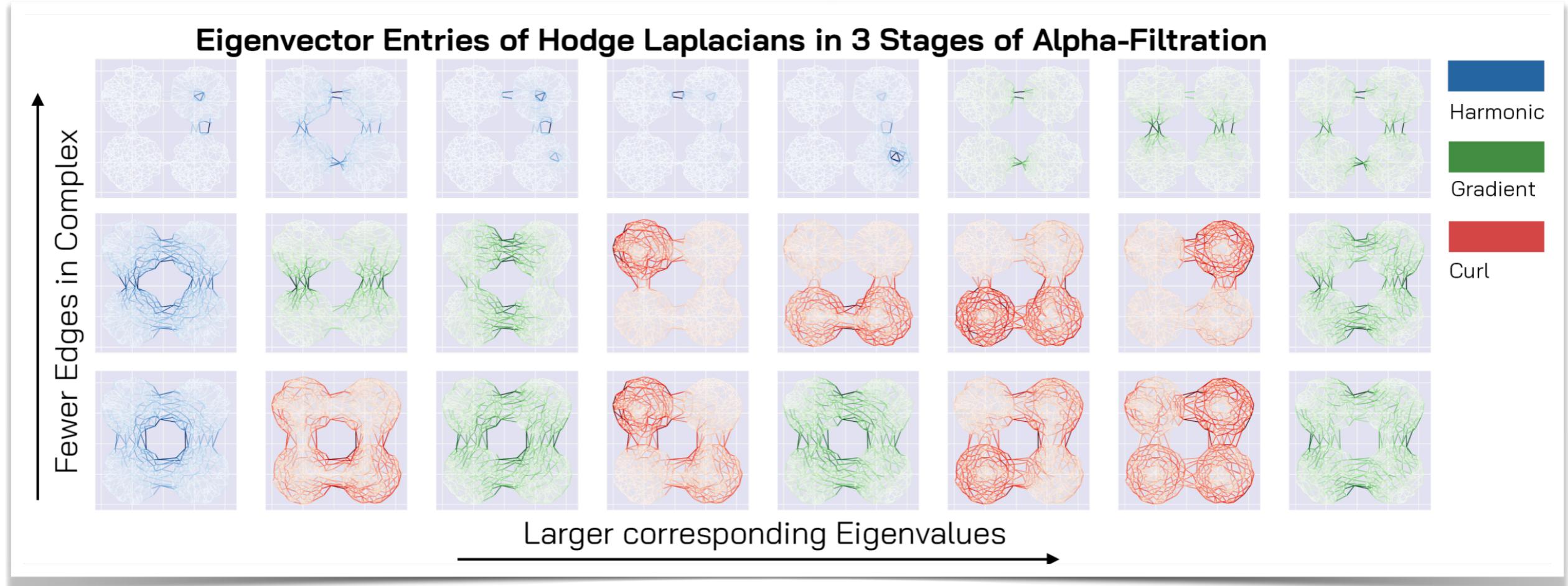
$g(A)X$

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

Future Directions

- Hodge decomposition

- Decompose dynamics into 3 categories



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

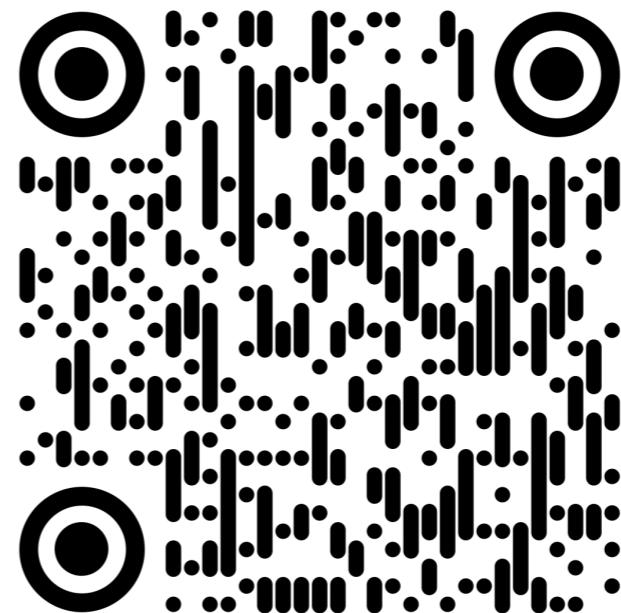
Thank You & Q/A

Awesome Spectral Graph Neural Networks

PRs Welcome  awesome

Contents

- [Survey Papers](#)
- [Milestone Papers](#)
- [Spatial and Spectral Views](#)
- [Twin Papers](#)
- [Applications](#)
- [Code](#)
- [Citation](#)



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