CVPR 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

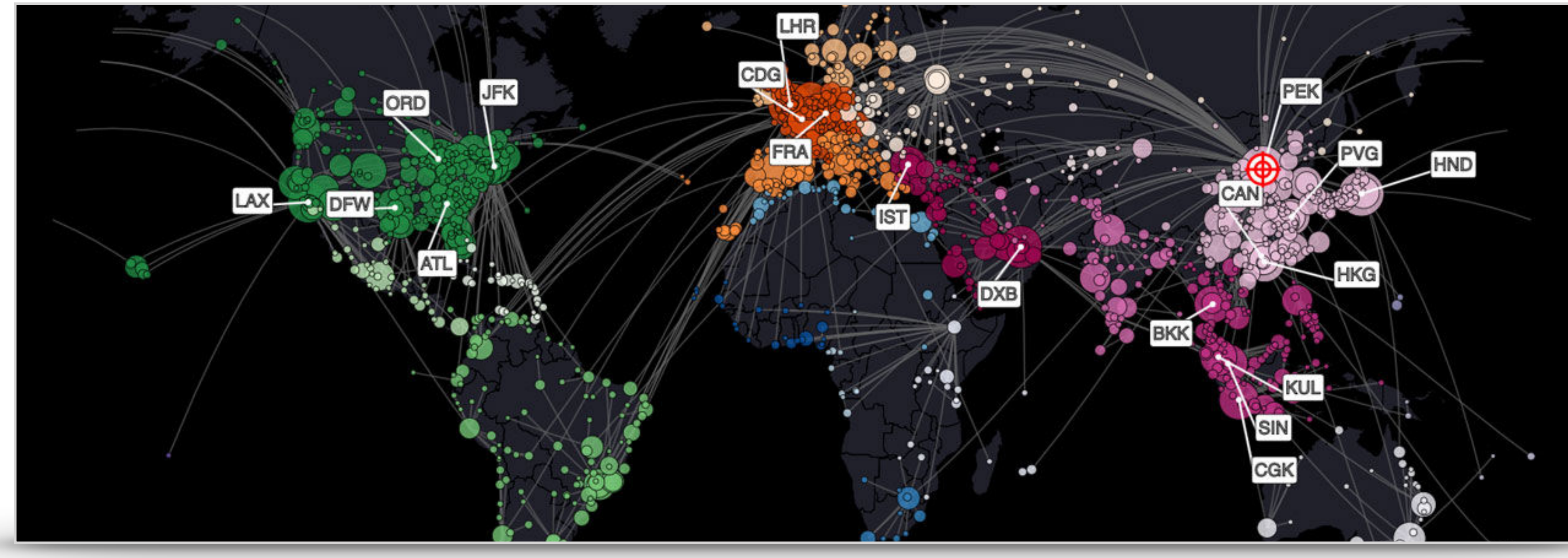
Agenda

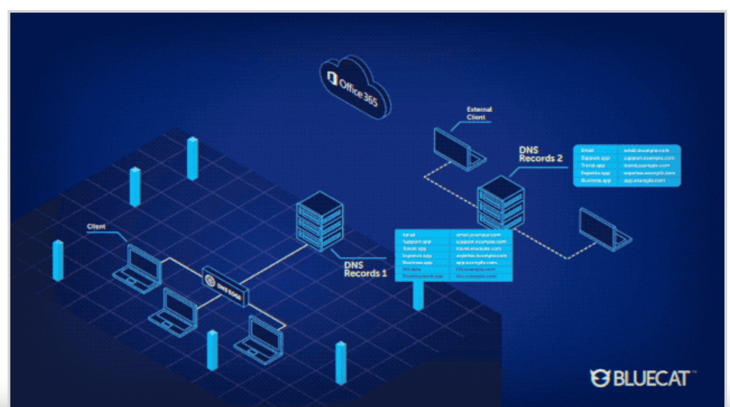
- **First Half (1 hour 15 min)**
  - *Background*: **unified frameworks for GNN** (35 min)
  - *Preliminary*: **graph convolutions** (40 min)
  - BREAK (15min)
- **Second Half (1 hour)**
  - *Introduction*: **a new unified framework** (40 min)
  - *Future directions* (20min)
  - Q&A (15min)

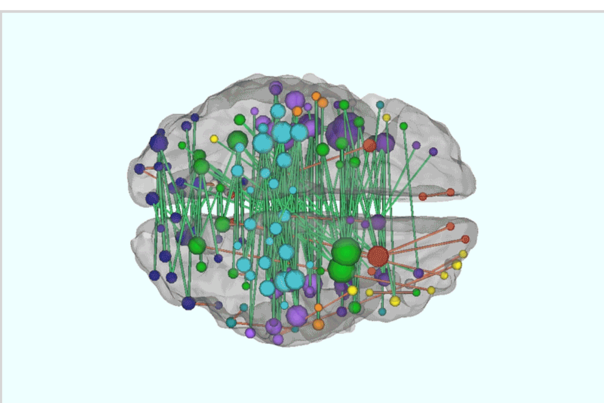
Agenda

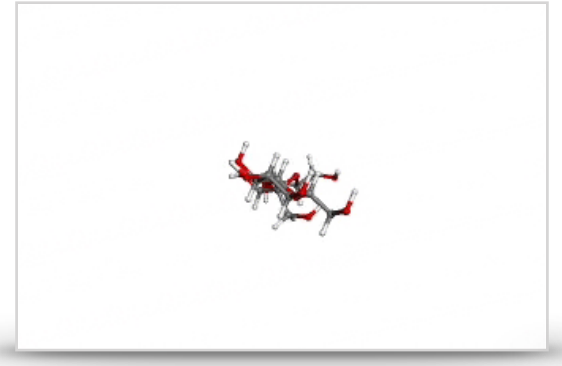
- **First Half (1 hour 15 min)**
  - *Background*: **unified frameworks for GNN** (35 min)
  - *Preliminary*: **graph convolutions** (40 min)
- **BREAK (15min)**
- **Second Half (1 hour)**
  - *Introduction*: **a new unified framework** (40 min)
  - *Future directions* (20min)
- **Q&A (15min)**

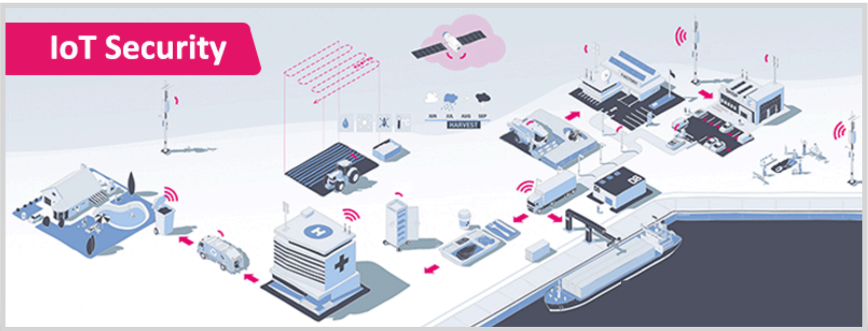
Graph is Pervasive

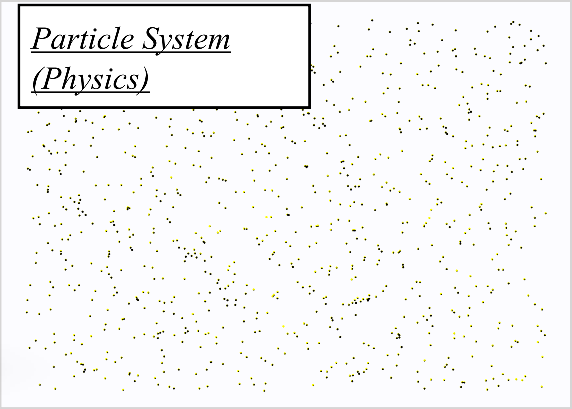




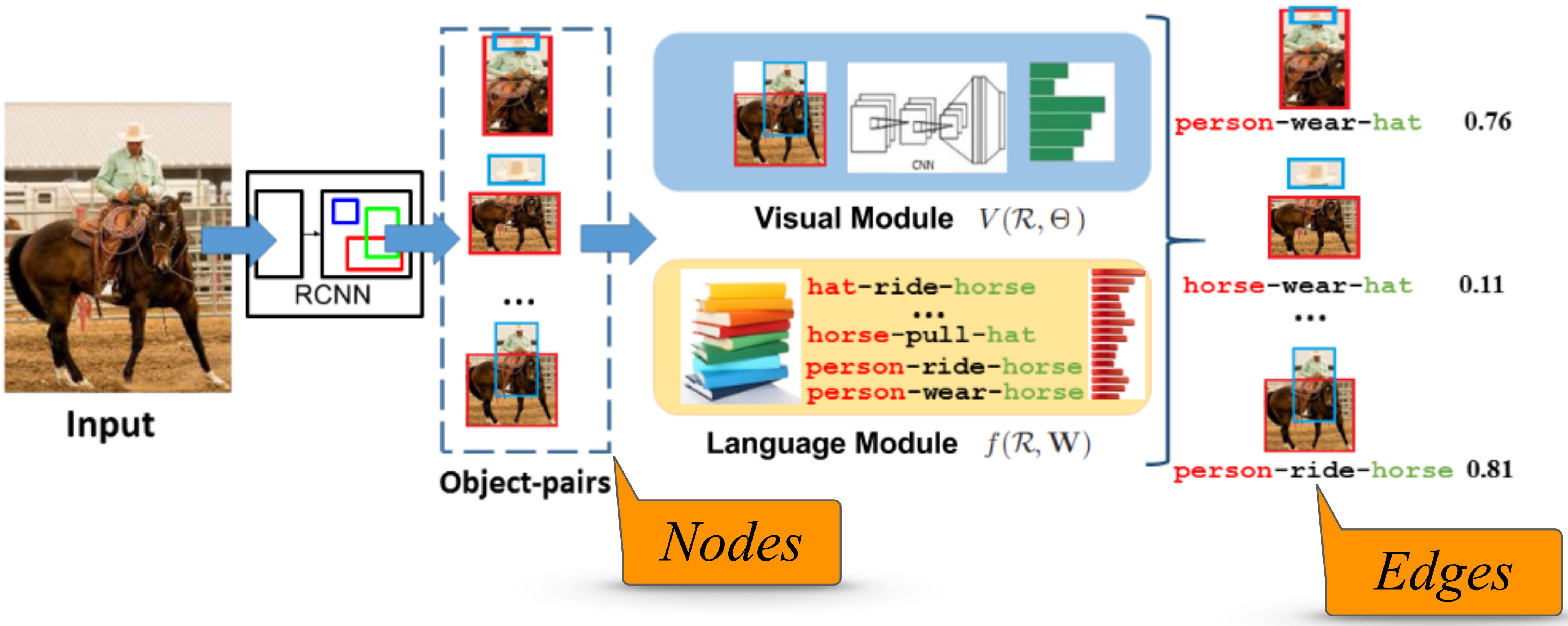








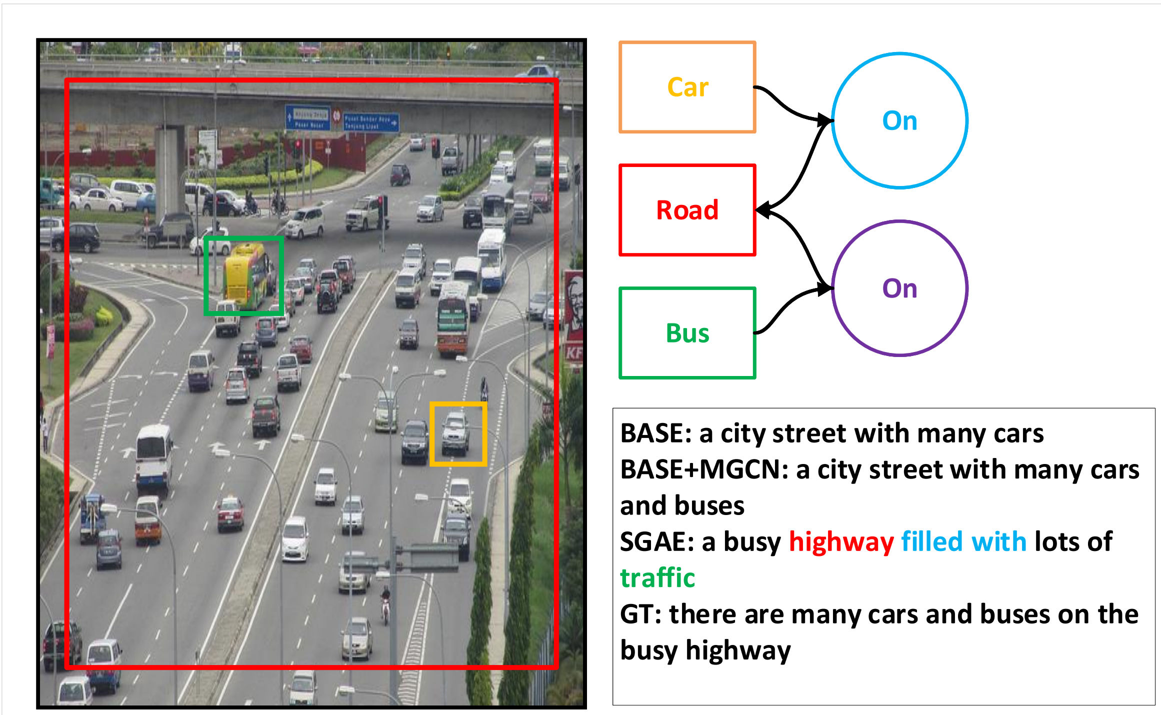
Graphs in CV



Graphs in Image Data:
Scene Graph

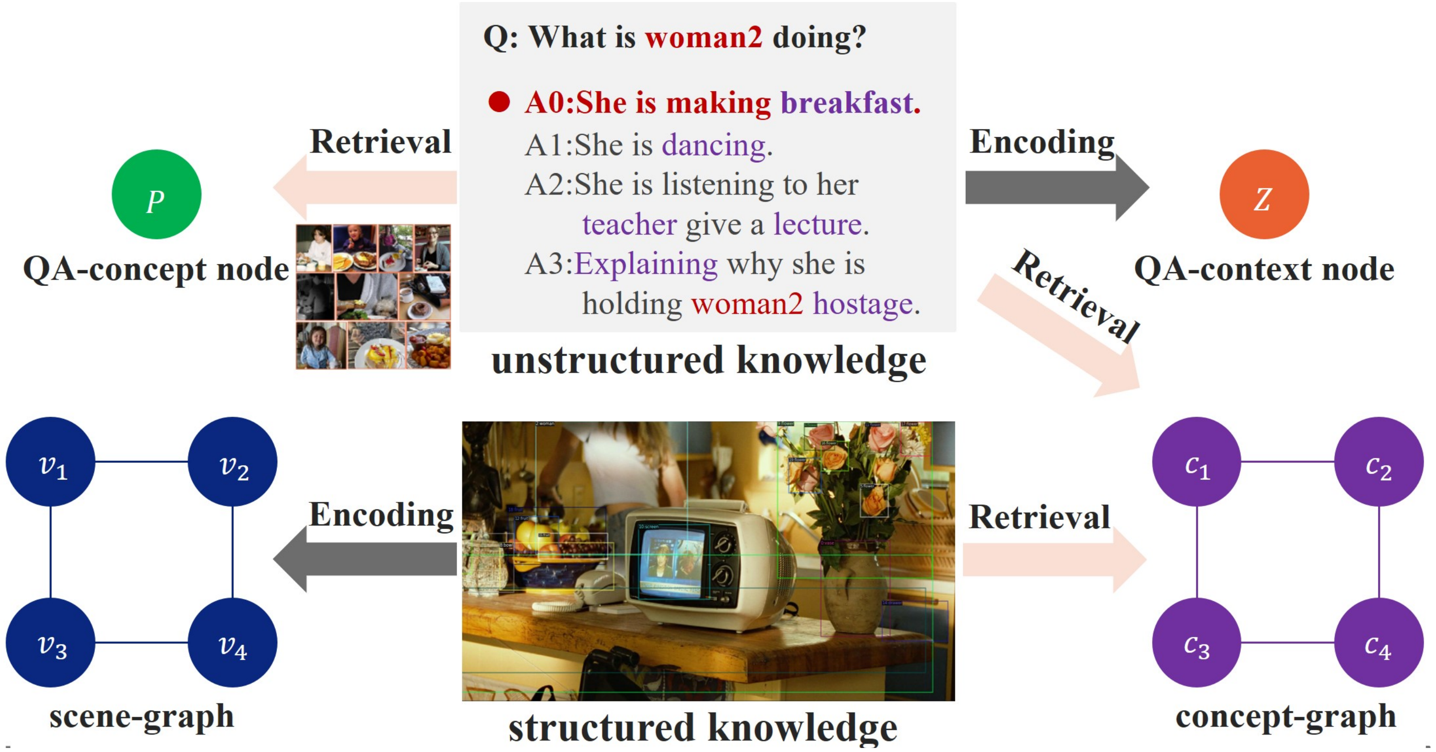
Lu, Cewu, et al. "**Visual relationship detection with language priors.**" ECCV, 2016.

Graphs in CV



Image/Video Captioning

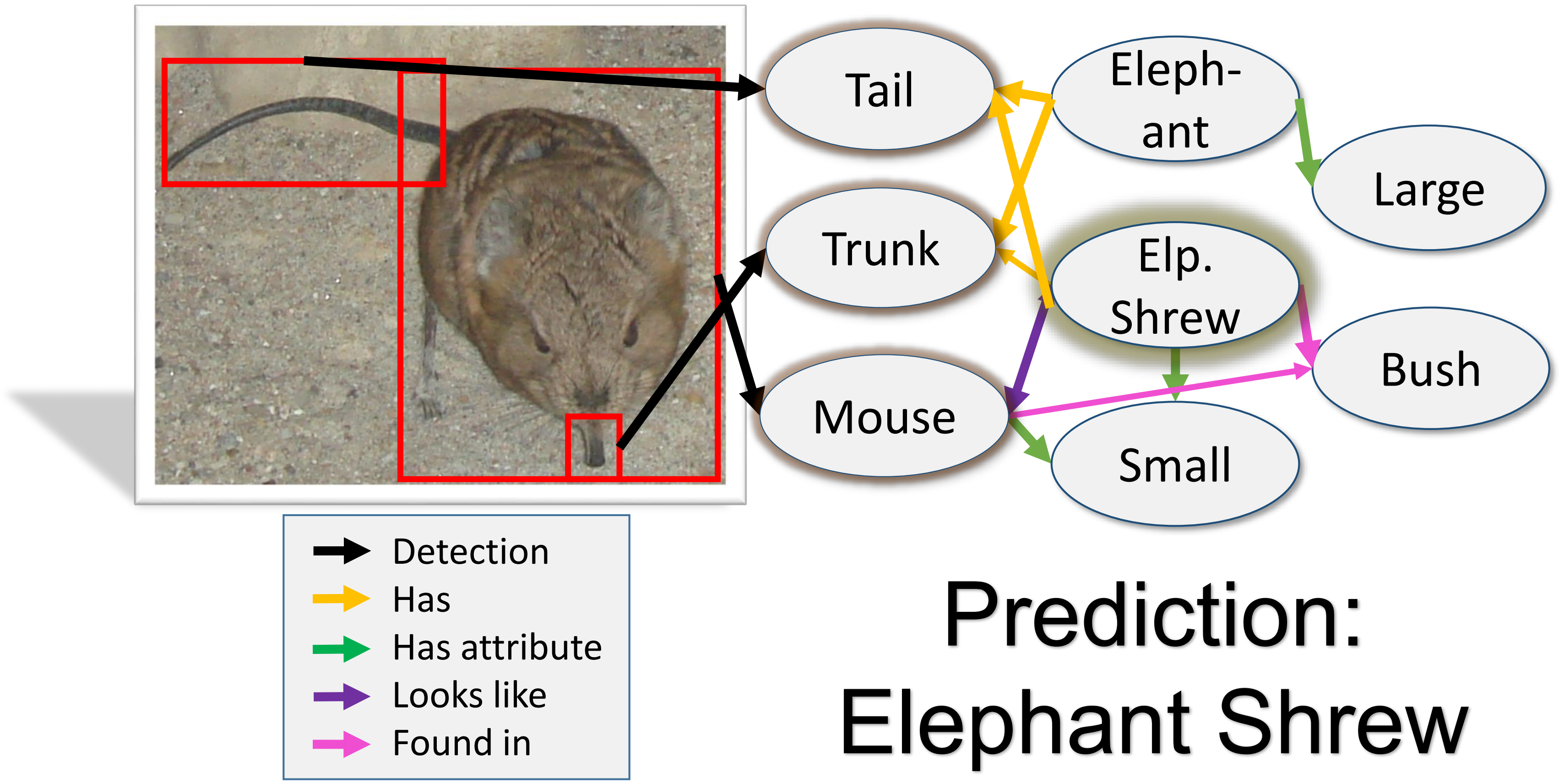
Yang, Xu, et al. "Auto-encoding scene graphs for image captioning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.



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

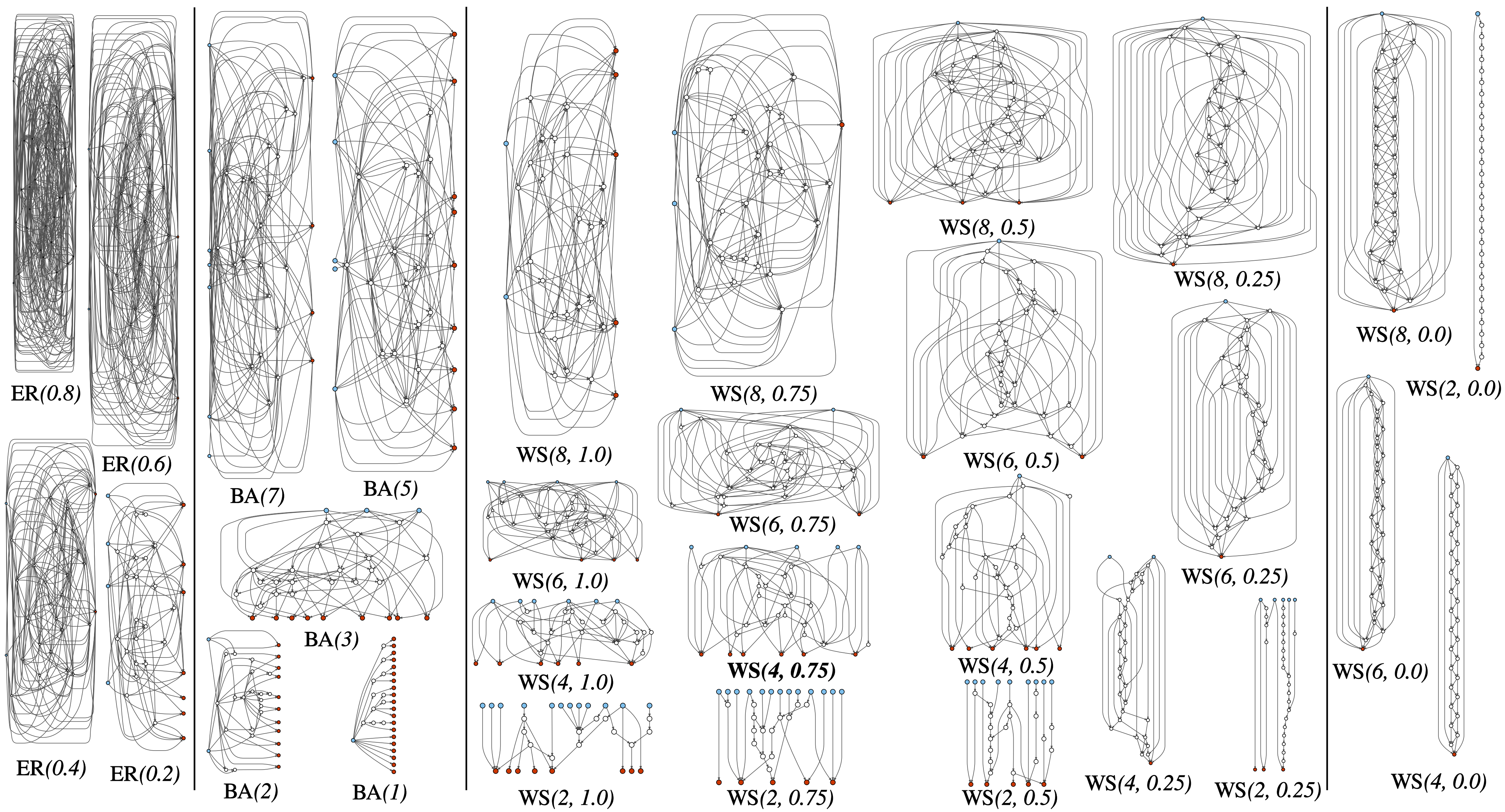


Prediction:
Elephant Shrew

Graph as Auxiliary:
Knowledge graph for image classification

Marino, Kenneth et al. "**The More You Know: Using Knowledge Graphs for Image Classification.**" CVPR 2016

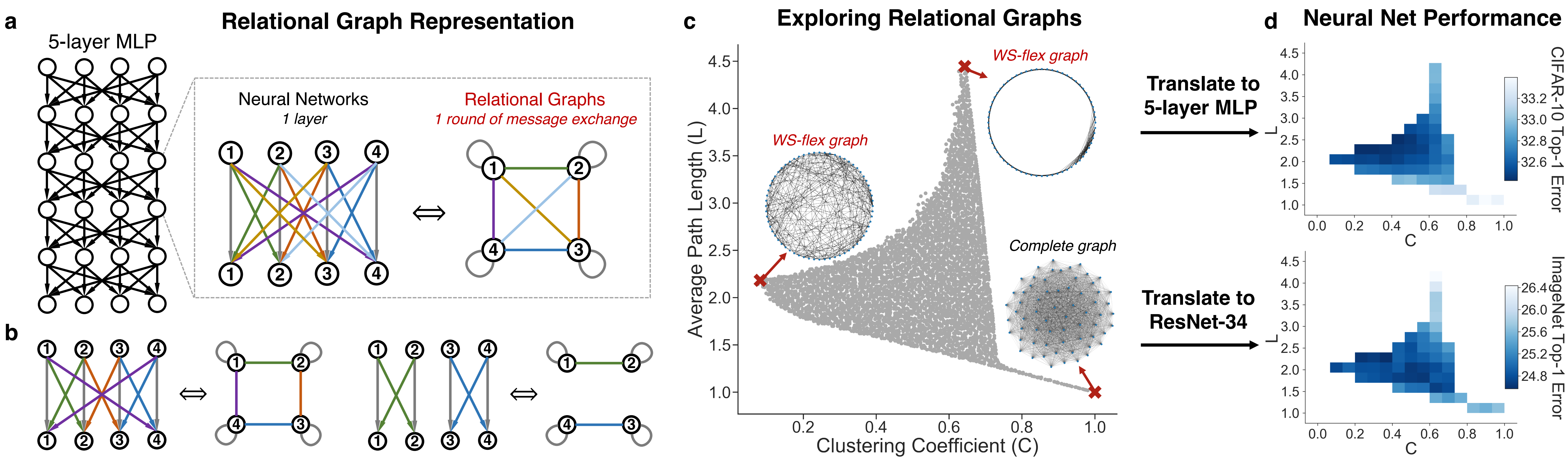
Graphs in CV



Graphs in neural network architecture

Xie, Saining, et al. ***"Exploring randomly wired neural networks for image recognition."*** *ICCV 2019.*

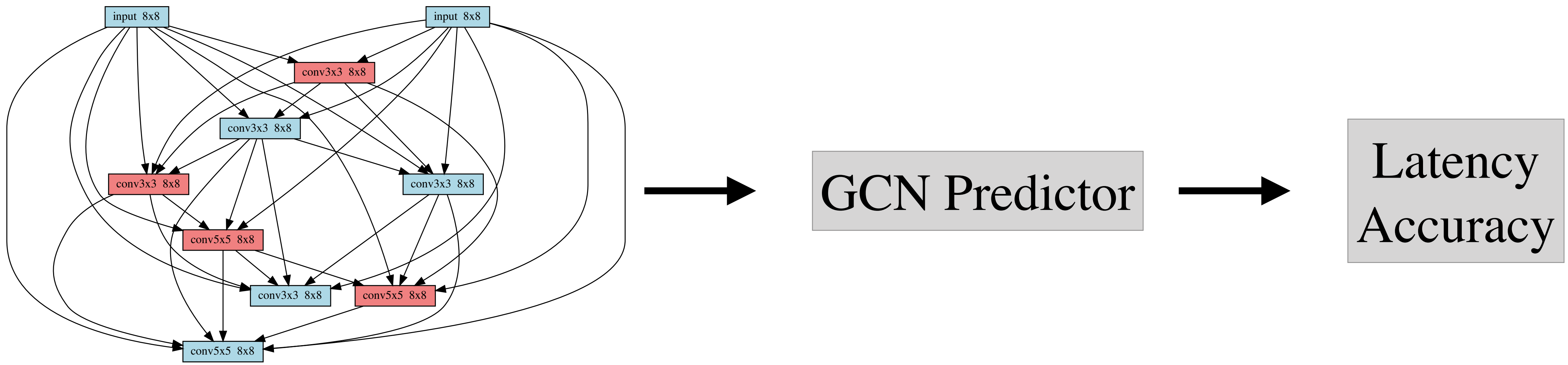
Graphs in CV



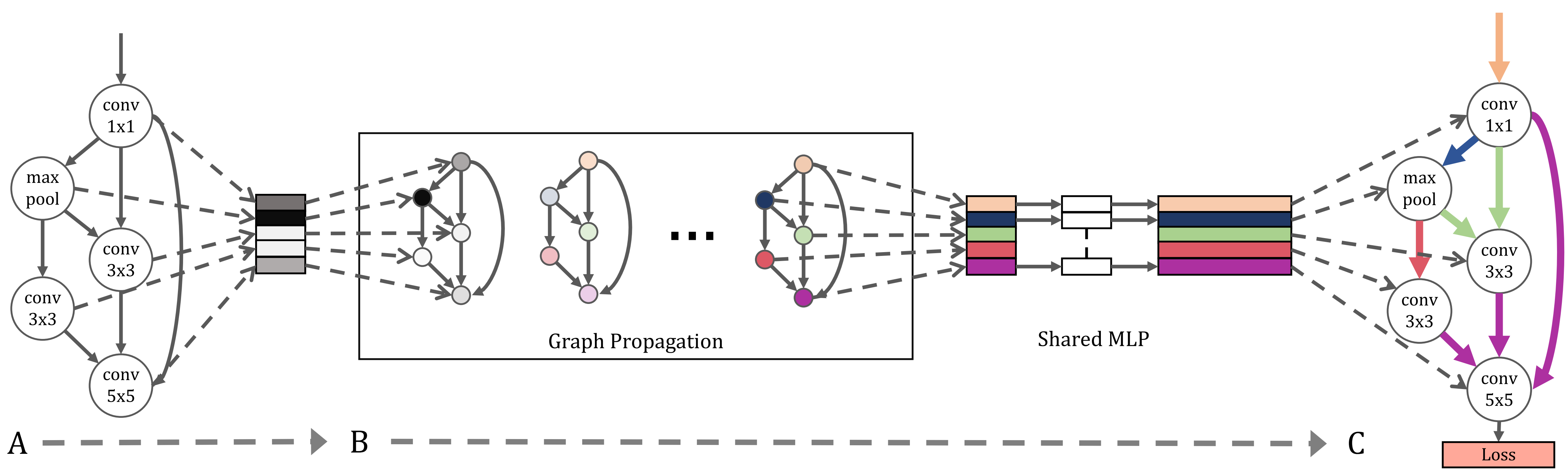
Graph topology and neural network architectures

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

Graphs in CV



*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*

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

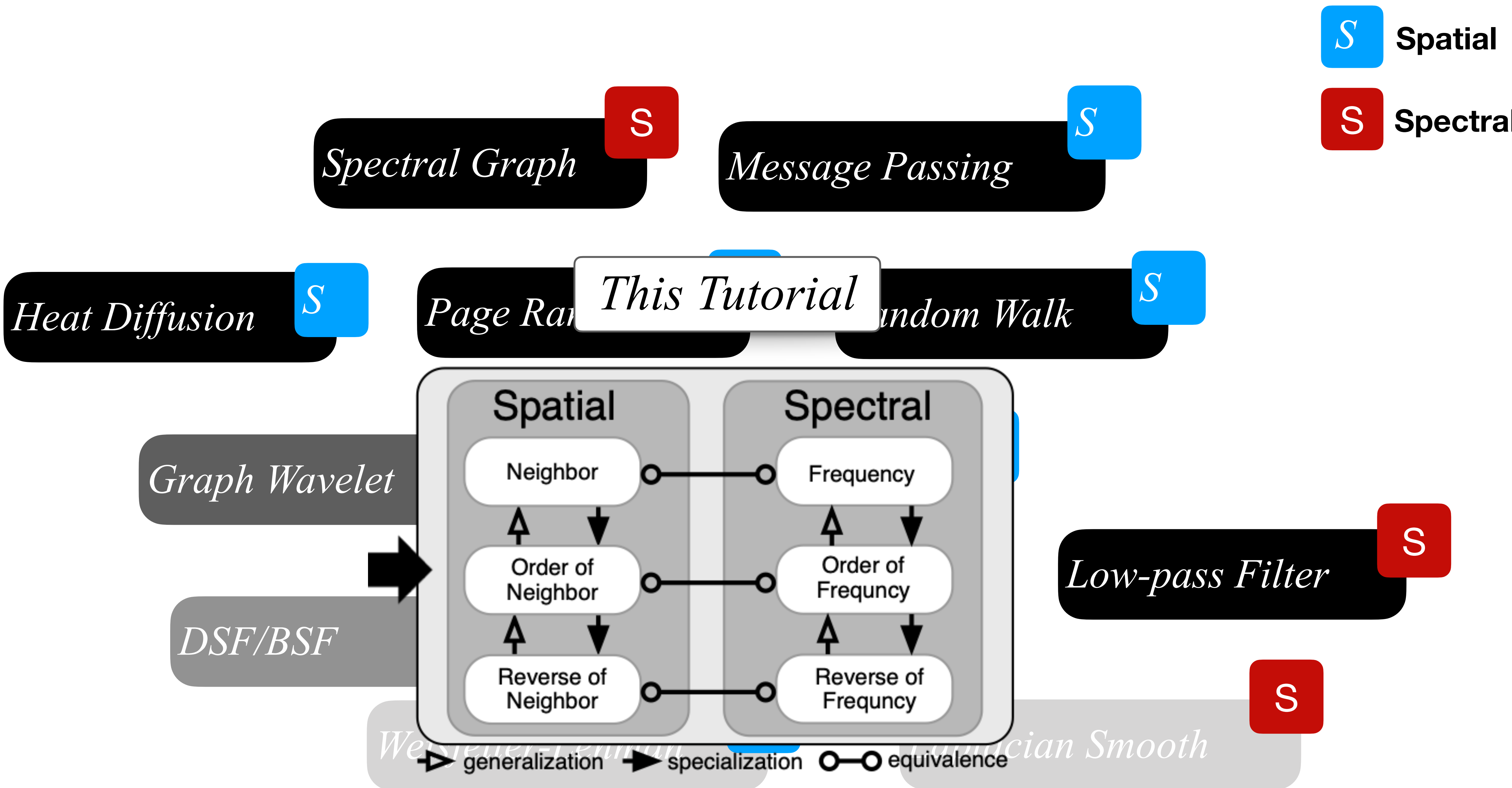
Weisfeiler-Lehman S

Laplacian Smooth S

…

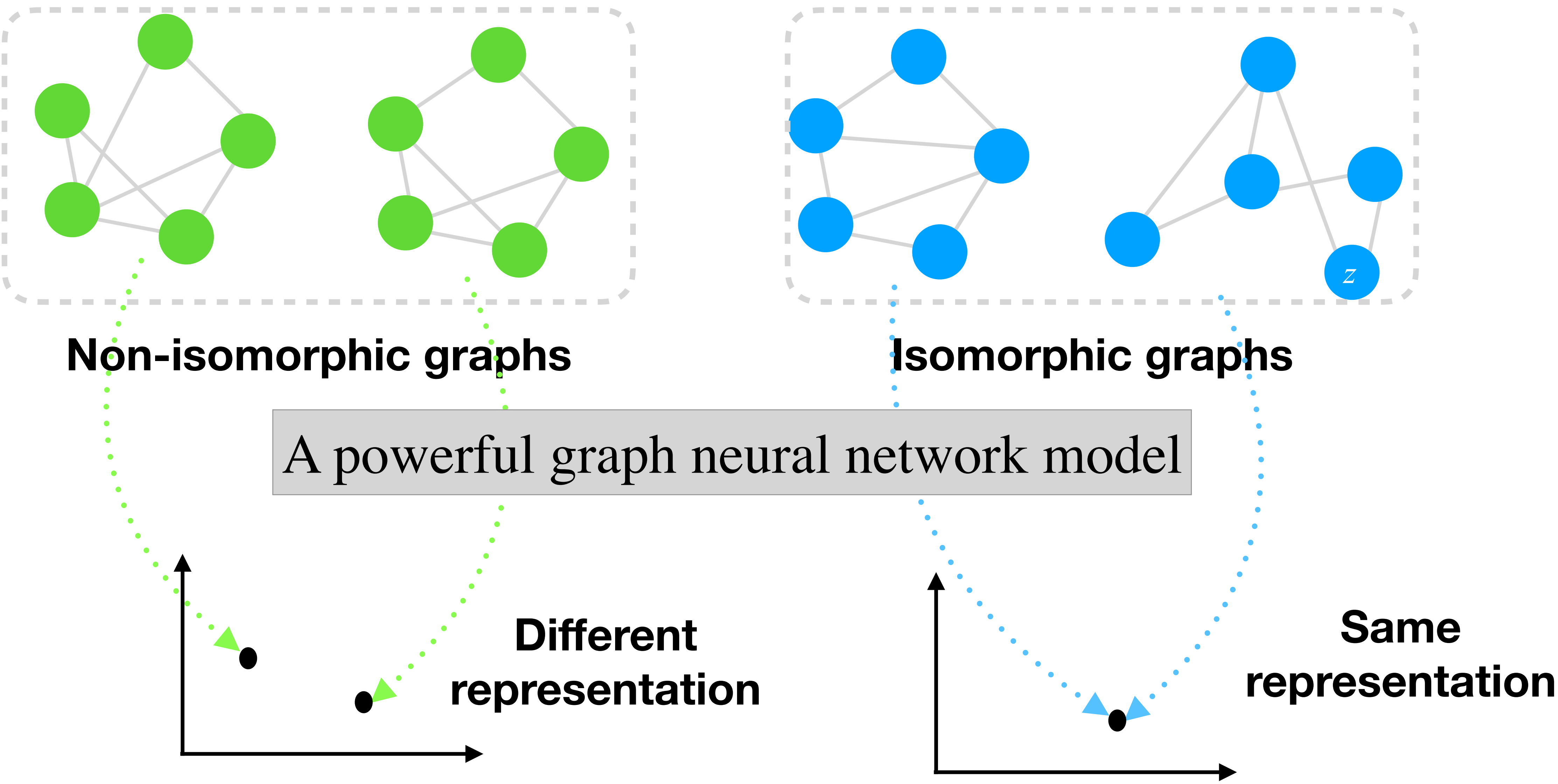
Motivation: A Unified View

A large number of graph neural networks, with different mechanisms



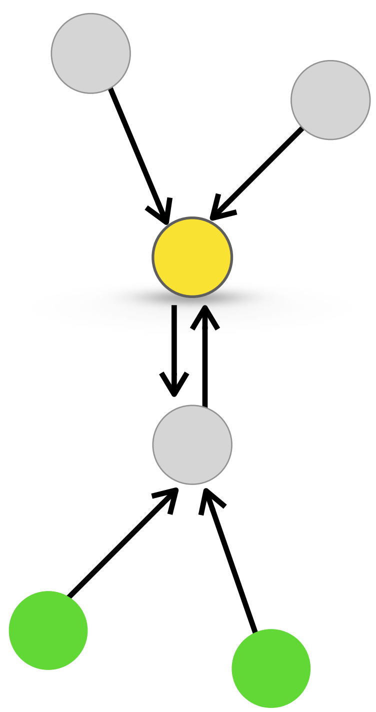
...

Attempts to Unify GNNs



Compare GNNs with respect to their **<u>expressive power</u>** (ability to distinguish different graph structures)

WL-Test: on 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.*

WL-Test: on Spatial GNNs

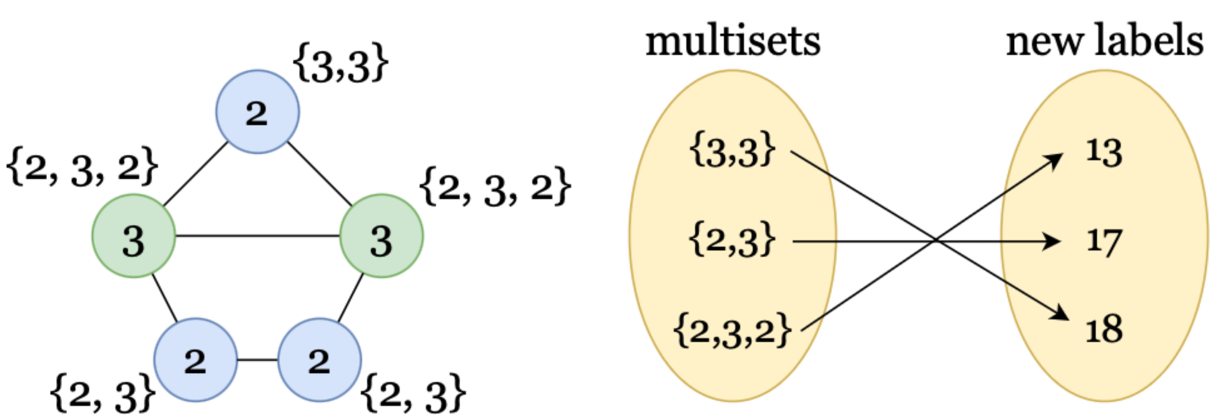
*GNNs are at most as powerful as a **Weisfeiler-Lehman** graph isomorphism test.*

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

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

*This upper bound is achieved if **AGGREGATE** and **READOUT** are **Injective Multiset Functions***

*Example GNNs that are **LESS** powerful than WL test: GCN, GraphSage*



Every possible output has at most one associated input

$$h_v^{(k)} = \text{ReLU}\left(W \cdot \text{MEAN}\left\{h_u^{(k-1)}, \forall u \in \mathcal{N}(v) \cup \{v\}\right\}\right).$$

GCN has mean AGGREGATE, so it is not an injective function. As a result, it is less powerful. *

Spectral GNNs and Isomorphism Test

Spatial GNNs

AGGREGATE + READOUT ⟷ Isomorphism Test

What is the connection?

Spectral GNNs

Spectral Filters ⟷ Isomorphism Test

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

Wang, X et al.,. ***How powerful are spectral graph neural networks.*** *International Conference on Machine Learning. PMLR, 2022.*

Spectral GNNs and Isomorphism Test

Spatial GNNs

AGGREGATE + READOUT ⟷ Isomorphism Test

*Doesn't work on GNNs with **Continuous** features.*
WL-test (1-WL test) is not perfect.

Spectral GNNs

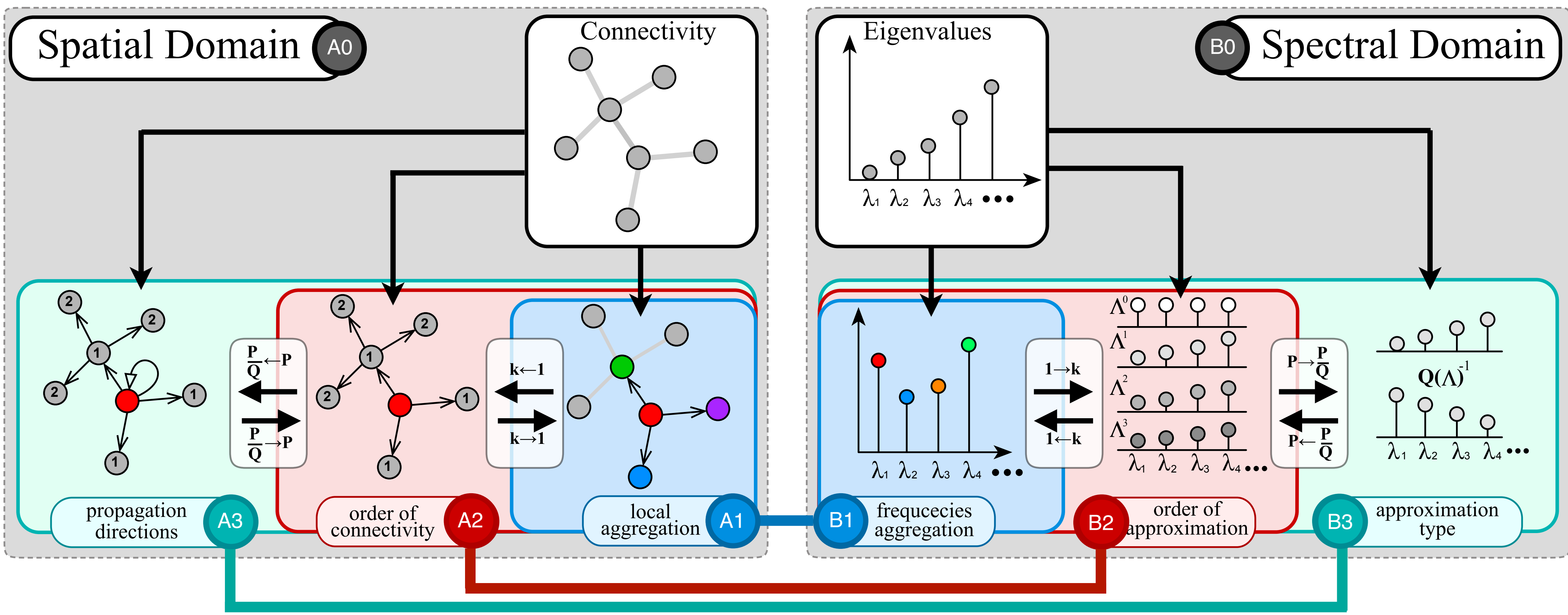
Spectral Filters ⟷ Isomorphism Test

Spectral filters are universal approximators when satisfying two conditions.
*The analysis is **limited to Spectral** GNNs without nonlinearity.*

Xu, K., et al., **How powerful are graph neural networks?** In International Conference on Learning Representations, 2019.
Wang, X et al.,. **How powerful are spectral graph neural networks.** International Conference on Machine Learning. PMLR, 2022.

Comparison

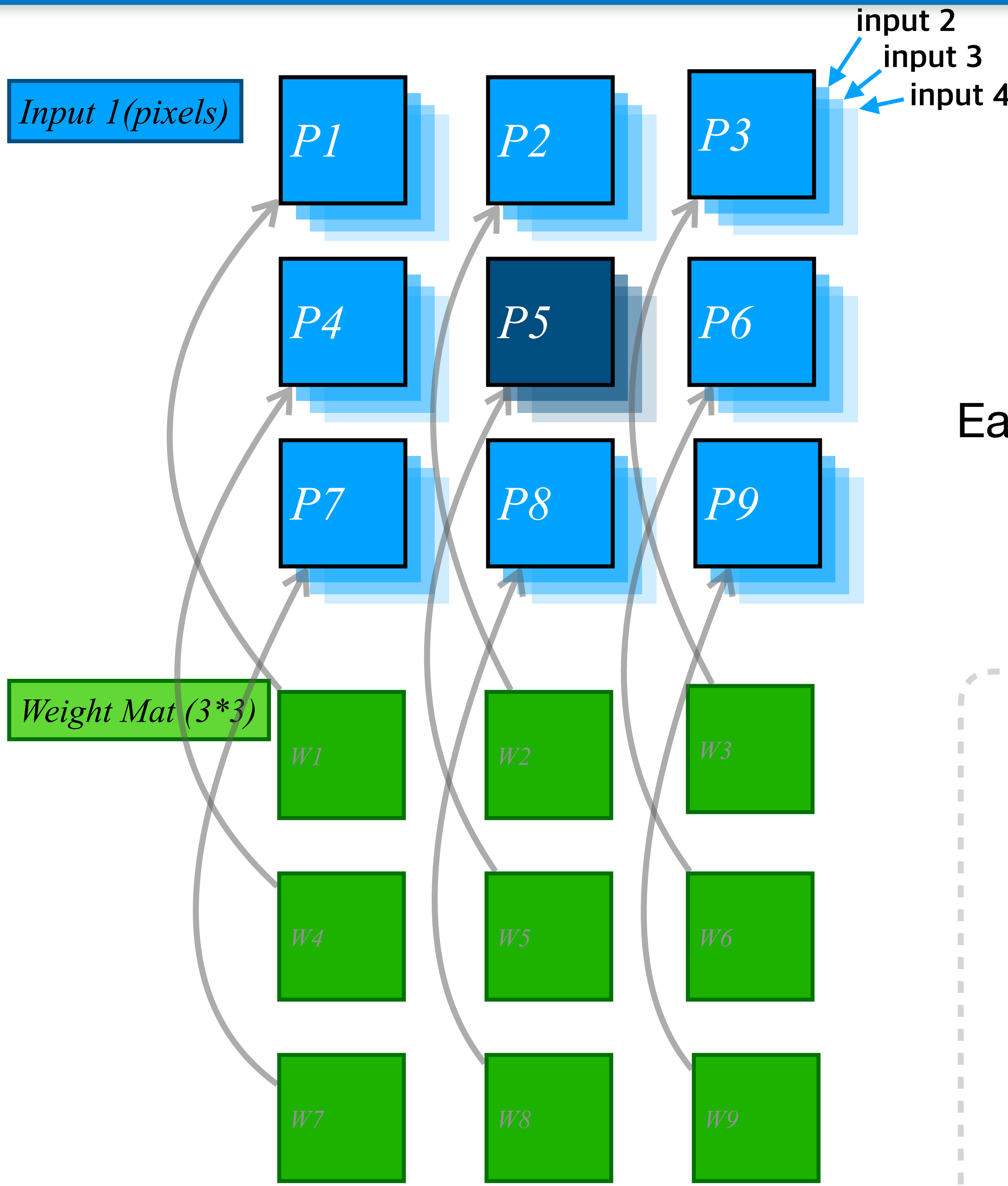
- Existing surveys and theoretical analyses focus on either the spatial or the spectral GNNs, not all of them.
- Here is the framework we will introduce later



Agenda

- **First Half (1 hour 15 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)

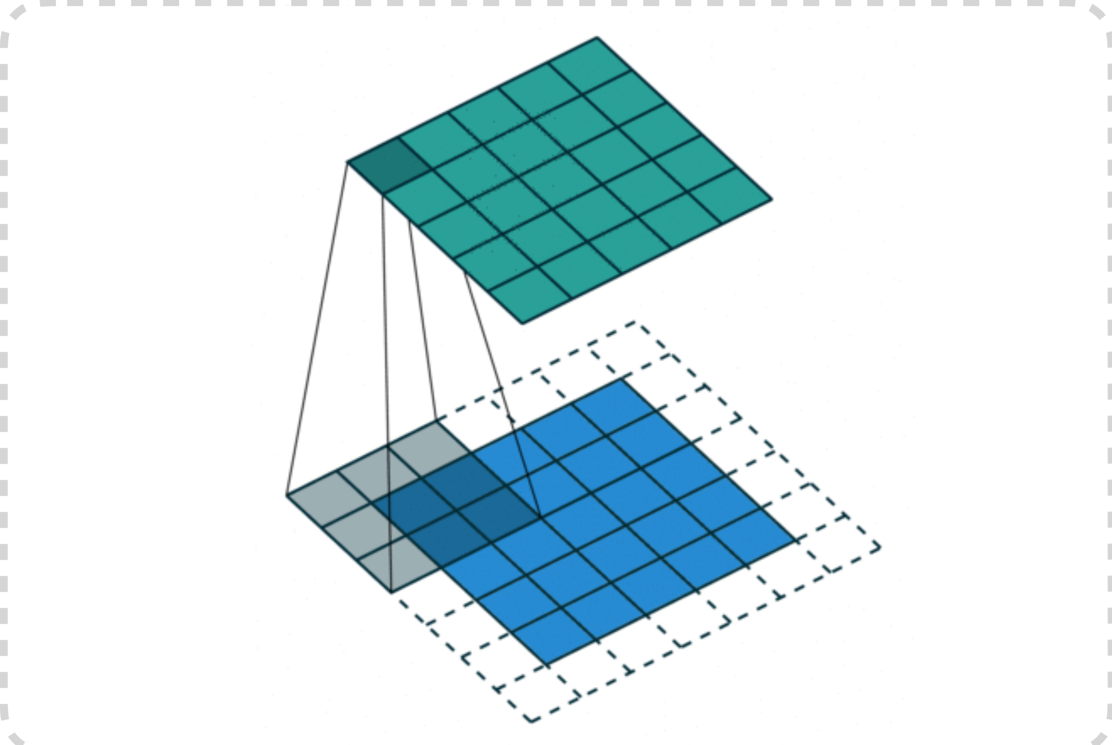
ConvNet



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

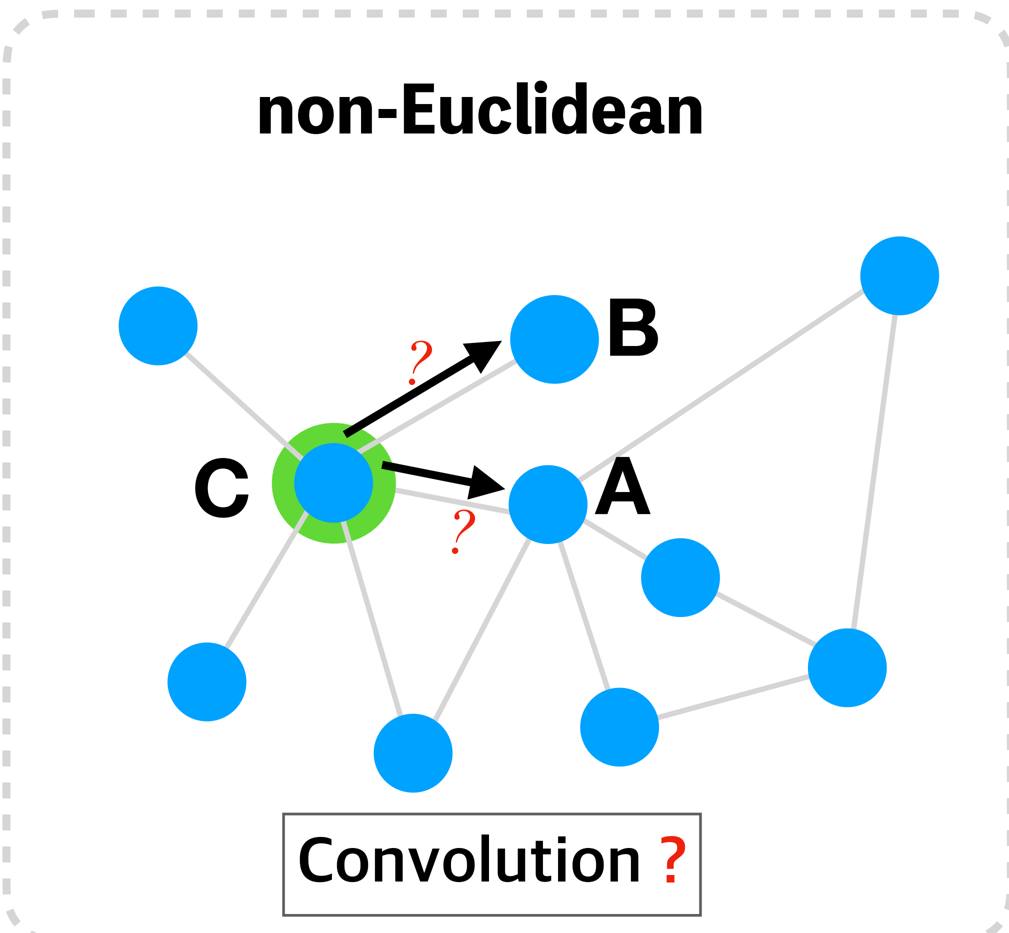
Each pixel has **fixed** number of neighbors

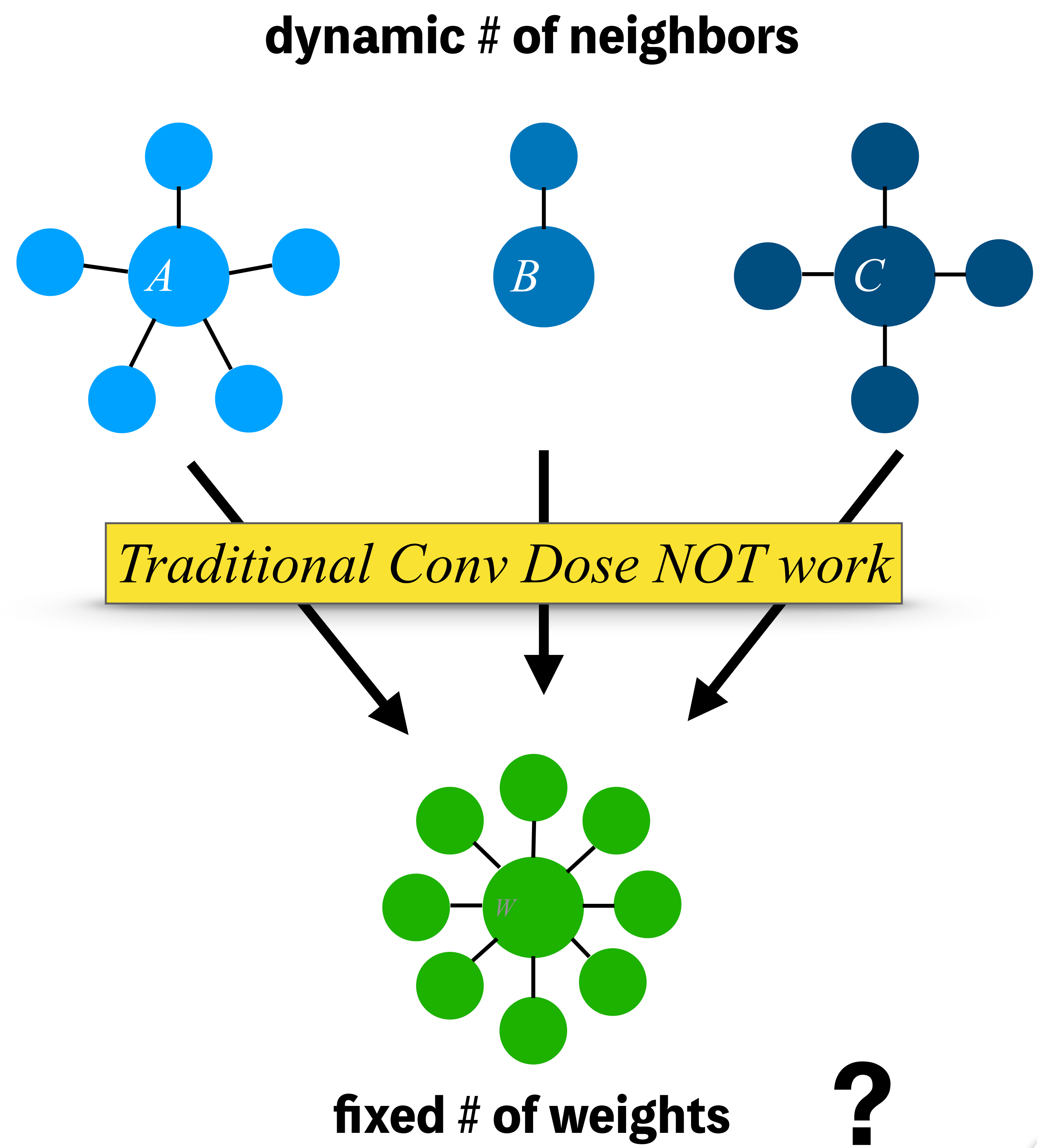
Convolution



Credit: Prof. Dr. Johannes Maucher

Challenge for ConvNet on Graphs





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.

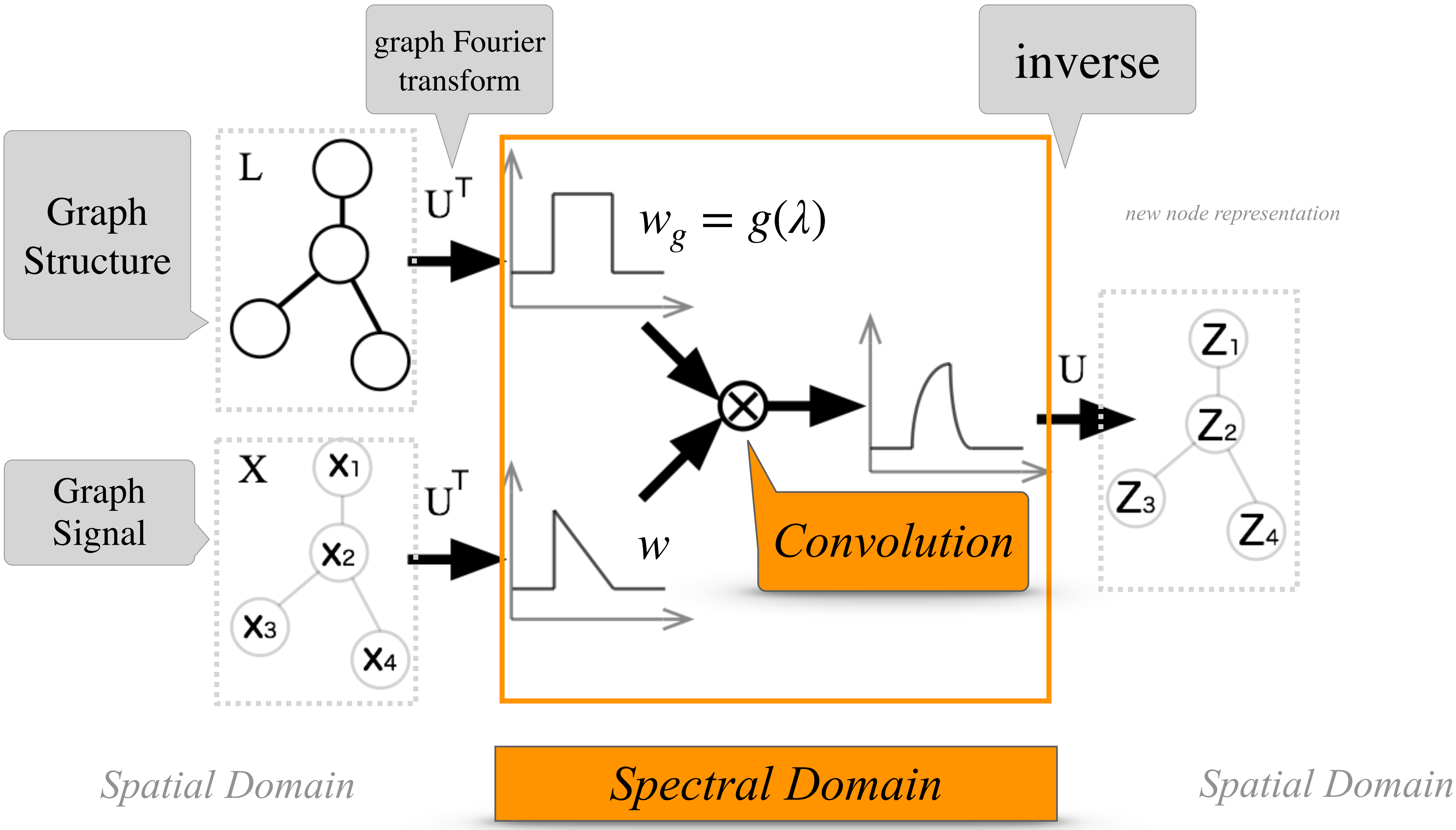
$$\mathscr{F}\{f * g\} = \mathscr{F}\{f\} \cdot \mathscr{F}\{g\}$$

Space convolution = frequency multiplication

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

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

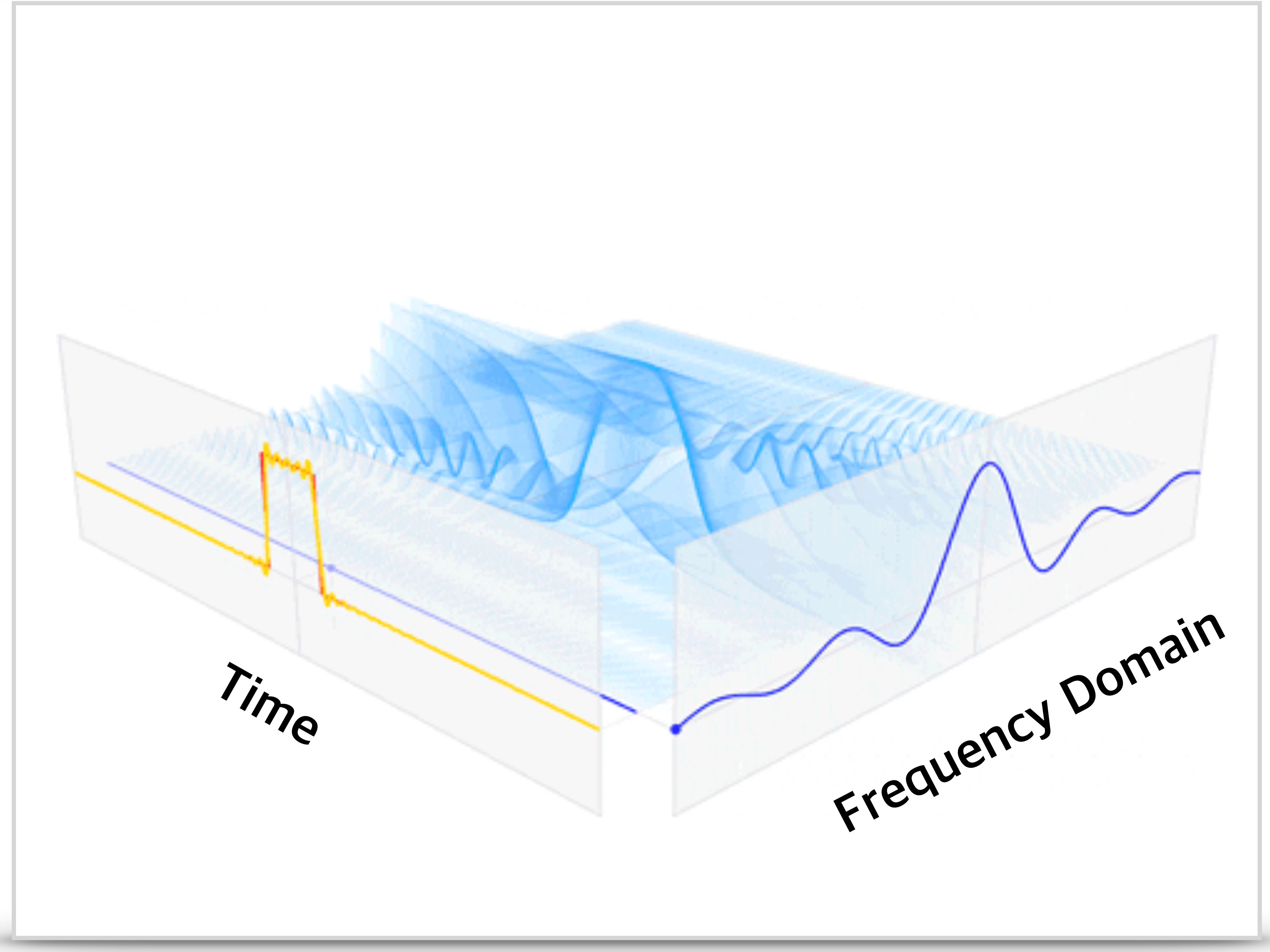
Convolution theorem: $f * g = \mathscr{F}^{-1}\{\mathscr{F}\{f\} \cdot \mathscr{F}\{g\}\}$

$\Downarrow$

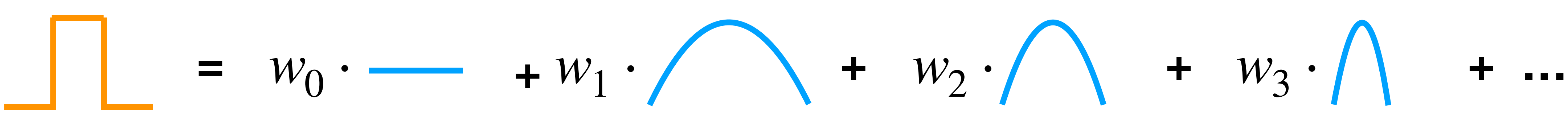
What is the Fourier Transform on Graphs?

What is the Inverse Fourier Transform on Graphs?

Spectral Analysis

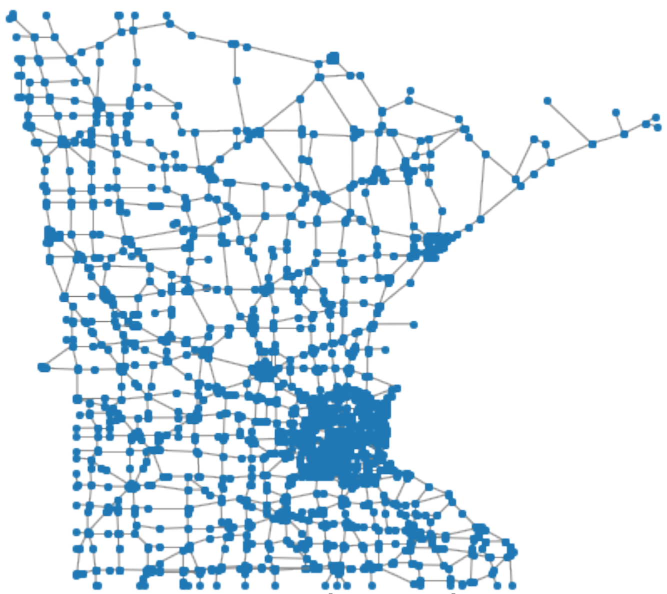


credit: giphy



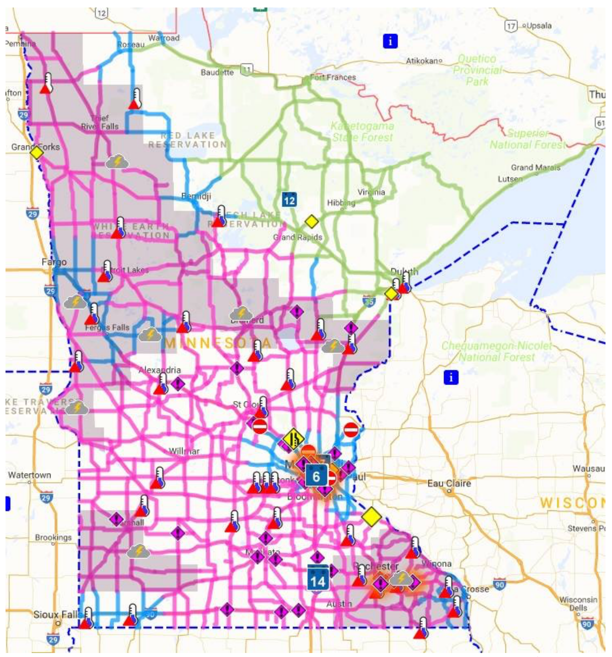
Spectral Analysis for Graph

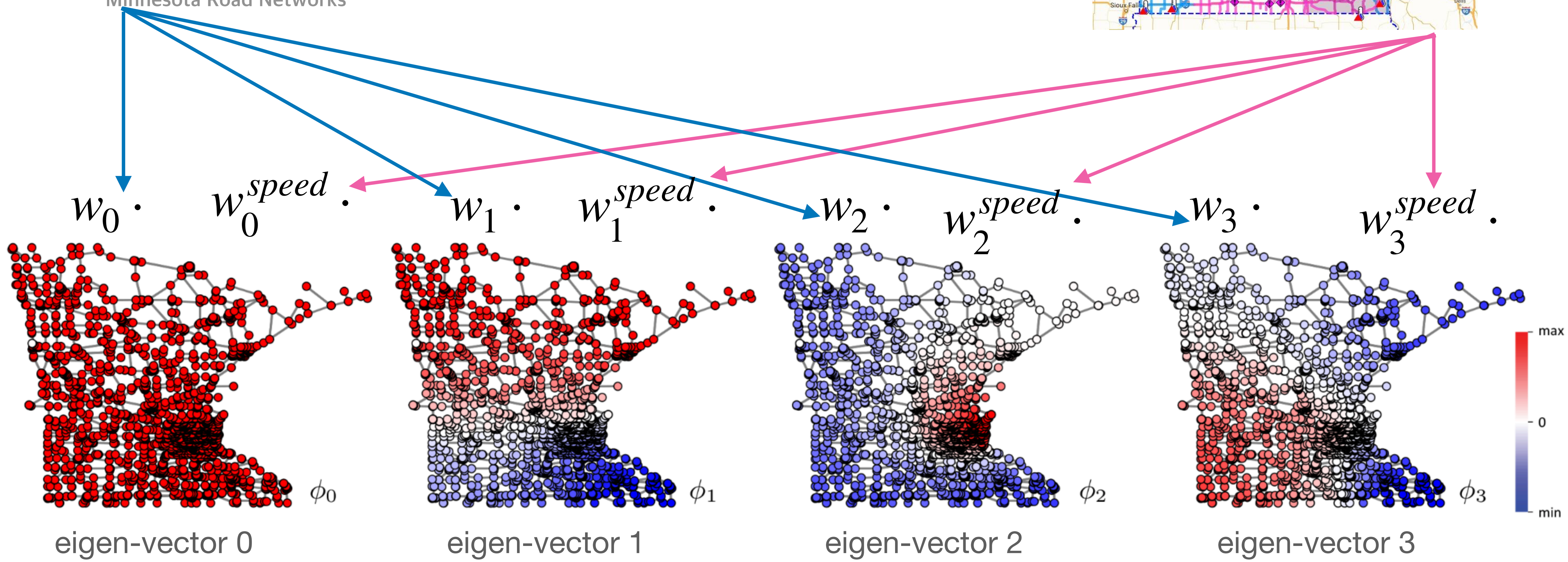
Graph Structure



Minnesota Road Networks

Graph Signal (e.g.,Traffic Speed)





Graph Fourier Transform (Spectral Decomposition)

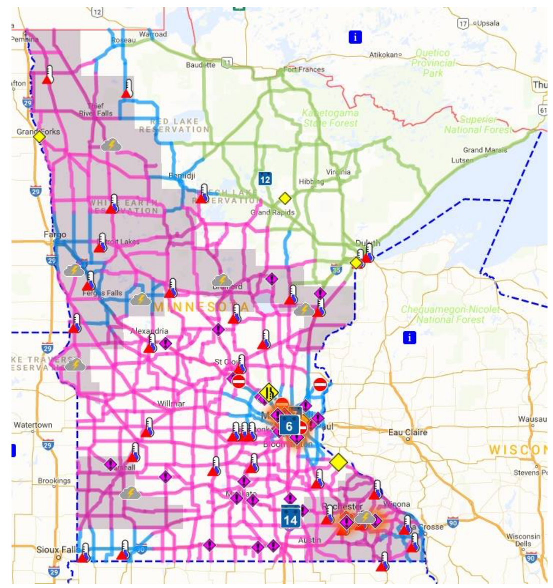
Convolution on Graph Data

- 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: Uf
- Now, what is U, and U^T ?
- $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$
- Finally, $g * X = U g(\Lambda) U^T X$

Convolution on Graph Data

Graph Structure

Graph Signal (e.g.,Traffic Speed)



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

1 graph Fourier transform

Get base by $L = U \Lambda U^\top$

L is graph Laplacian matrix

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

Convolution on Graph Data

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

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

Convolution on Graph Data

If constraint g_θ is polynomial

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

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

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

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

1-order polynomial without weights

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

A and L are normalized

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

$$= f + A \cdot f$$

Now we have GCN

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

Convolution on Graph Data

If constraint g_θ is polynomial

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

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

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

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

1-order polynomial without weights

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

A and L are normalized

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

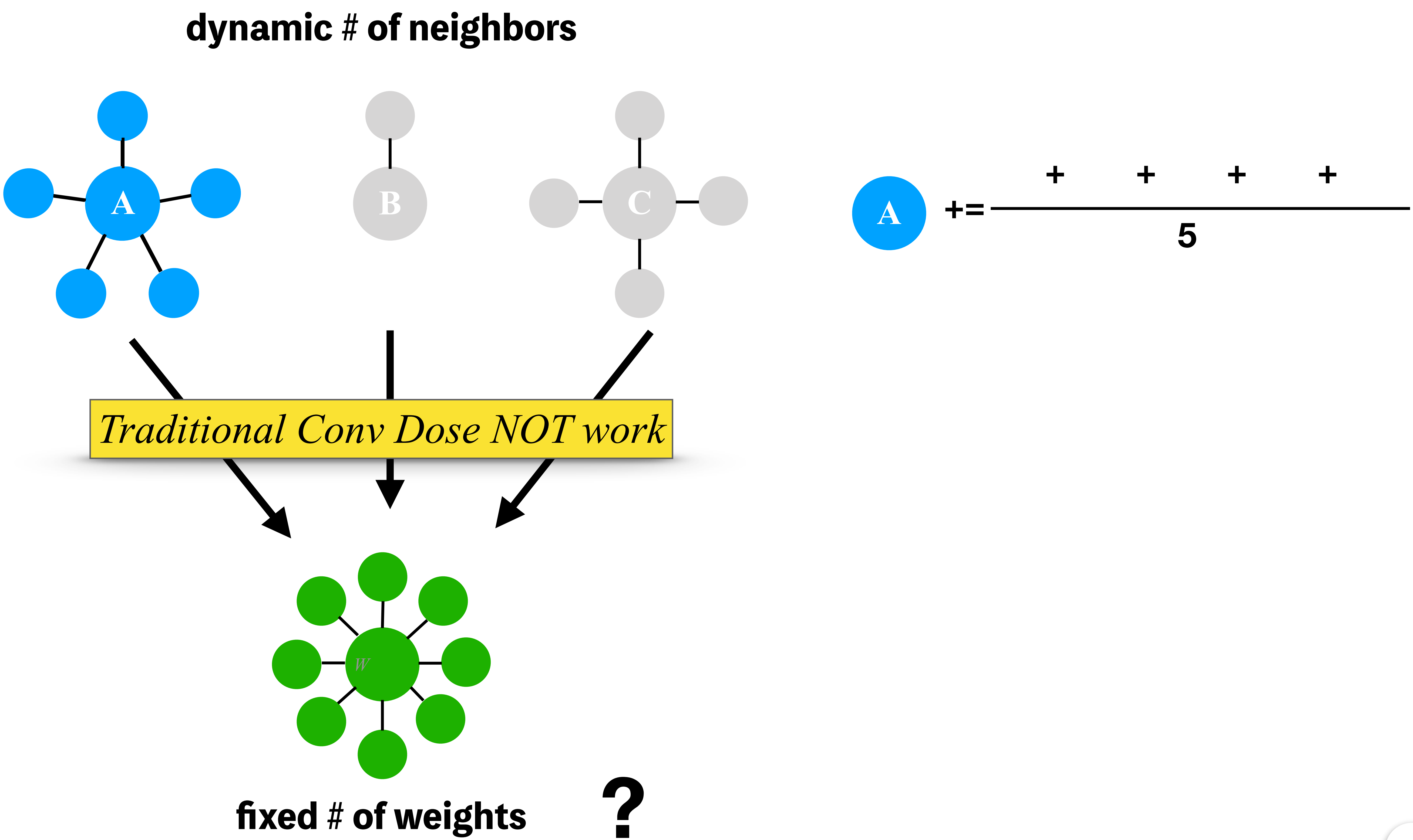
$$= \boxed{f} + \boxed{A \cdot f}$$

Self Average of neighbors

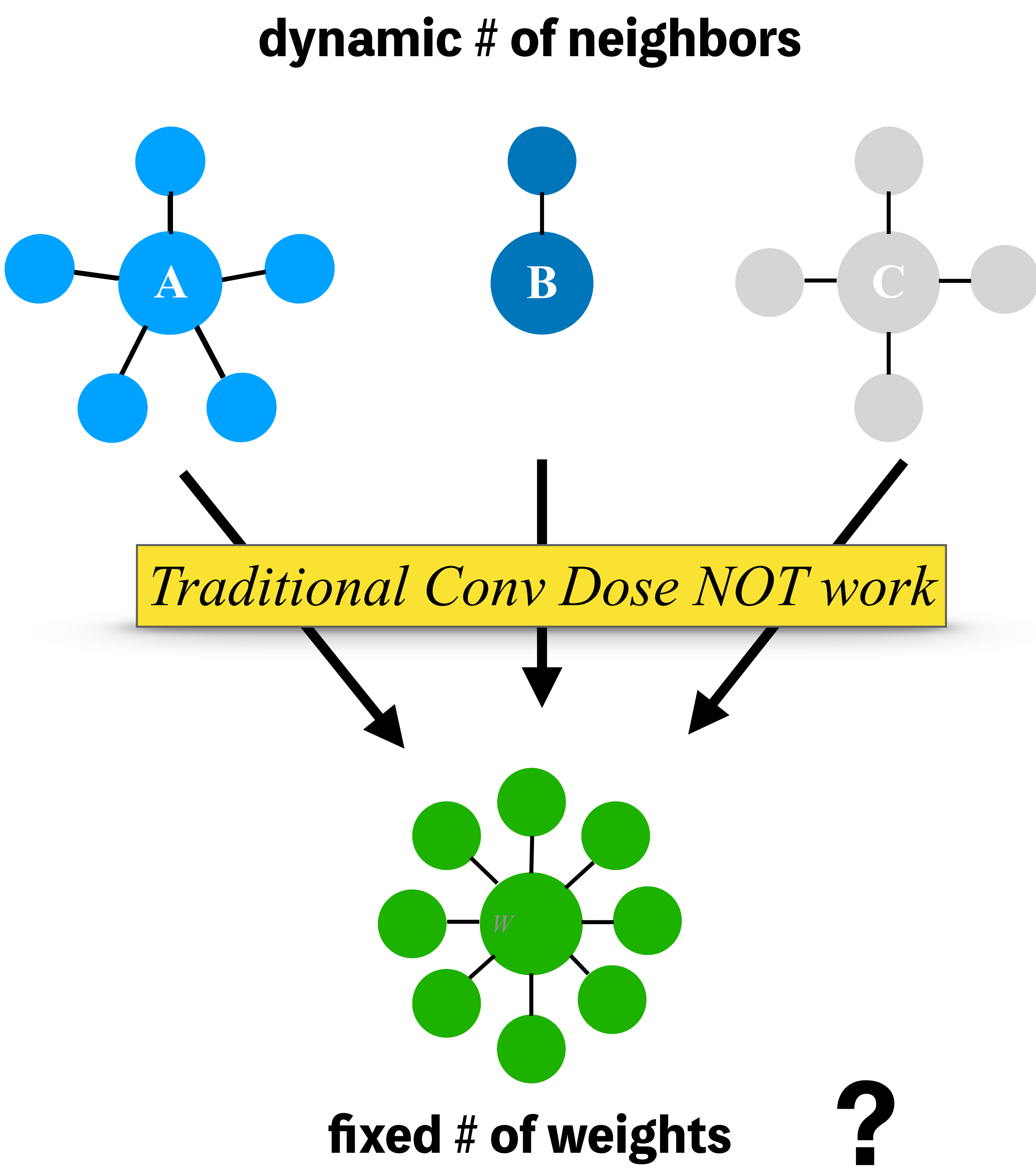
We reach here with the spectral method, but it also makes sense in spatial domain

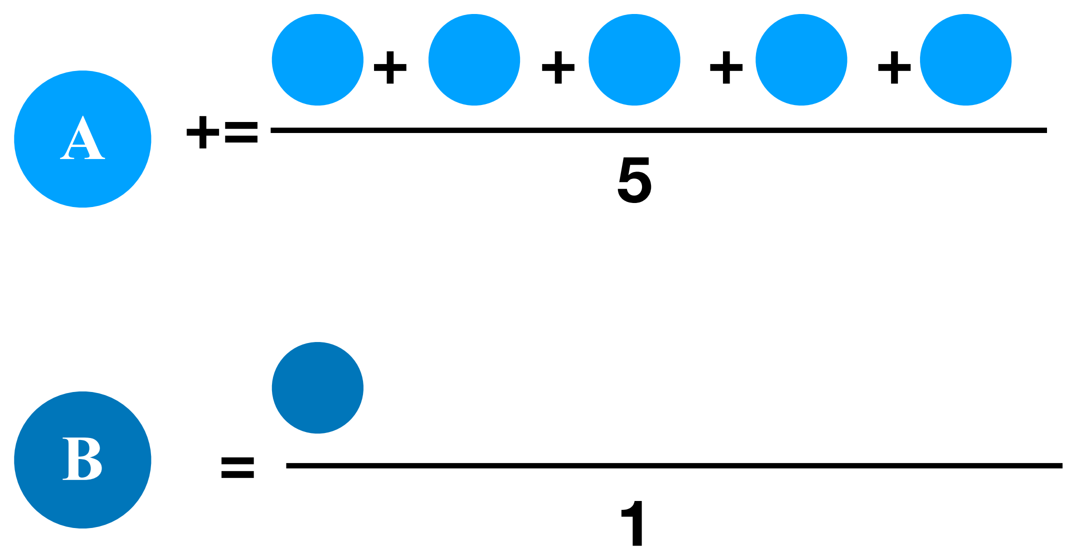
Kipf, T. N., & Welling, M. (2017). **Semi-supervised classification with graph convolutional networks**. ICLR,

Convolution on Graph Data



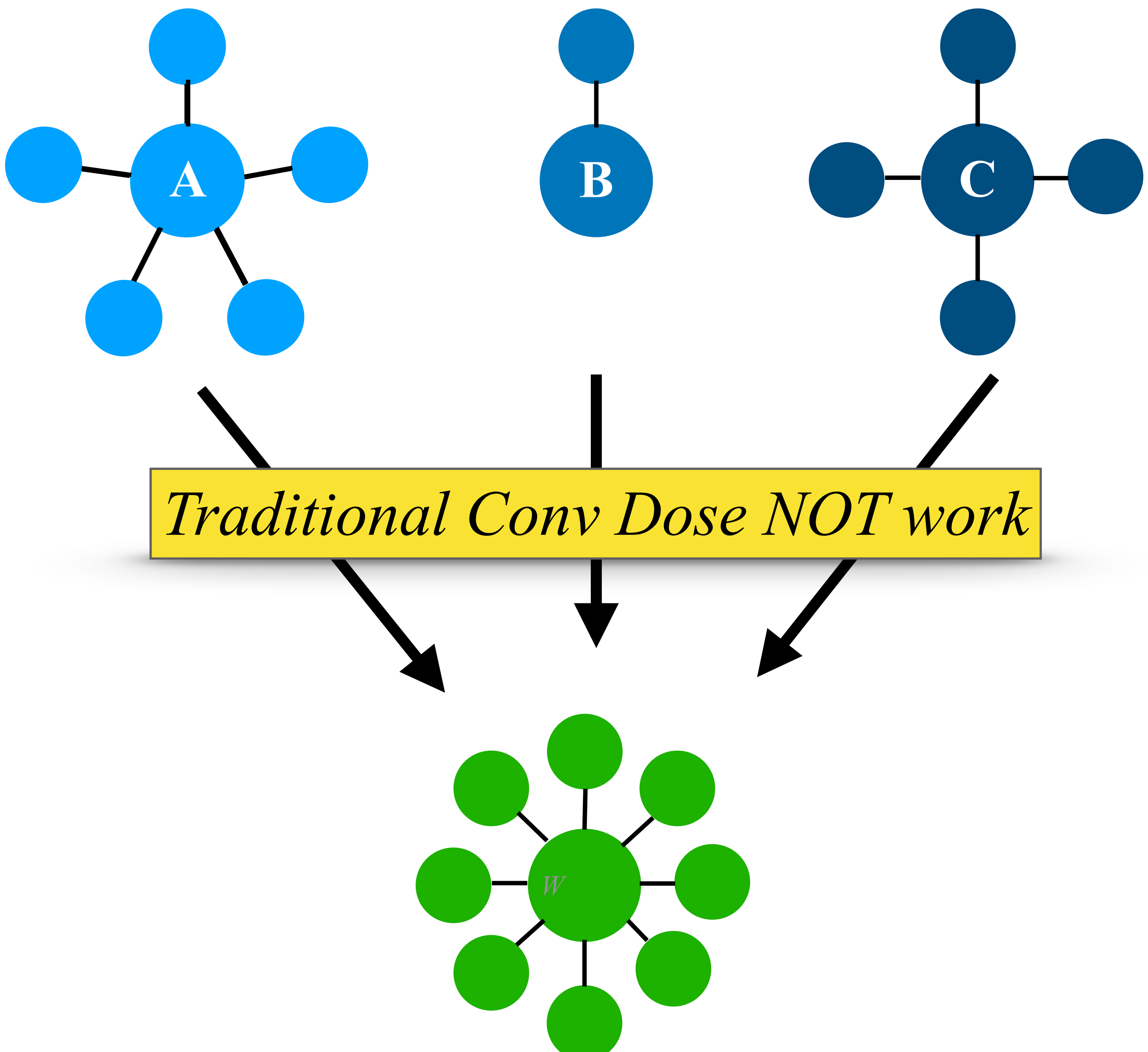
Convolution on Graph Data



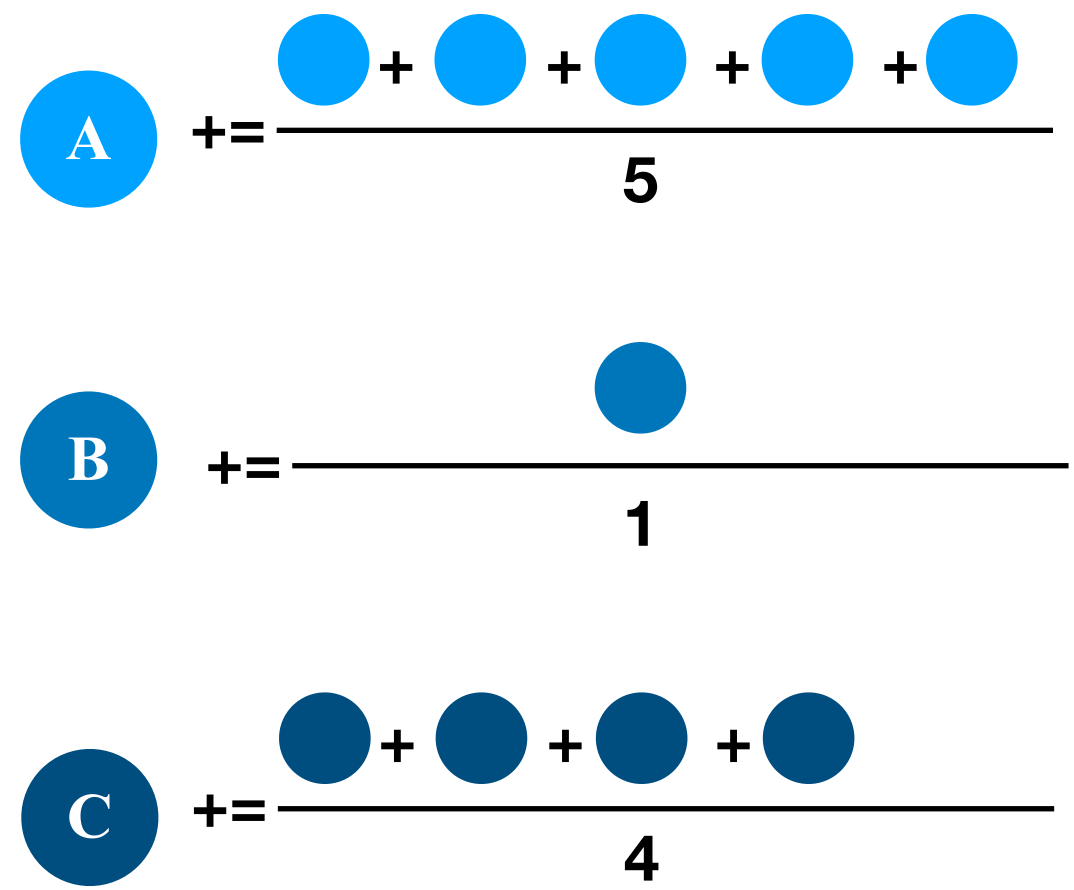


Convolution on Graph Data

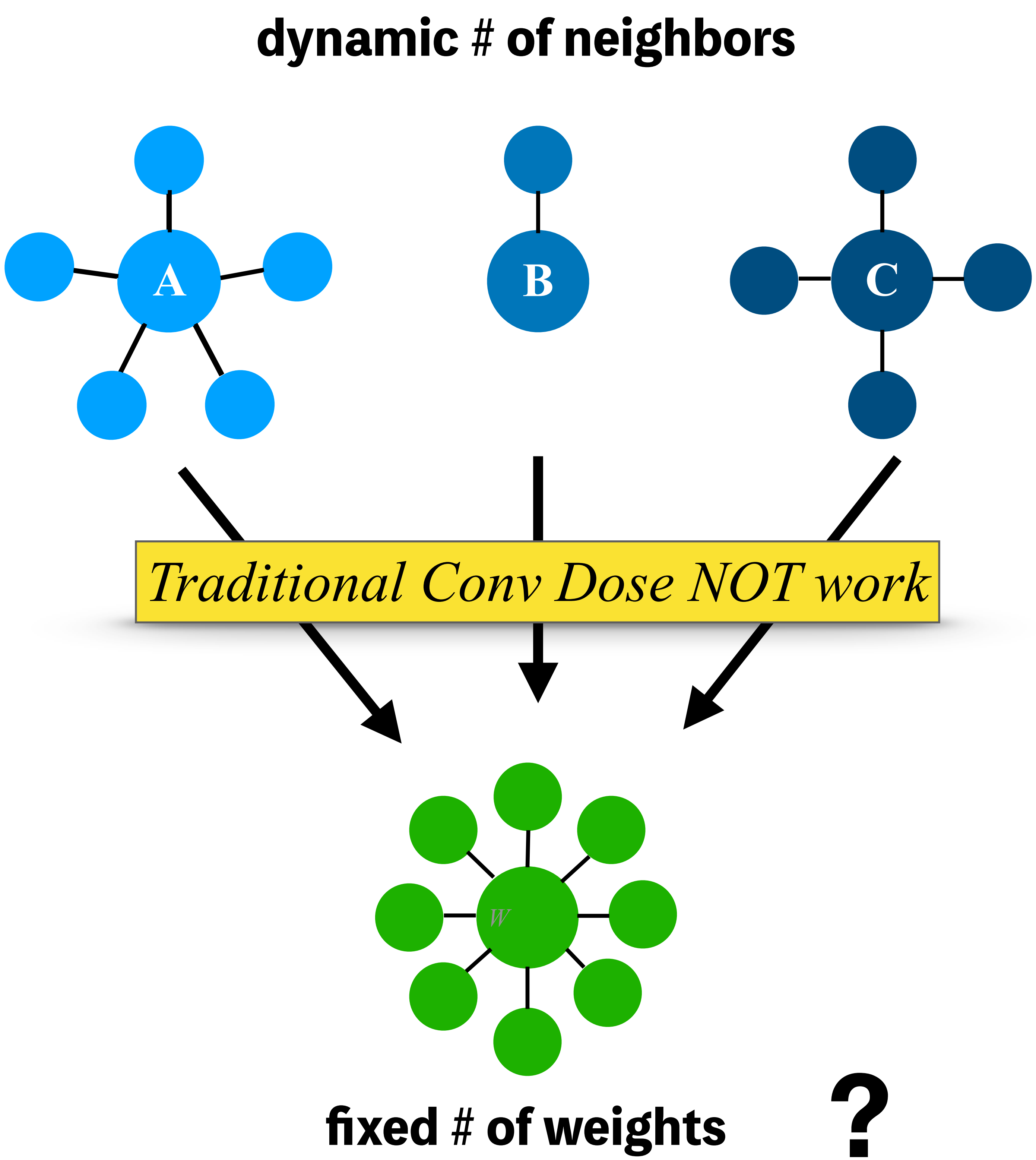
dynamic # of neighbors



fixed # of weights **?**



Convolution on Graph Data



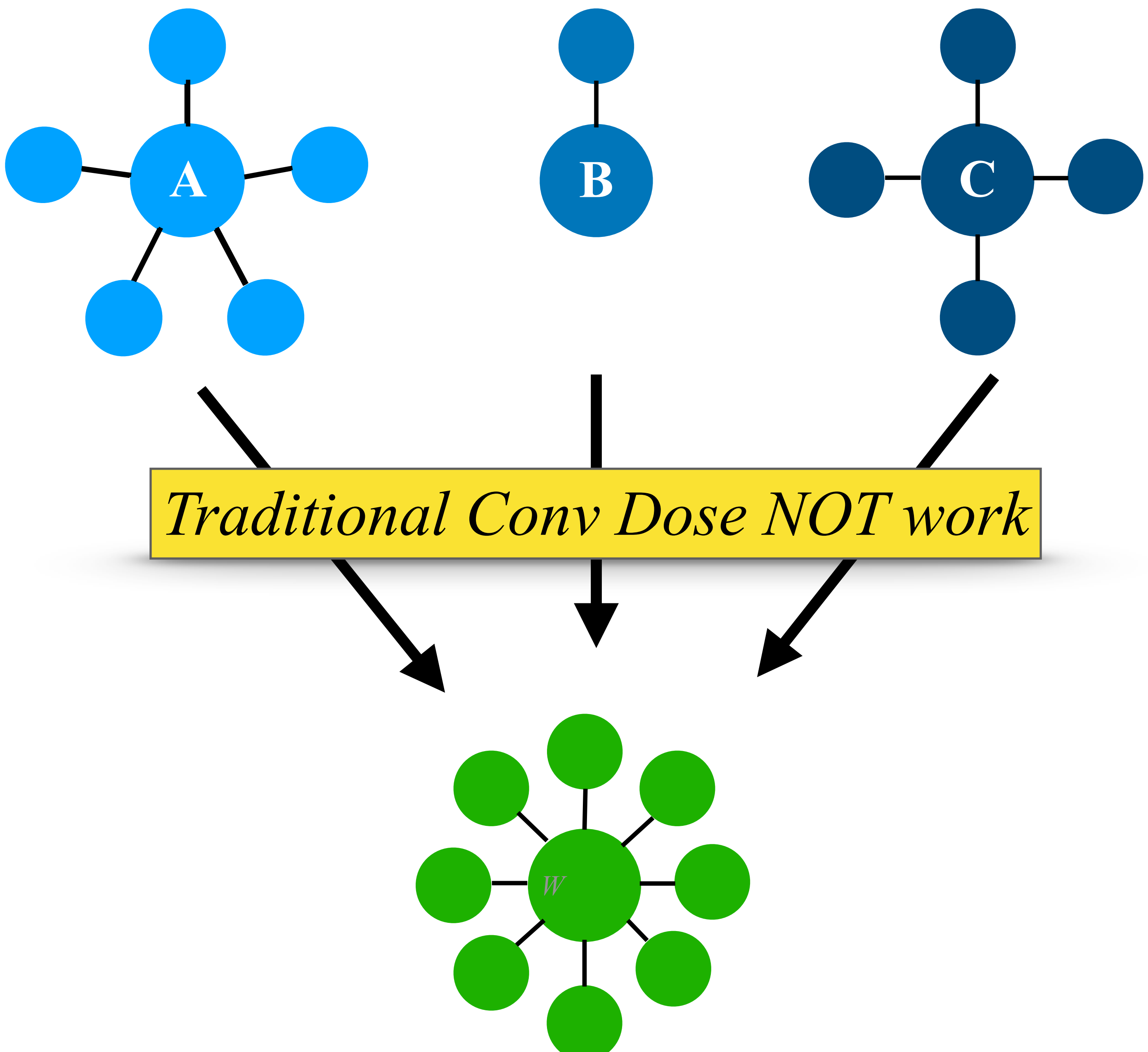
A $\in \mathbb{R}^{node}$

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

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

Convolution on Graph Data

dynamic # of neighbors



fixed # of weights **?**

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

B $\in \mathbb{R}^{node}$ → **Unified Dimension**

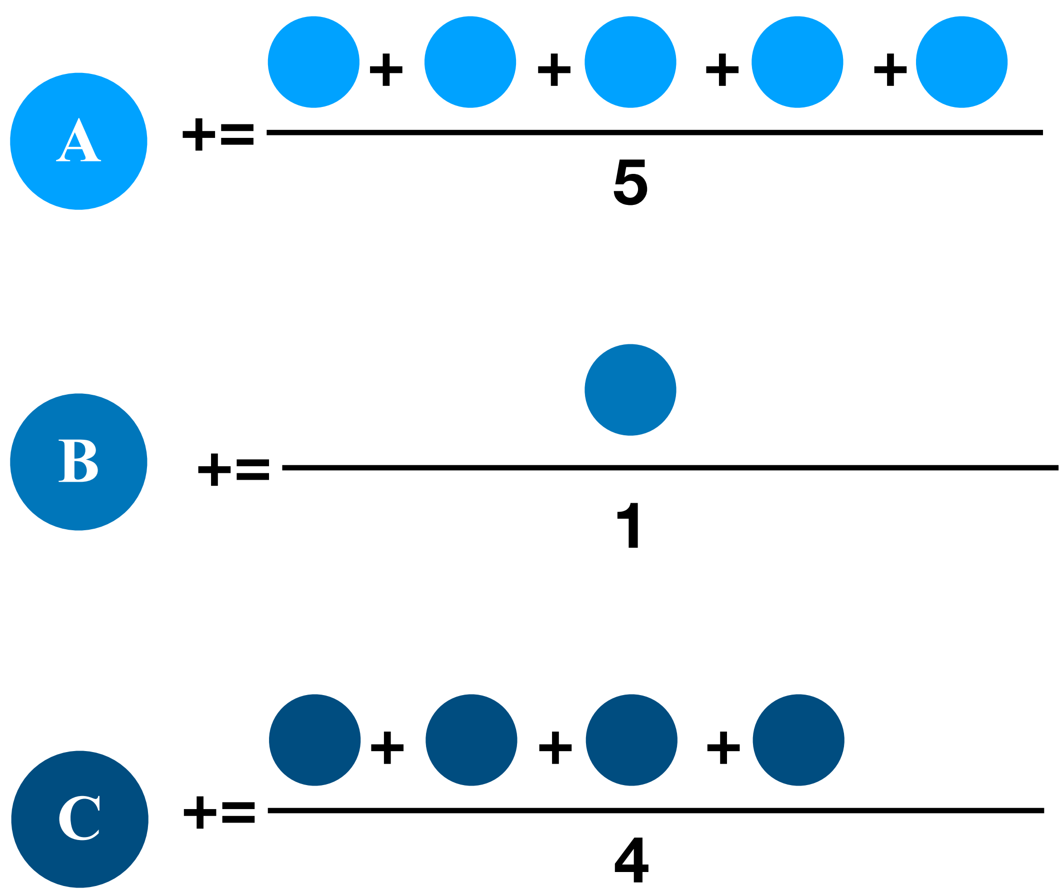
C $\in \mathbb{R}^{node}$

GCN works!

Convolution on Graph Data

If constraint g_θ is polynomial, rational or exp function

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

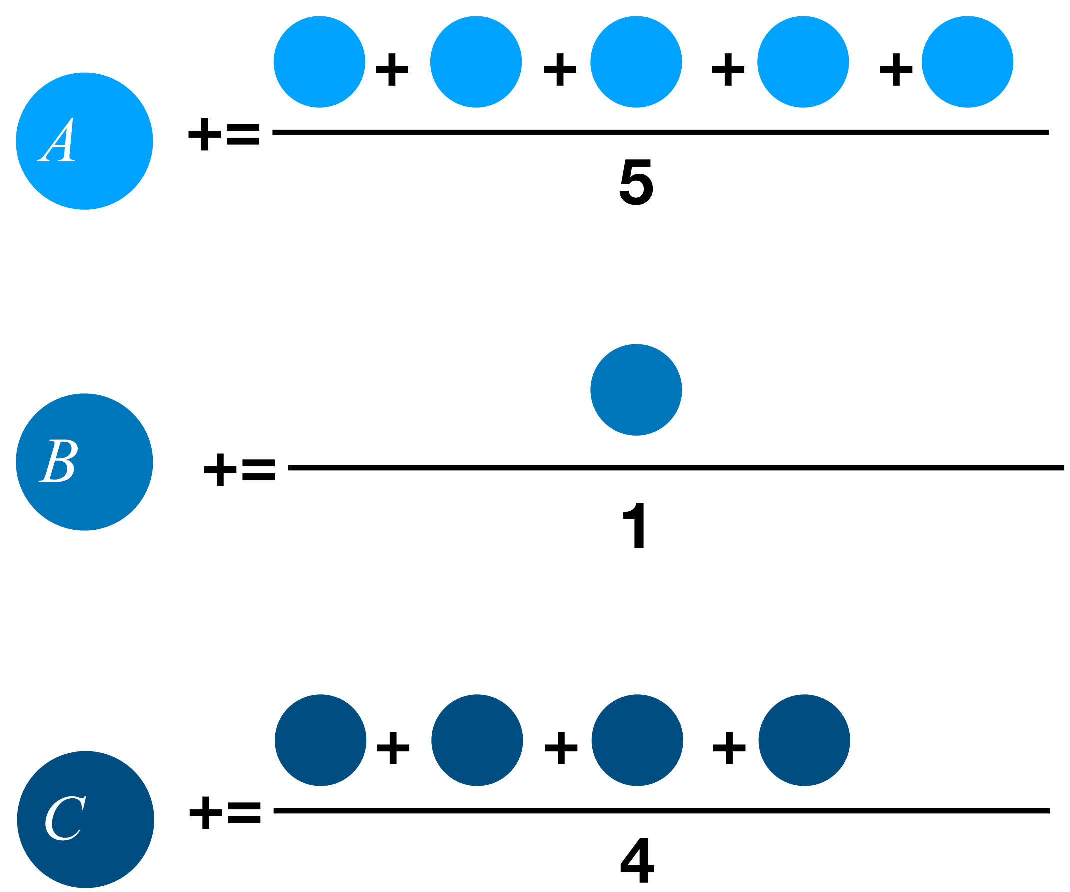


*Kipf, T. N., & Welling, M. (2017). **Semi-supervised classification with graph convolutional networks**. ICLR,*

Convolution on Graph Data

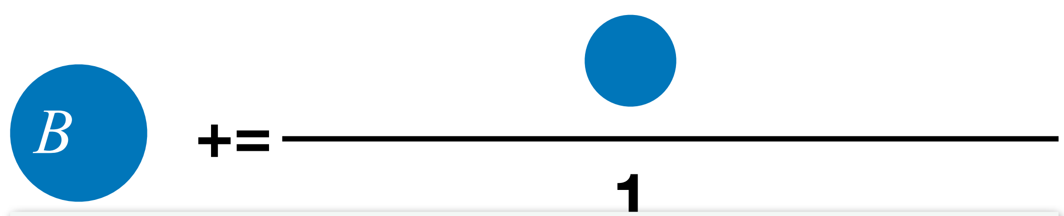
If constraint g_θ is polynomial or rational function

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



Convolution on Graph Data

How about g_θ in other GNNs?



Convolution theorem

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

Space convolution = frequency multiplication

Agenda

- **First Half (1 hour 15 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)

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

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

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

- **Ratio Cut**: $cut(A,B)(\frac{1}{|A|}+\frac{1}{|B|})$
  - Use # **node** to cluster graph

Unnormalized

- **Normalized Cut**: $cut(A,B)(\frac{1}{Vol(A)}+\frac{1}{Vol(B)})$
  - Use # **link** to cluster graph

Normalized

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

Normalization

- Spectral reason

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

Unnormalized

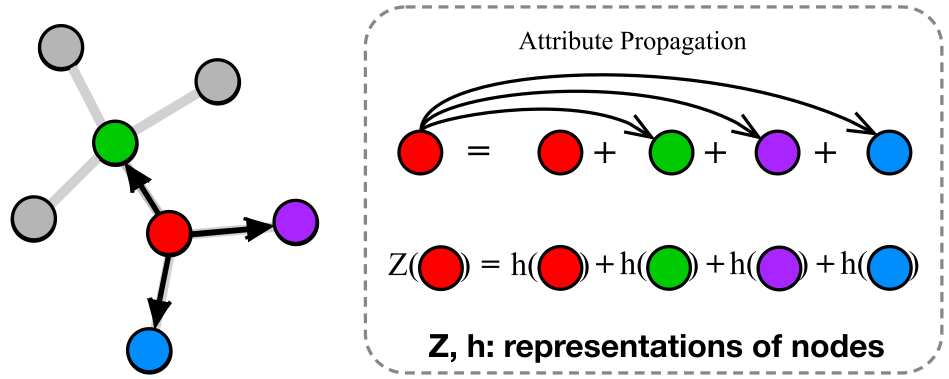
- **Eigenvalue** $\in [0,2]$
  - random walk or symmetric normalization

Normalized

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

Case Study 1: GCN

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



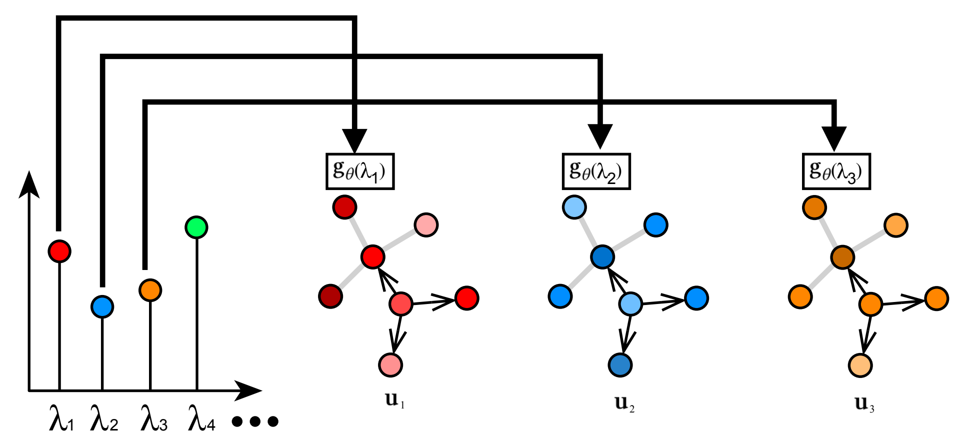
$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}\hat{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

In form of $g_\theta(A)X$

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

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

Case Study 1: GCN

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

Spatial

Average neighbors

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

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

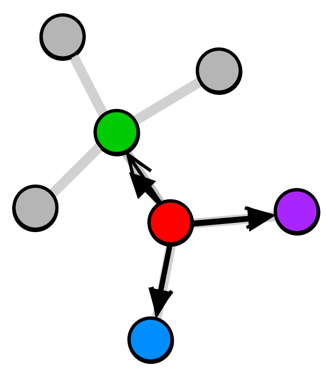
Spectral

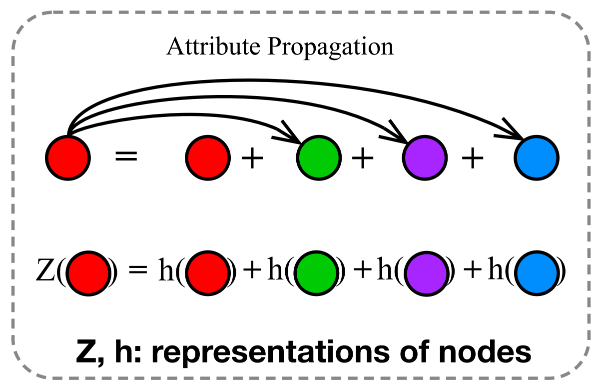
Linear function

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

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

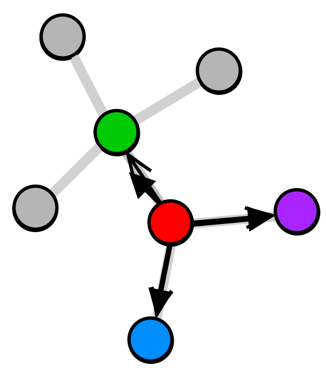
w/ mean aggregator

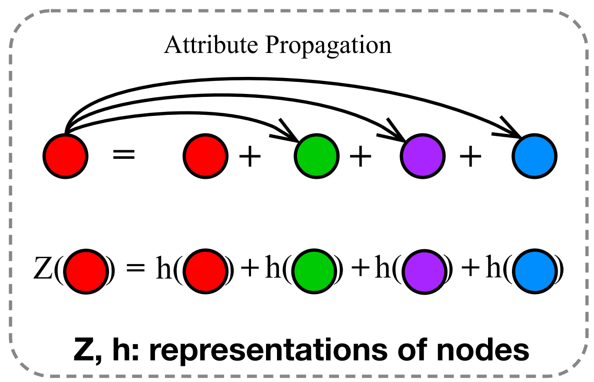
Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." Advances in neural information processing systems 30 (2017).

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

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

Control hyperparameter

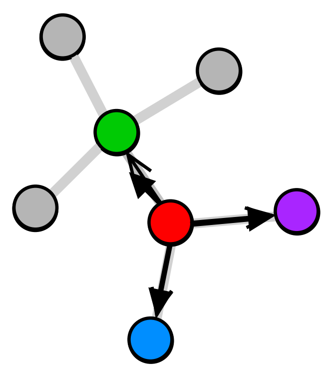
Xu, Keyulu, et al. "How Powerful are Graph Neural Networks?." International Conference on Learning Representations. 2018.

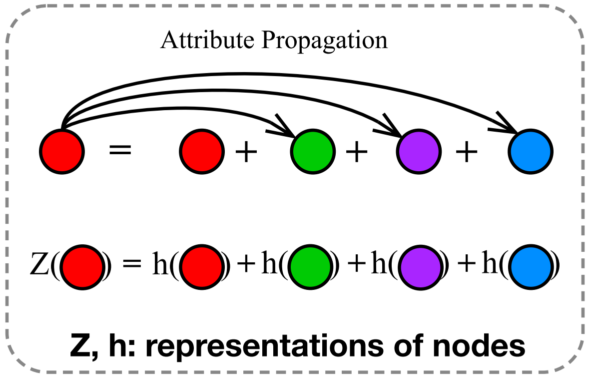
Spatial-based GNN: Linear

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)*

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

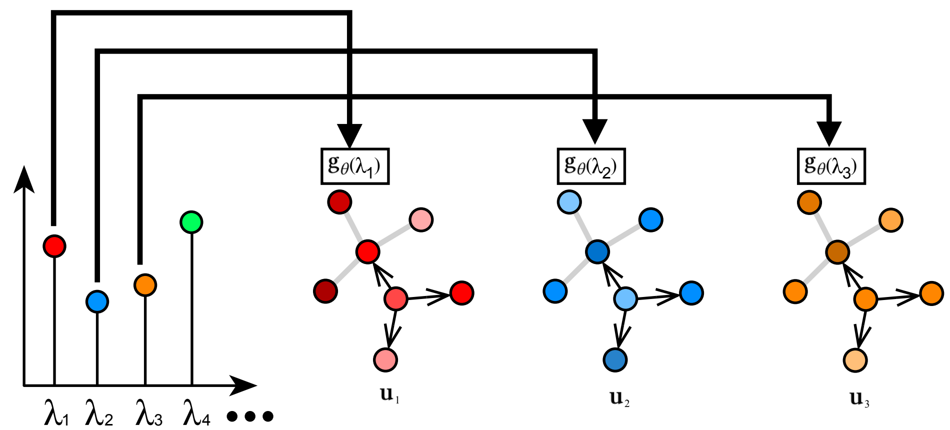
GIN

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

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)*

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

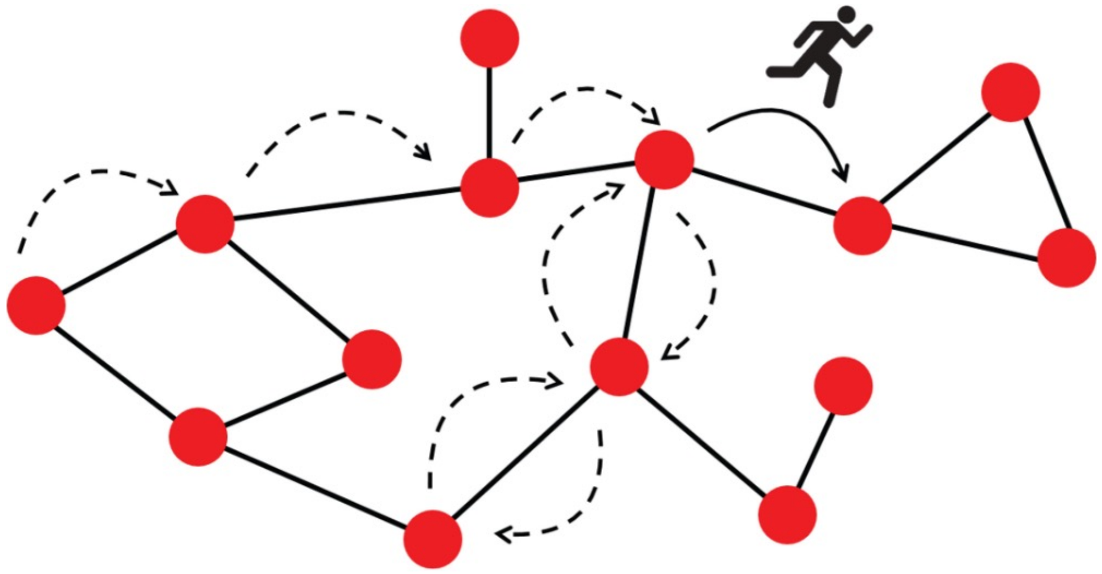
Case Study 2: DeepWalk

- 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

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

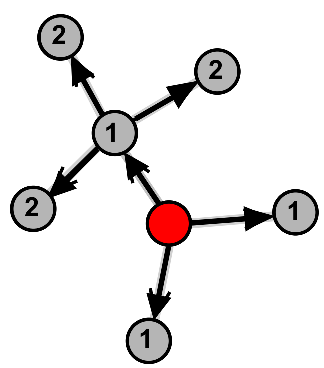


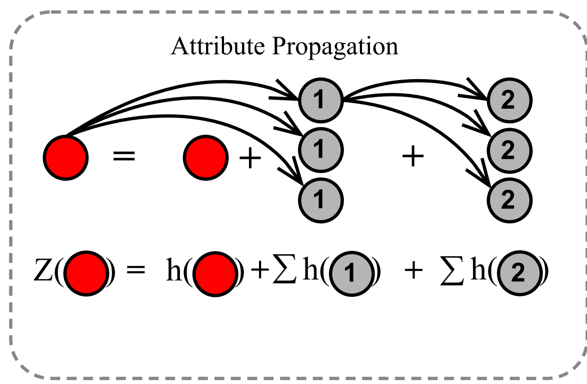
Img credit: DOI: (10.1002/sim.9346)

Spectral-based GNN: Polynomial

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

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)*

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

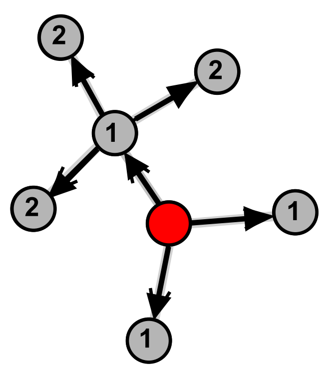
Chebyshev polynomial (1st kind) of L

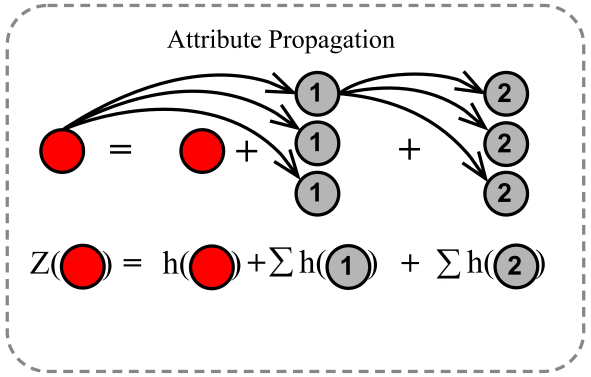
*Defferrard, et al., "**Convolutional neural networks on graphs with fast localized spectral filtering.**" Advances in neural information processing systems 29 (2016).*

Spectral-based GNN: Polynomial

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

Polynomial





DeepWalk *Bryan Perozzi et al. (2014)*

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

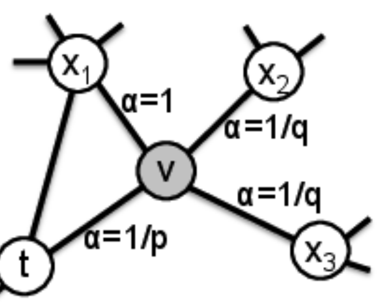
ChebyNet *Defferrard, Michael et al. (2016)*

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

Node2Vec *Aditya Grover et al. (2016)*

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

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

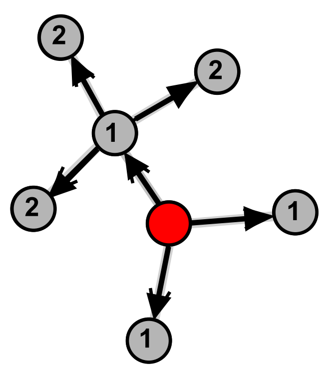


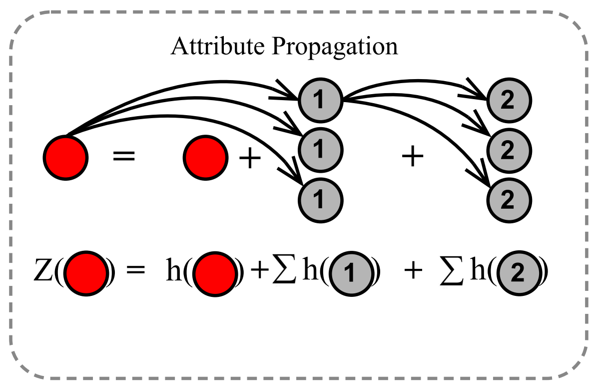
$$\mathbf{Z} = \left(\frac{1}{p} \cdot \mathbf{I} + \tilde{\mathbf{A}} + \frac{1}{q}\left(\tilde{\mathbf{A}}^2 - \tilde{\mathbf{A}}\right)\right)\mathbf{X} = \left[\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\tilde{\mathbf{A}} + \frac{1}{q}\tilde{\mathbf{A}}^2\right]\mathbf{X} = \mathbf{P}(\tilde{\mathbf{A}})\mathbf{X}$$

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)*

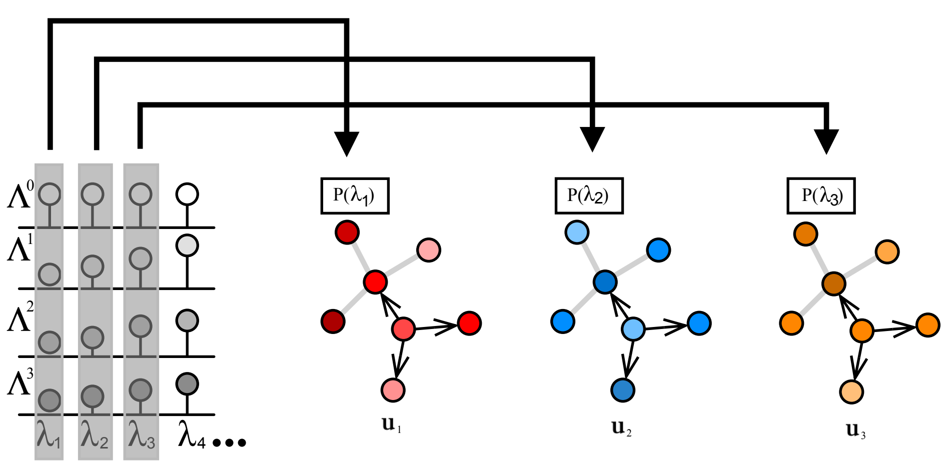
$$\mathbf{Z} = \sum_{k=0}^{K-1}\theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \left[\tilde{\theta}_0\mathbf{I} + \tilde{\theta}_1(\mathbf{I} - \tilde{\mathbf{A}}) + \tilde{\theta}_2(\mathbf{I} - \tilde{\mathbf{A}})^2 + \ldots\right]\mathbf{X} = \left(\phi\mathbf{I} + \sum_{i=1}^{k}\psi_i\tilde{\mathbf{A}}^i\right)\mathbf{X}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

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)*

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

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

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

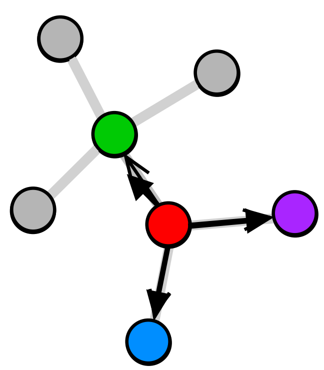
Node2Vec *Aditya Grover et al. (2016)*

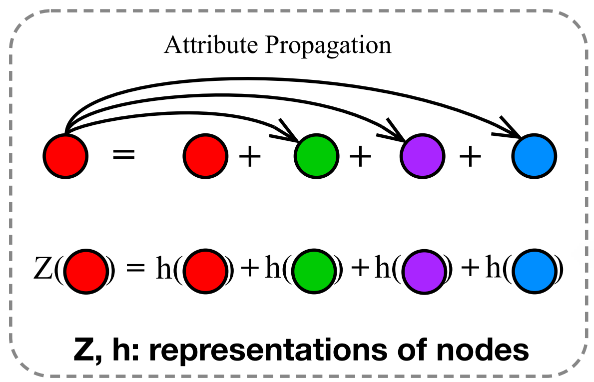
$$\mathbf{Z} = \left[\left(1 + \frac{1}{p}\right)\mathbf{I} - \left(1 + \frac{1}{q}\right)\tilde{\mathbf{L}} + \frac{1}{q}\tilde{\mathbf{L}}^2\right]\mathbf{X} = \mathbf{U}\left[\left(1 + \frac{1}{p}\right) - \left(1 + \frac{1}{q}\right)\tilde{\Lambda} + \frac{1}{q}\tilde{\Lambda}^2\right]\mathbf{U}^\top\mathbf{X}$$

Linear v.s. Polynomial

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

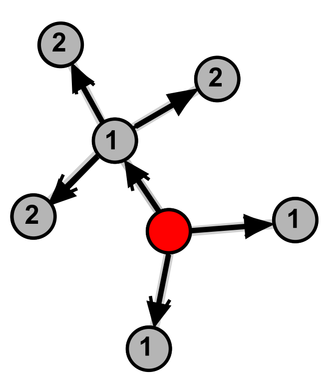
Linear

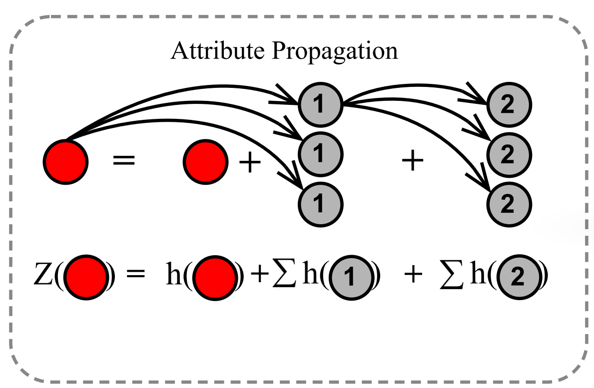




only consider the ***direct*** *neighbors*

Polynomial





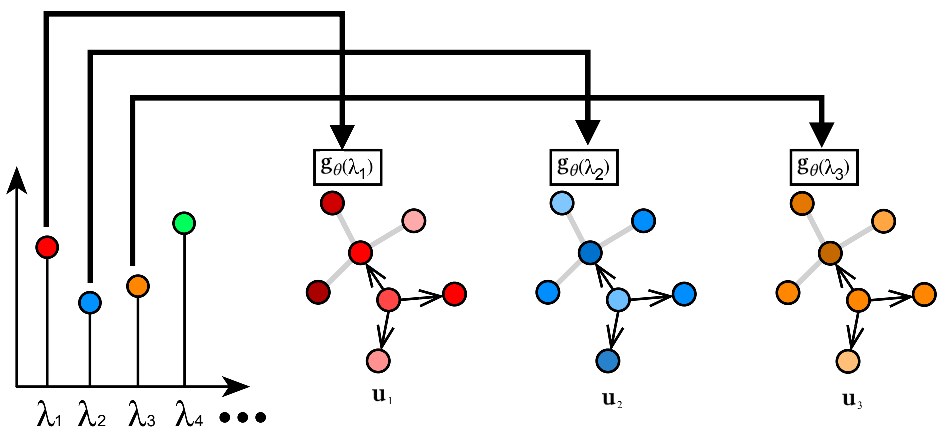
consider the ***higher-order*** *neighbors*

Linear v.s. Polynomial

"spectral response functions"

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

Linear



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

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

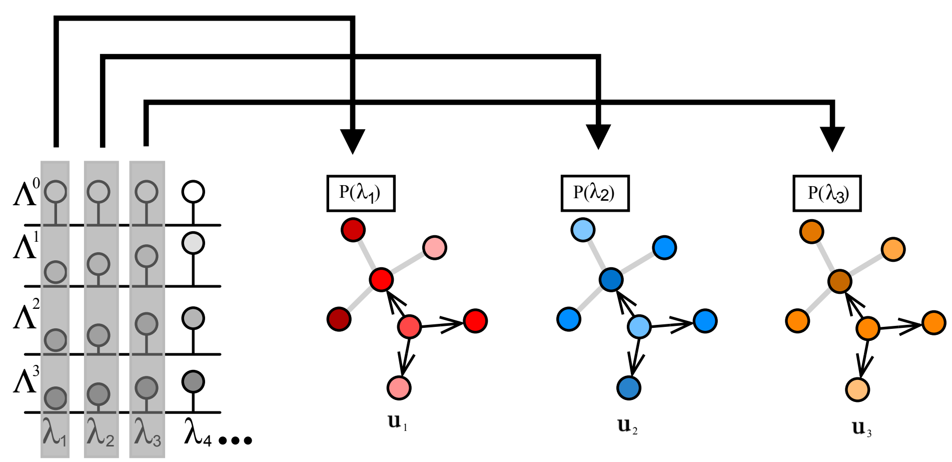
GraphSAGE *Will Hamilton et al. (2017)*

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

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)*

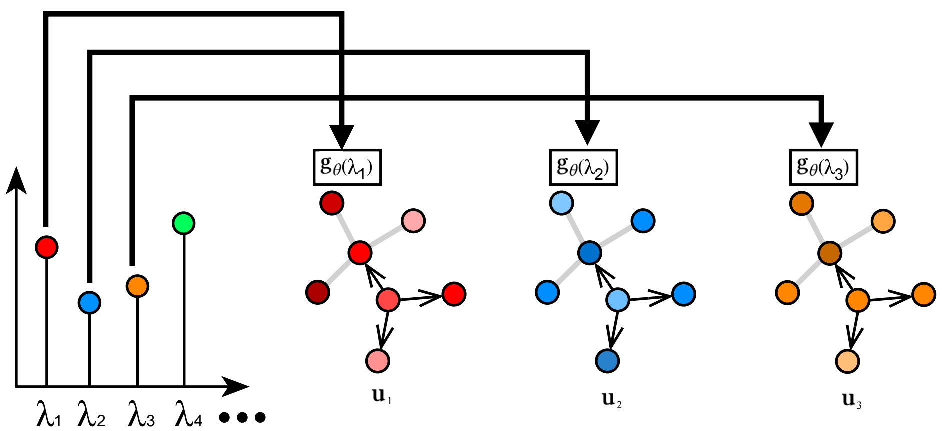
$$\mathbf{Z} = \left[\left(1+\frac{1}{p}\right)\mathbf{I}-\left(1+\frac{1}{q}\right)\tilde{\mathbf{L}}+\frac{1}{q}\tilde{\mathbf{L}}^2\right]\mathbf{X} = \mathbf{U}\left[\left(1+\frac{1}{p}\right)-\left(1+\frac{1}{q}\right)\tilde{\Lambda}+\frac{1}{q}\tilde{\Lambda}^2\right]\mathbf{U}^\top\mathbf{X}$$

Linear v.s. Polynomial

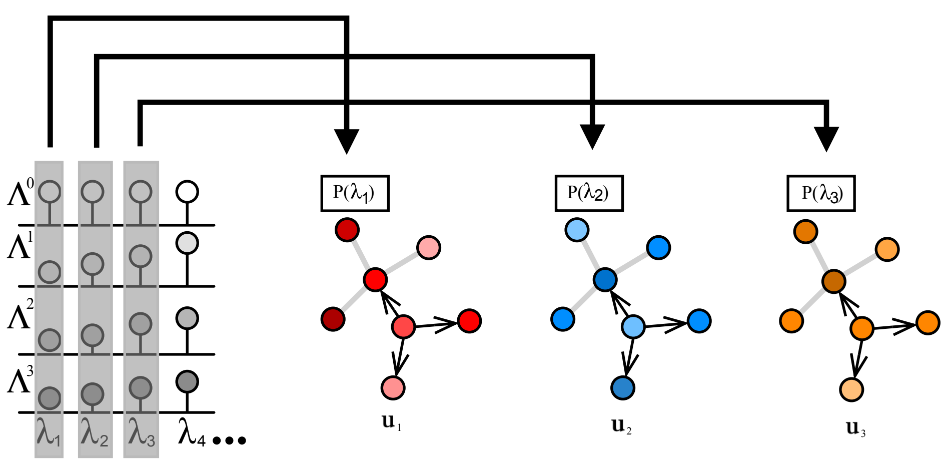
"spectral response functions"

Function of **eigenvalue** $\quad U{\color{red}g_\theta(\Lambda)}U^TX$

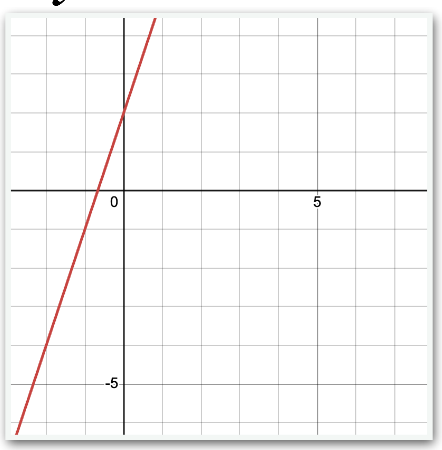
Linear



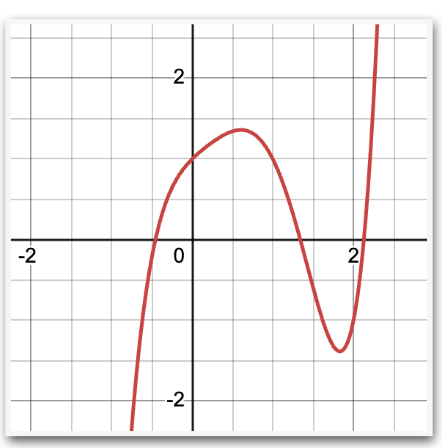
Polynomial



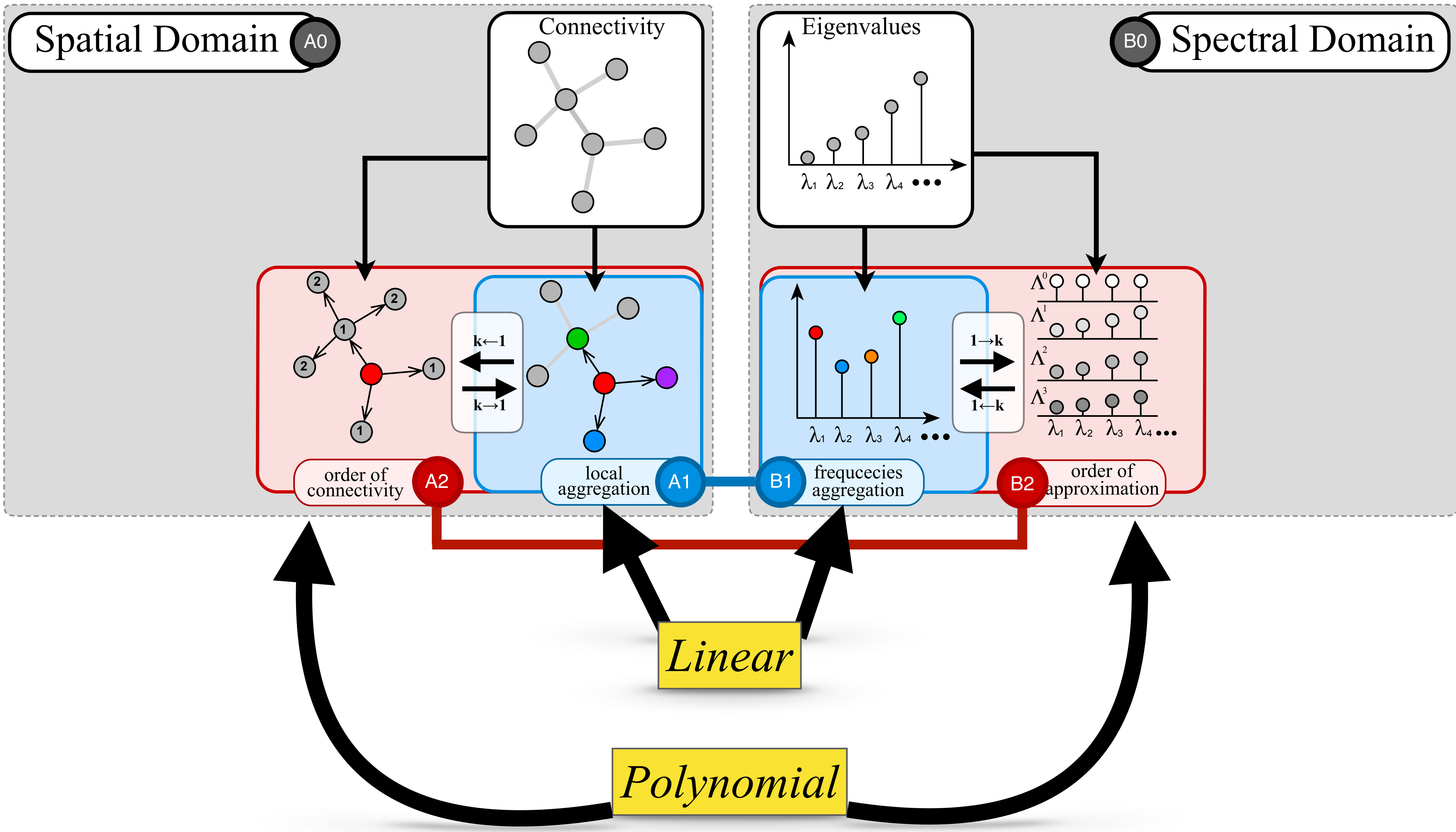
$$y = 3x + 2$$



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



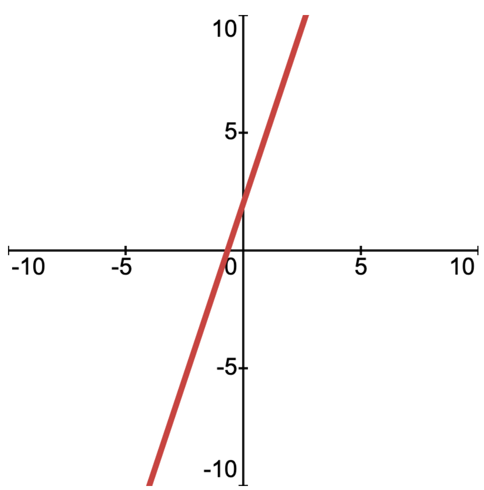
Linear v.s. Polynomial



Beyond Polynomial: Rational Model

GCN

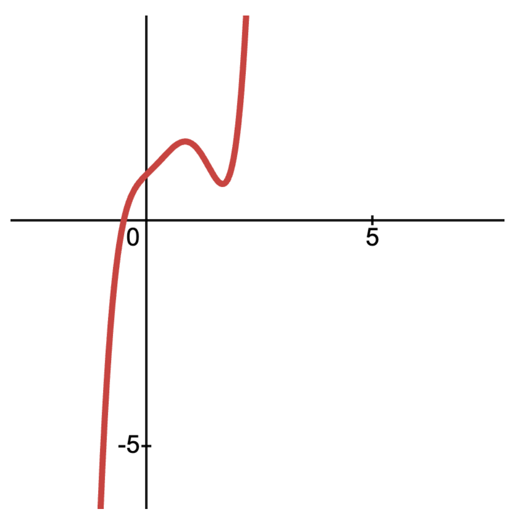
$$y = 3x + 2$$



Linear

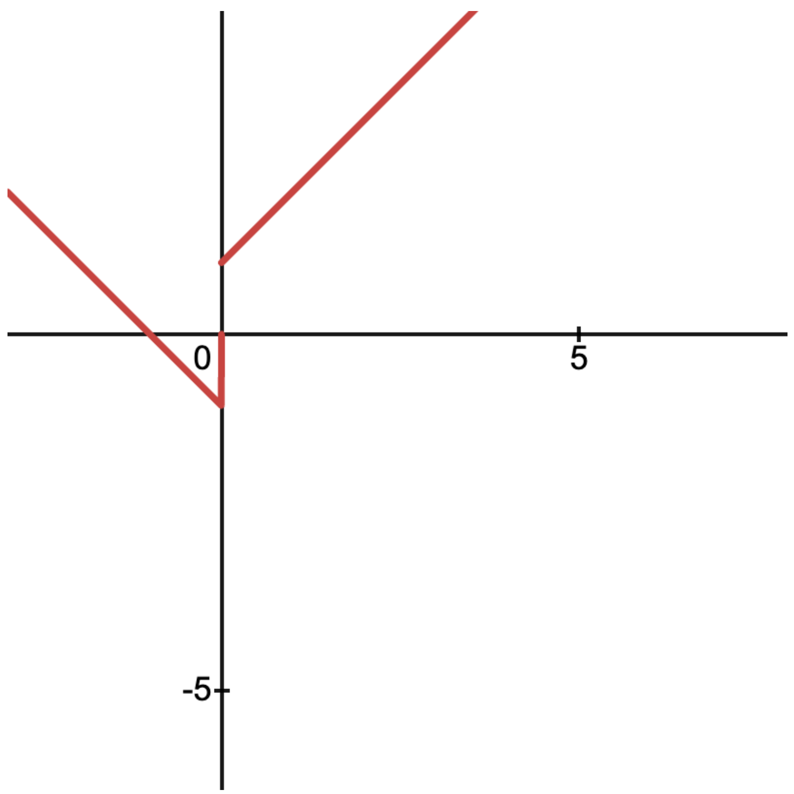
ChebNet

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



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

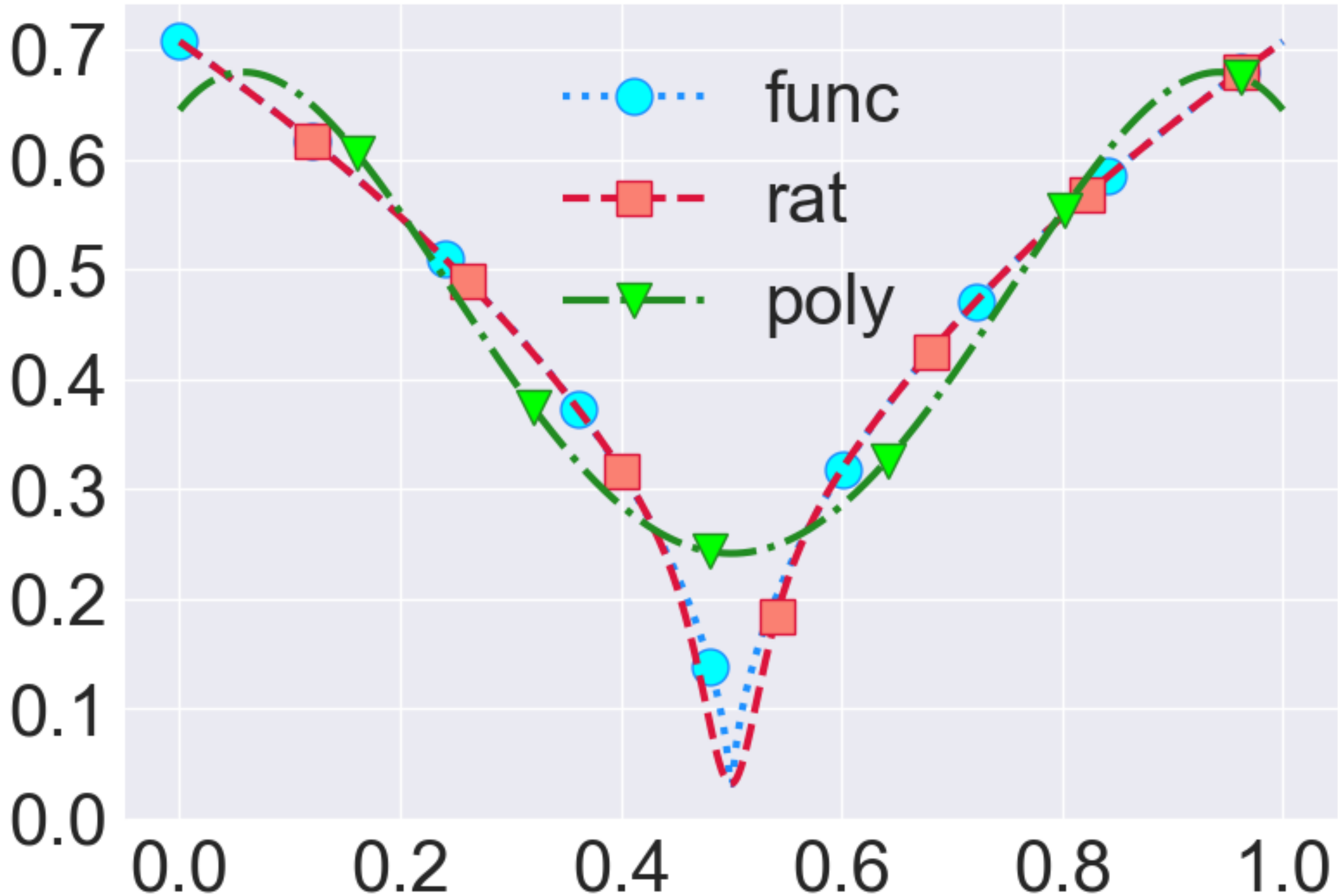
model **complicated** structure with a fairly low degree

excellent interpolatory/extrapolatory/asymptotic properties

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

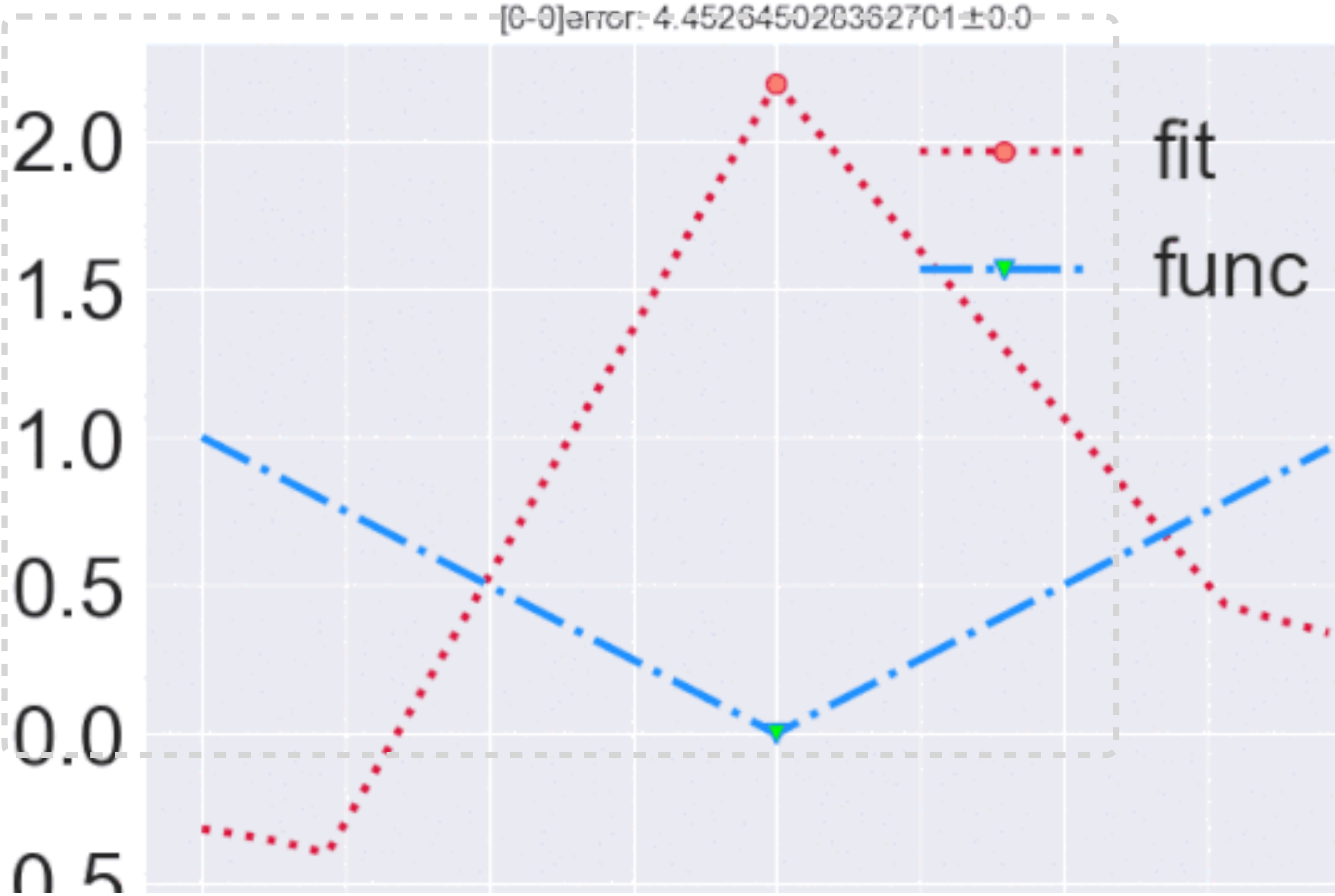
Beyond Polynomial: Rational Model

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)

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

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

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

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

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

$\mathbf{L}$ graph Laplacian

x node attributes

Rational

Z. Chen et al. (2018)

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

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

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

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

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

Polynomial v.s. Rational

Polynomial

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

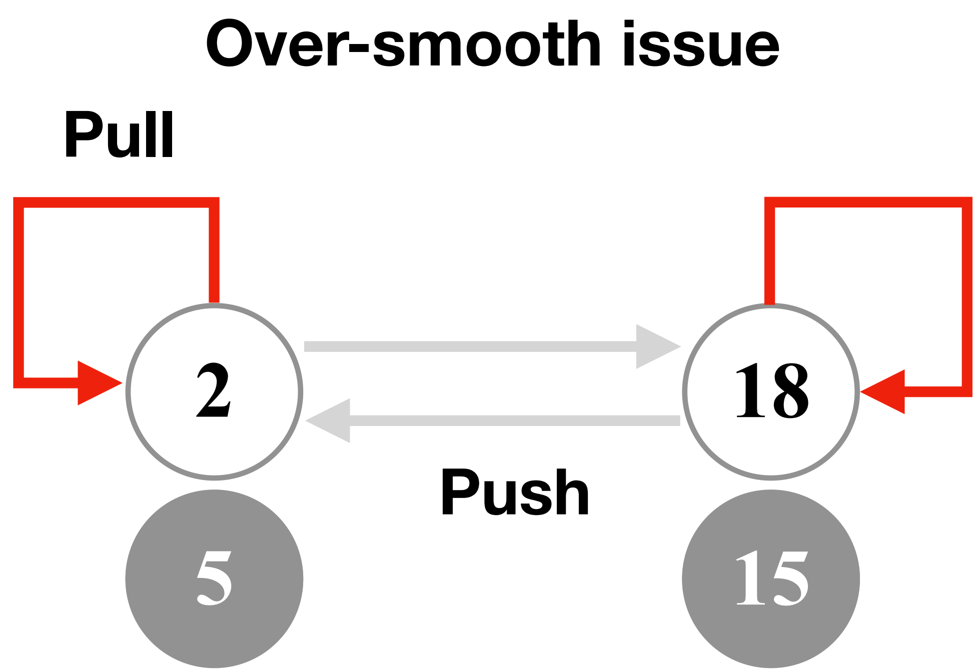
Label propagation

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

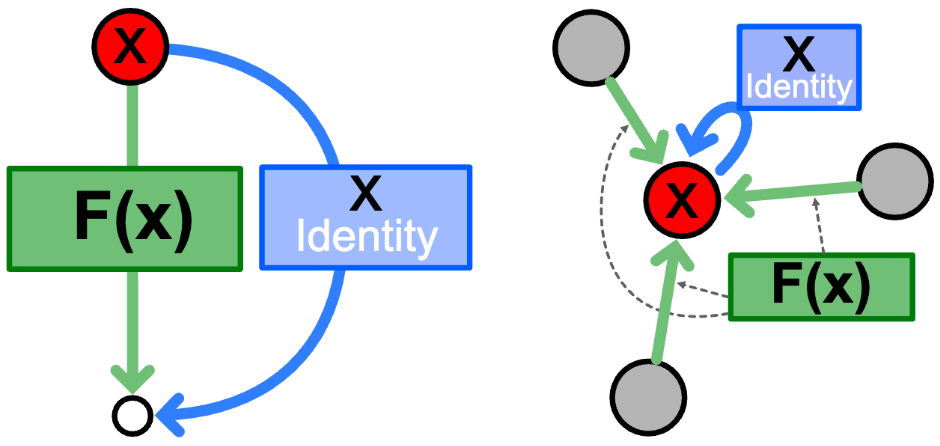


FIG. 6. Left: Residual Learning $x' = F(x) + x$; Right: Rational Aggregation: $x' = F(x) + x$

Polynomial v.s. Rational

Polynomial

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

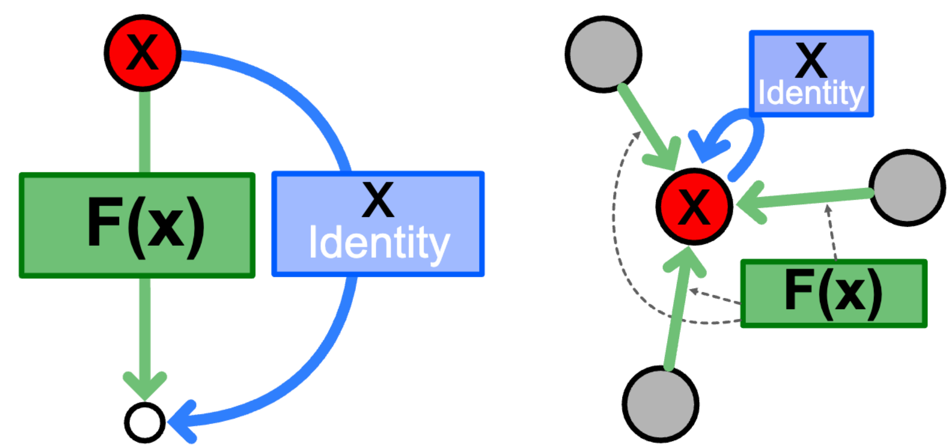
Label propagation

Rational

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

Reverse Label propagation

Polynomial



$$f + A \cdot f$$

Rational

Self Average of neighbors

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

Polynomial v.s. Rational

Polynomial

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

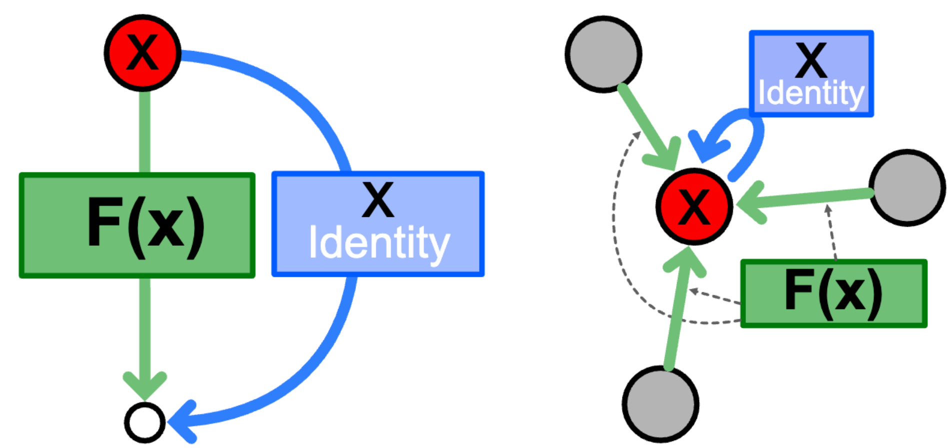
Label propagation

Rational

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

Reverse Label propagation

Polynomial



$$f + A \cdot f$$

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

Rational

Self Average of neighbors

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

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

Beyond Polynomial: Rational Model

Johannes Klicpera et al. (2018)

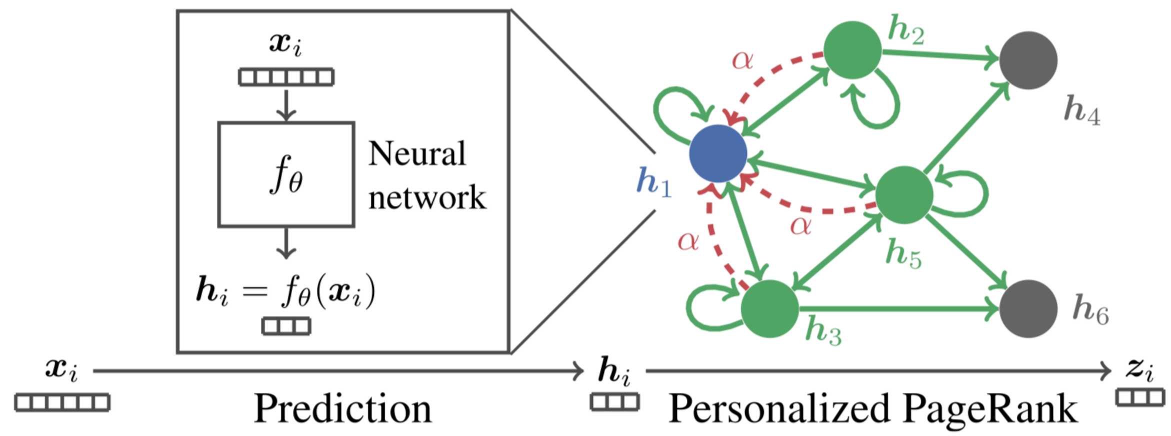
Personalized Page Rank (information retrieval)

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

$(1-\alpha)$ 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:

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

$$\downarrow$$

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

Beyond Polynomial: Rational Model

Johannes Klicpera et al. (2018)

Personalized Page Rank (information retrieval)

Filippo Maria Bianchi et al. (2018)

ARMA (time series)

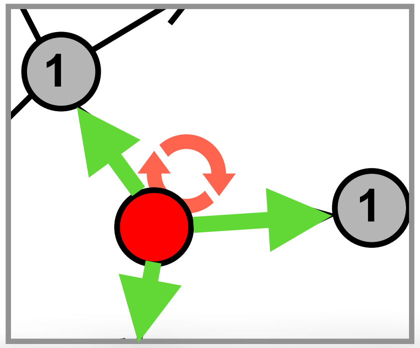
$$\boldsymbol{\pi}_{\text{ppr}}(\boldsymbol{i}_x) = (1-\alpha)\hat{\tilde{\mathbf{A}}}\boldsymbol{\pi}_{\text{ppr}}(\boldsymbol{i}_x) + \alpha \boldsymbol{i}_x$$

$(1-\alpha)$ α

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

a b

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



One iteration

K iterations →K-order rational function

Bianchi, Filippo Maria, et al. "Graph neural networks with convolutional arma filters." IEEE transactions on pattern analysis and machine intelligence 44.7 (2021): 3496-3507.

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

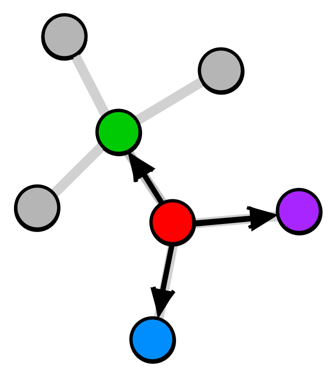
      Imply 5 iterations/layers
- **No**
  - Computational Complexity: Matrix Inversion

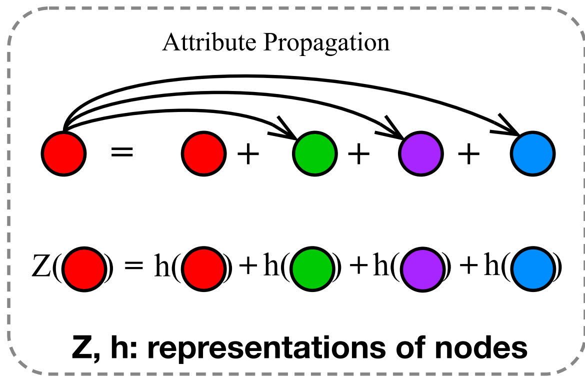
    $\mathcal{O}(n^3)$

Spatial-based GNN

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

Linear





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

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

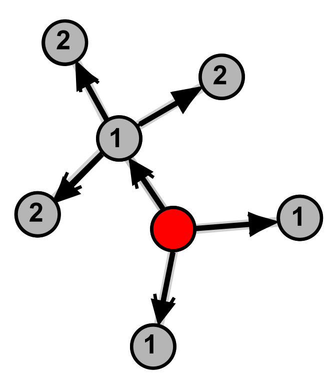
GraphSAGE *Will Hamilton et al. (2017)*

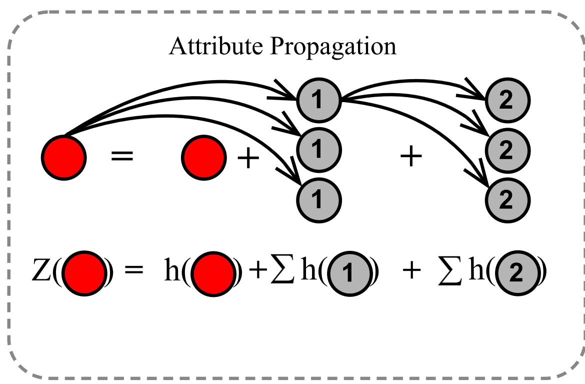
$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I}+\mathbf{A})\mathbf{D}^{-\frac{1}{2}}\mathbf{X} = (\mathbf{I}+\tilde{\mathbf{A}})\mathbf{X}$$

GIN *Xukeyu Lu et al. (2019)*

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

Polynomial





DeepWalk *Bryan Perozzi et al. (2014)*

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

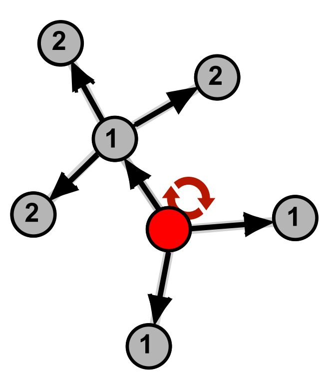
ChebyNet *Defferrard, Michael et al. (2016)*

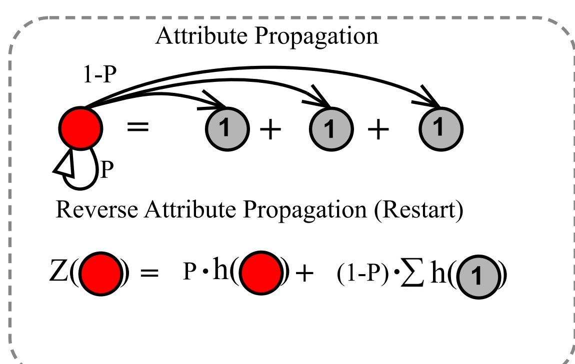
$$\mathbf{Z} = \textstyle\sum_{k=0}^{K-1}\theta_k T_k(\tilde{\mathbf{L}})\mathbf{X} = \left[\tilde{\theta}_0\mathbf{I}+\tilde{\theta}_1(\mathbf{I}-\tilde{\mathbf{A}})+\tilde{\theta}_2(\mathbf{I}-\tilde{\mathbf{A}})^2+\ldots\right]\mathbf{X} = \left(\phi\mathbf{I}+\textstyle\sum_{i=1}^{k}\psi_i\tilde{\mathbf{A}}^i\right)\mathbf{X}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

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)*

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

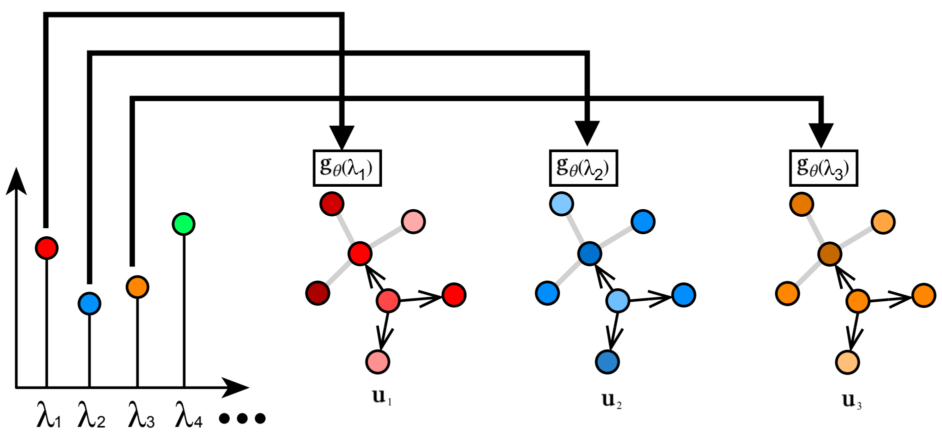
Auto Regressive Filter *Qimai Li et al. (2019)*

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

Spectral-based GNN

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

Linear



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

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

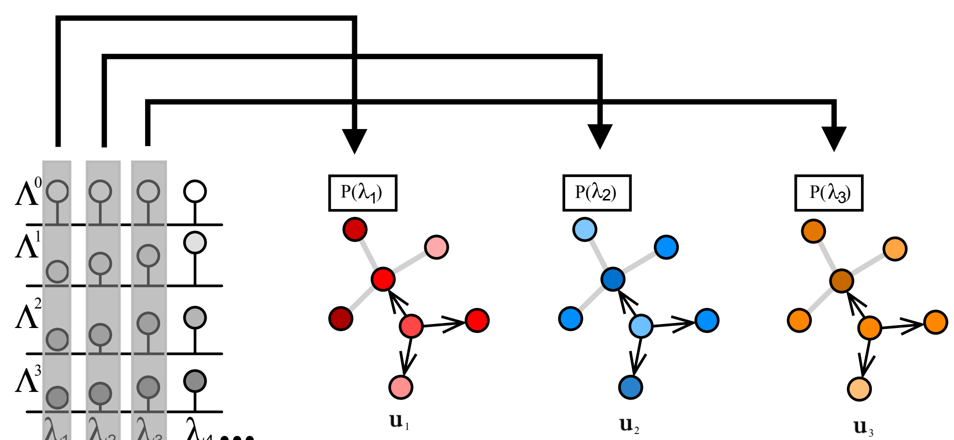
GraphSAGE *Will Hamilton et al. (2017)*

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

GIN *Xukeyu Lu et al. (2019)*

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

Polynomial



DeepWalk *Bryan Perozzi et al. (2014)*

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

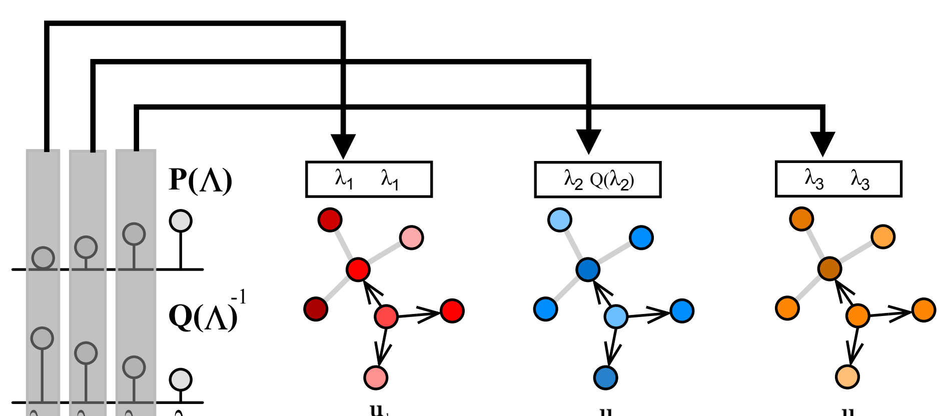
ChebyNet *Defferrard, Michael et al. (2016)*

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

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)*

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

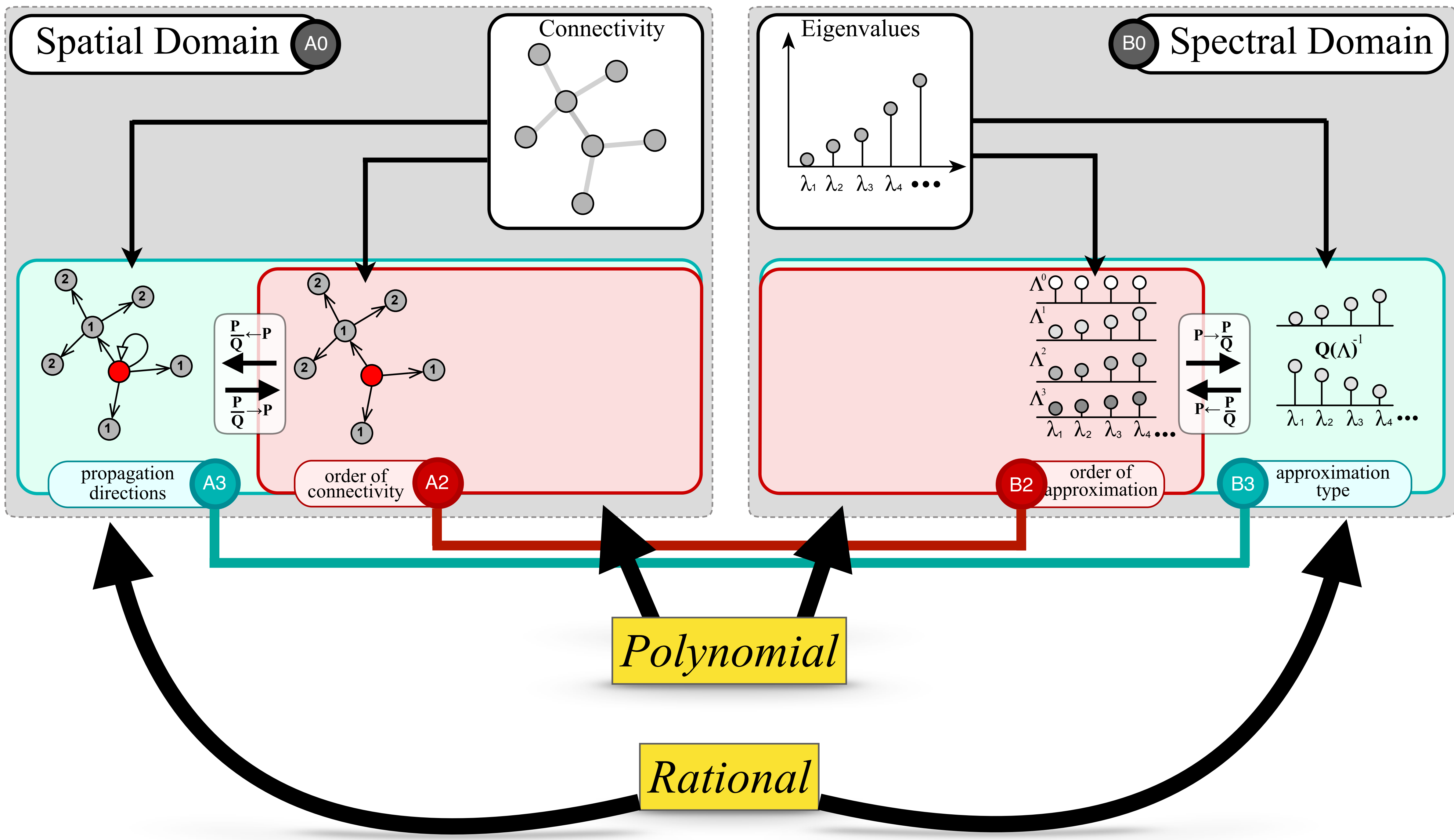
ARMA Filter *Filippo Maria Bianchi et al. (2018)*

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

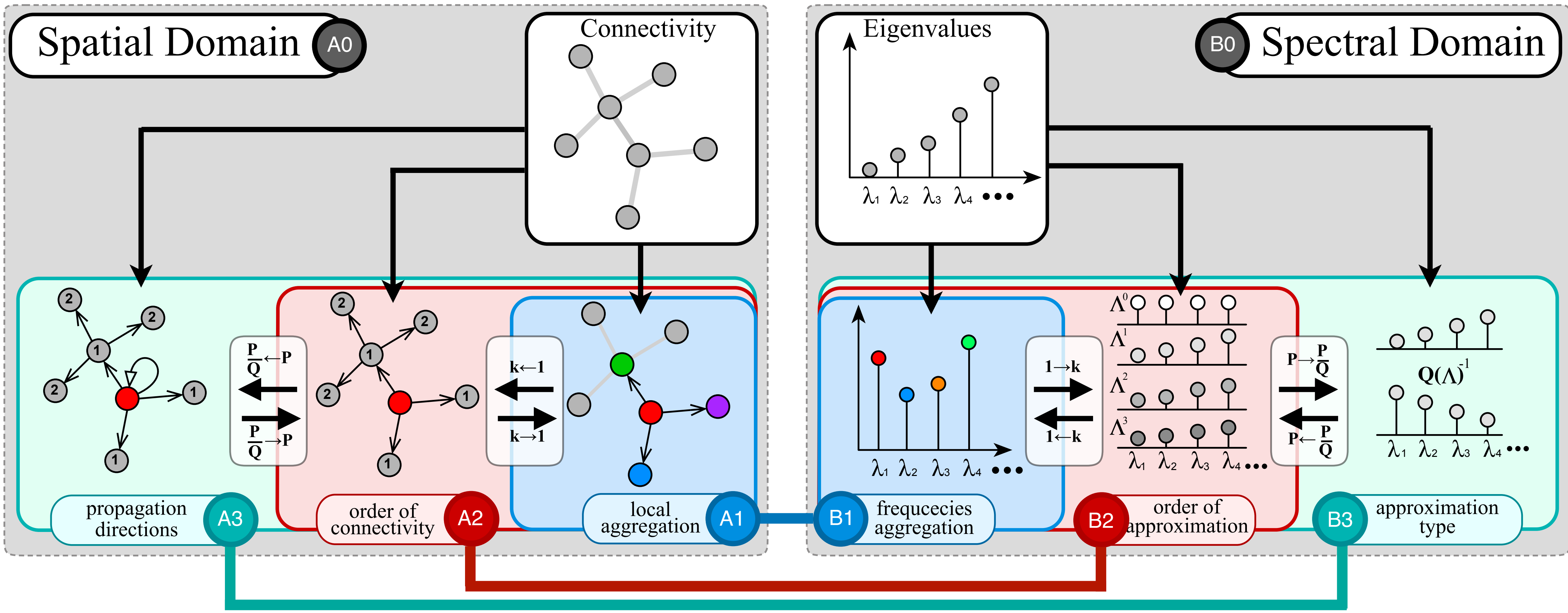
Auto Regressive Filter *Qimai Li et al. (2019)*

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

Rational v.s. Polynomial



The Unified Framework



	Linear	Polynomial	Rational
Spatial			
Spectral			

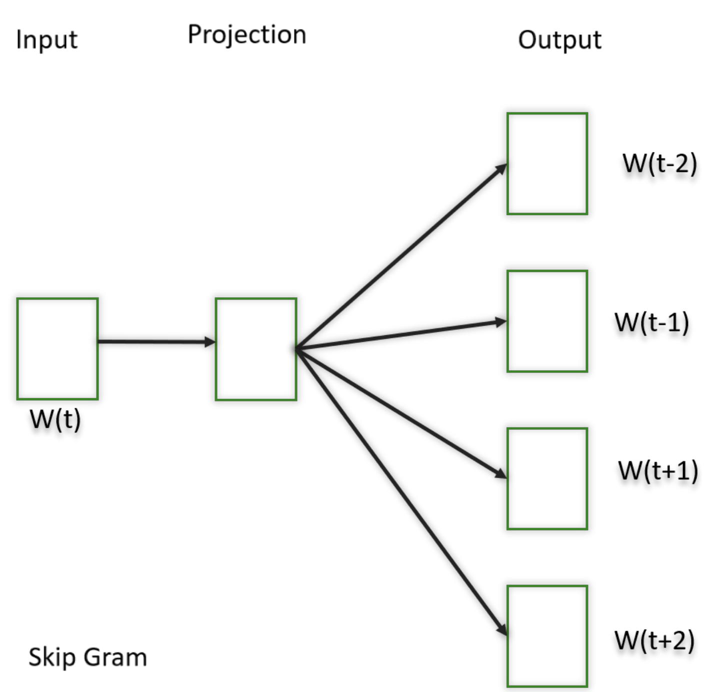
Complex, Powerful

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**

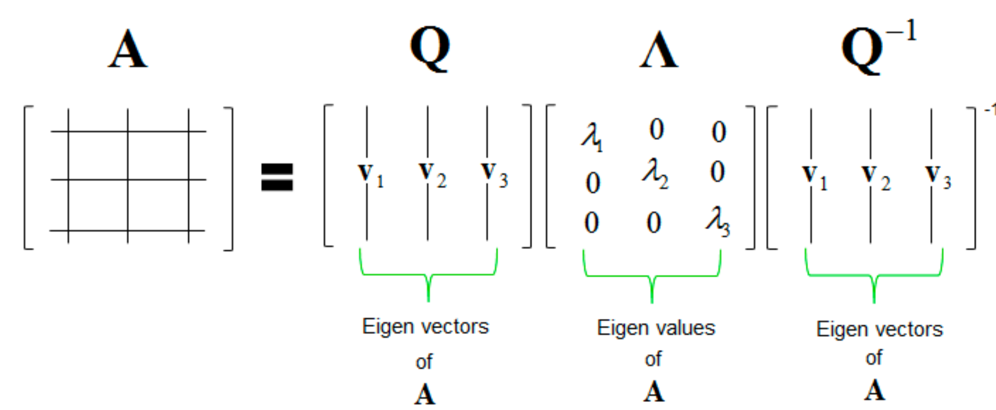
SpectralNet

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 15 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)

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)
  - etc

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

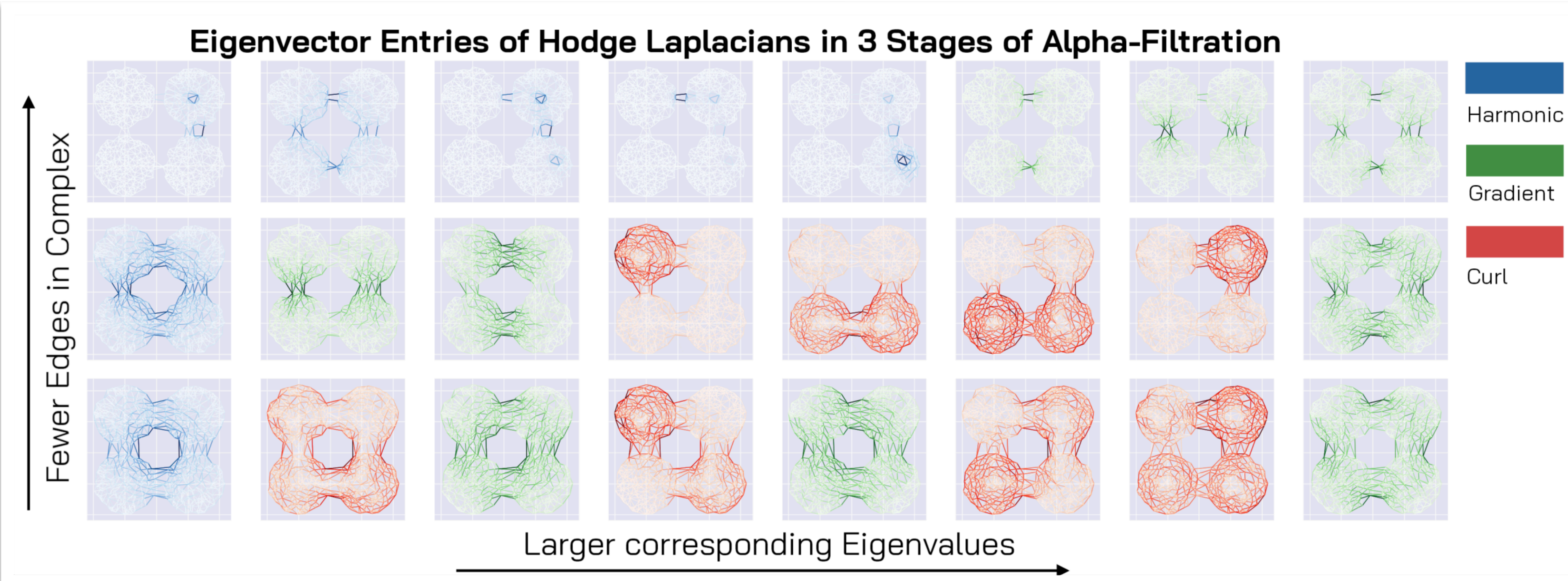
Function of **graph matrix** $g(A)X$

- 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

- Survey Papers
- Milestone Papers
- Spatial and Spectral Views
- Twin Papers
- Applications
- Code
- Citation

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