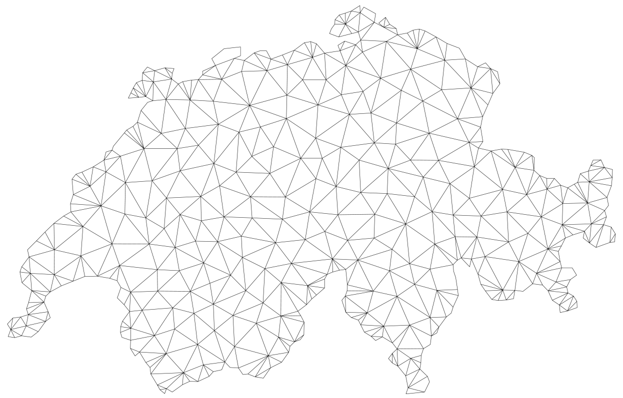
Graph Signal Processing for Machine Learning
A Review and New Perspectives

Xiaowen Dong, Dorina Thanou, Laura Toni,
Michael Bronstein, Pascal Frossard

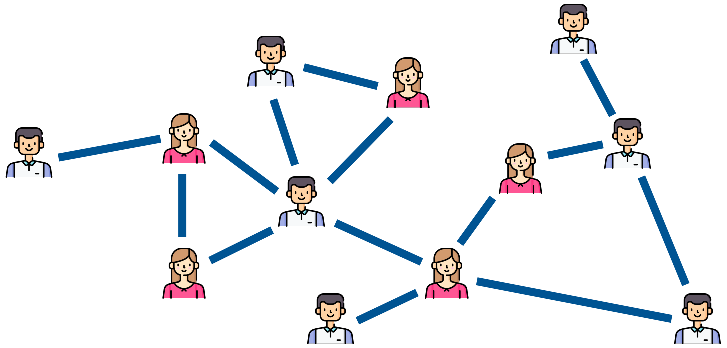
ICASSP Tutorial, June 2021

Networks are pervasive

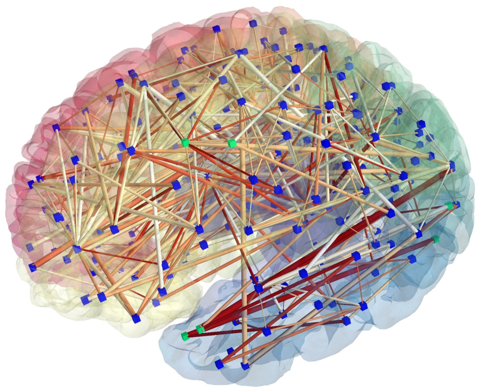


geographical network

traffic network



social network

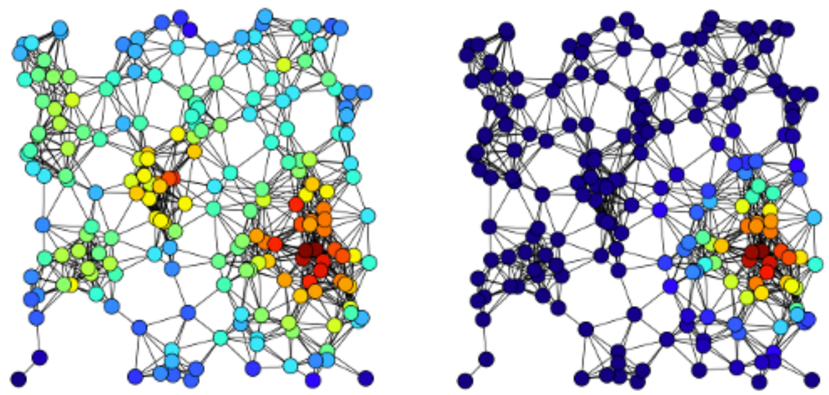


brain network

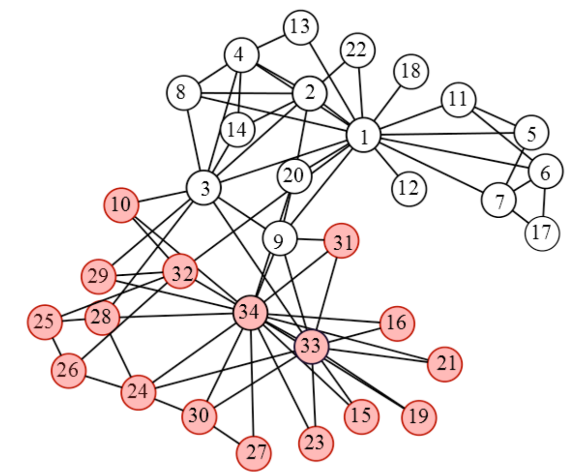
graphs provide mathematical representation of networks

The field of network science

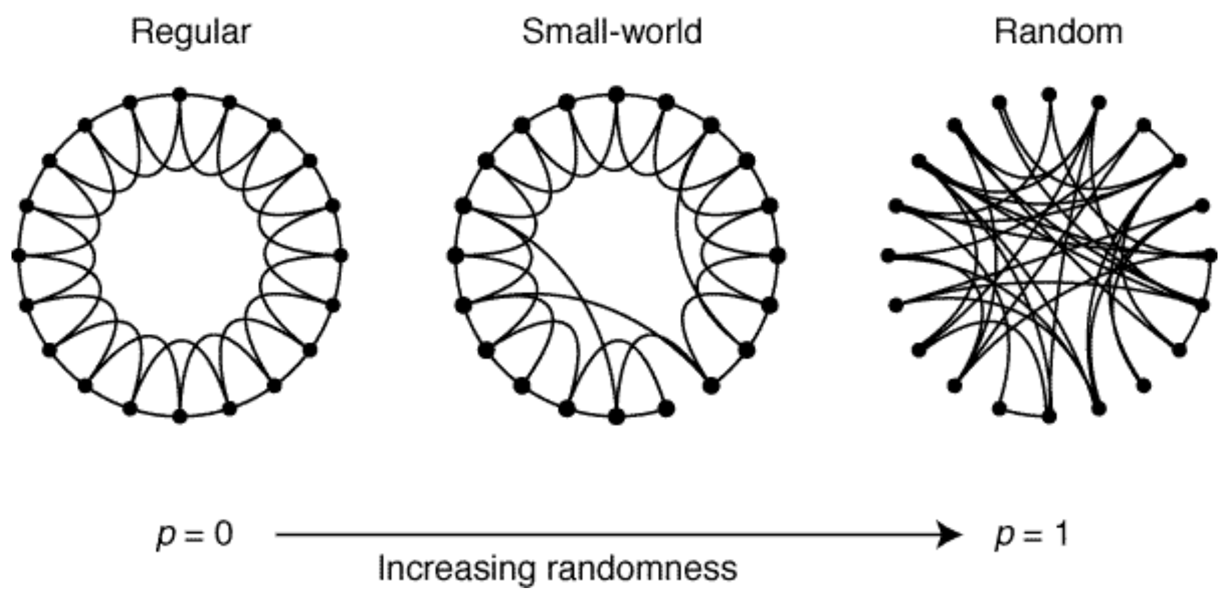
network centrality



community detection



random graph models

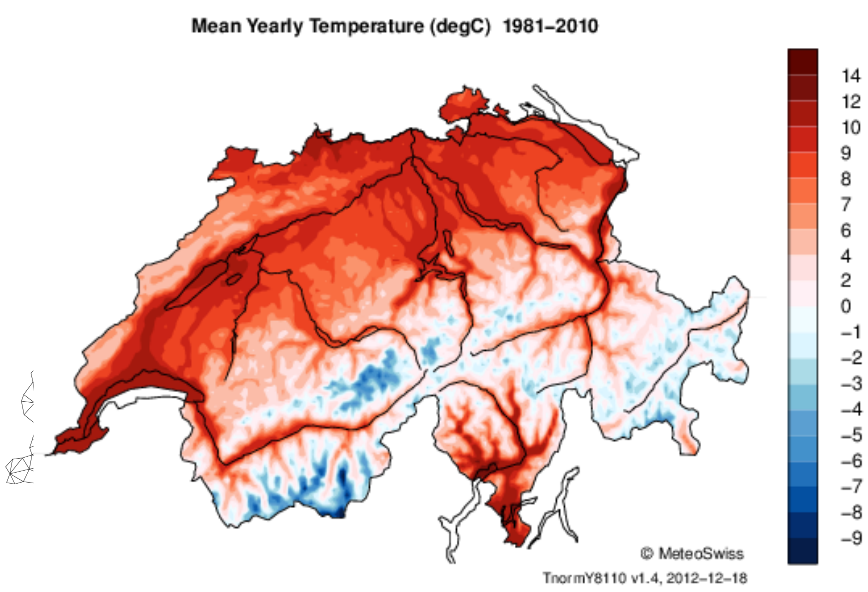


from edge attributes to node attributes

from graphs to graph-structured data

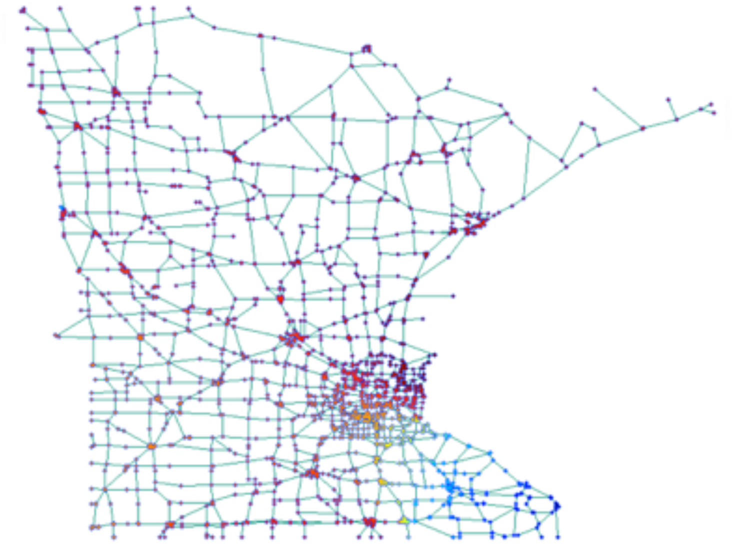
Watts and Strogatz, "Collective dynamics of 'small-world' networks," Nature, 1998.
Newman, "Networks: An introduction," Oxford University Press, 2010.

Graph-structured data are pervasive



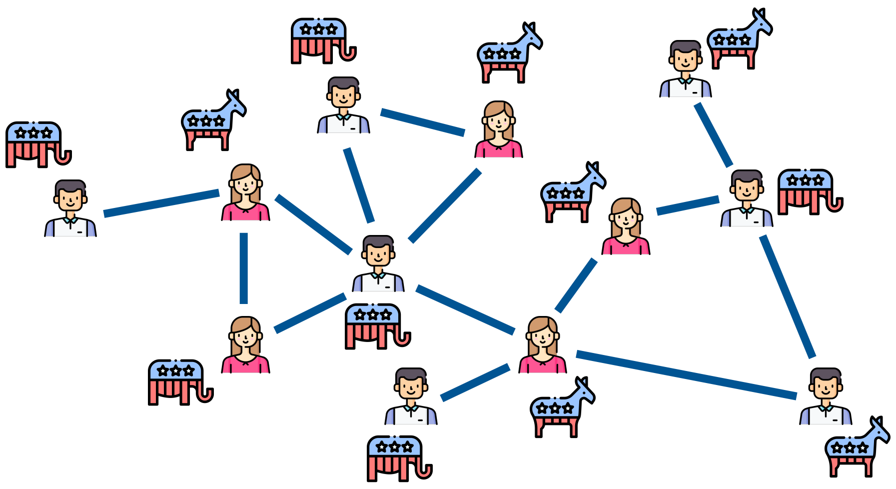
- nodes
  - geographical regions
- edges
  - geographical proximity between regions
- signal
  - temperature records in these regions

Graph-structured data are pervasive



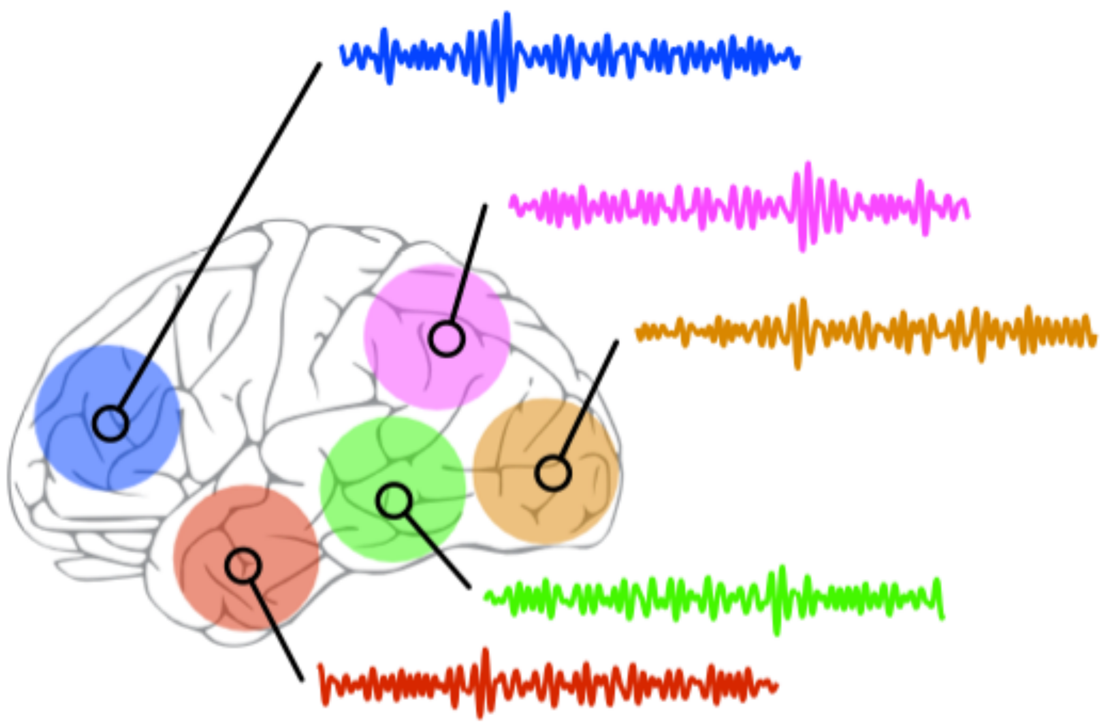
- nodes
  - road junctions
- edges
  - road connections
- signal
  - traffic congestion at junctions

Graph-structured data are pervasive



- nodes
  - individuals
- edges
  - friendship between individuals
- signal
  - political view

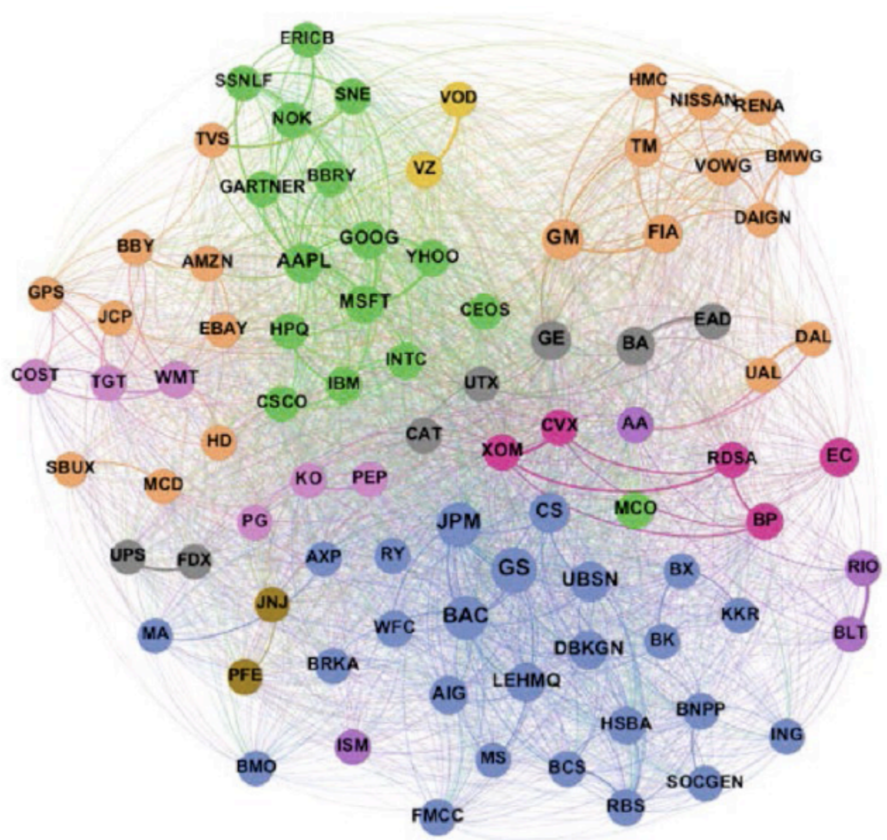
Graph-structured data are pervasive



- nodes
  - brain regions
- edges
  - structural connectivity between brain regions
- signal
  - blood-oxygen-level-dependent (BOLD) time series

Richiardi et al., "Machine learning with brain graphs," IEEE SPM, 2013.

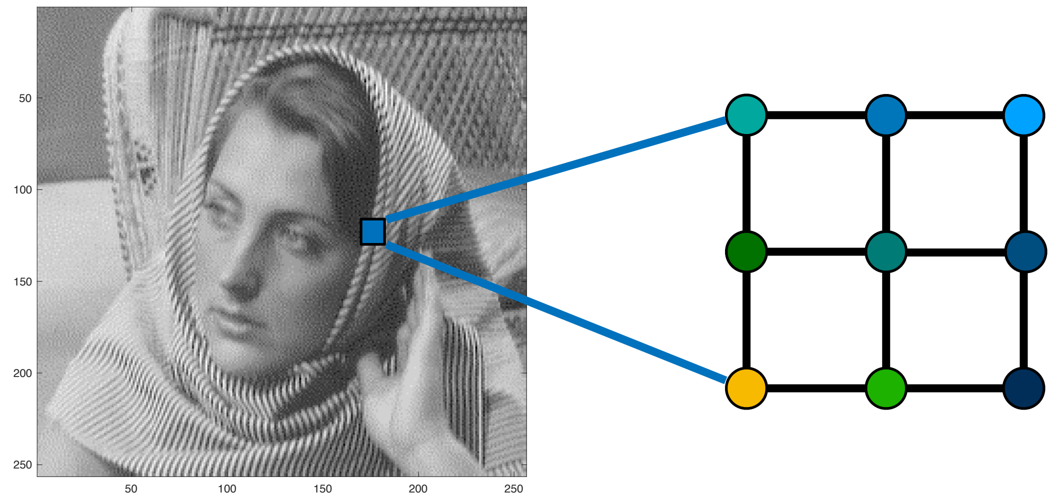
Graph-structured data are pervasive



- nodes
  - companies
- edges
  - co-occurrence of companies in financial news
- signal
  - stock prices of these companies

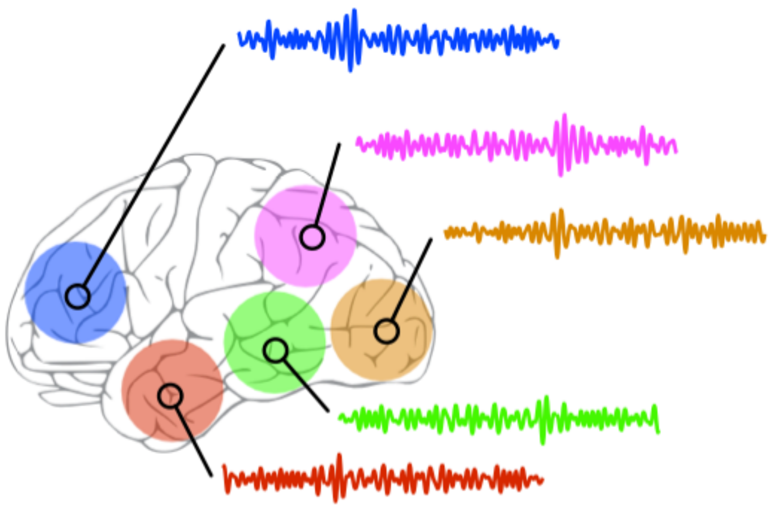
Wan et al., "Sentiment correlation in financial news networks and associated market movements," Scientific Reports, 2021.

Graph-structured data are pervasive



- nodes
  - pixels
- edges
  - spatial proximity between pixels
- signal
  - pixel values

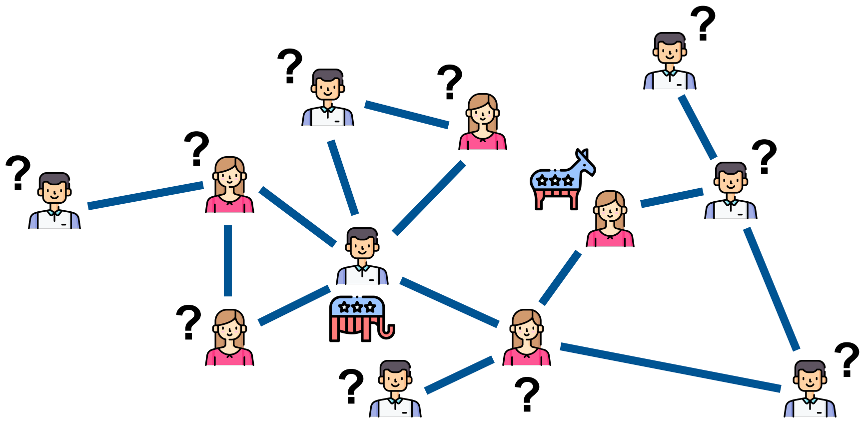
Learning with graph-structured data



condition?
no condition?

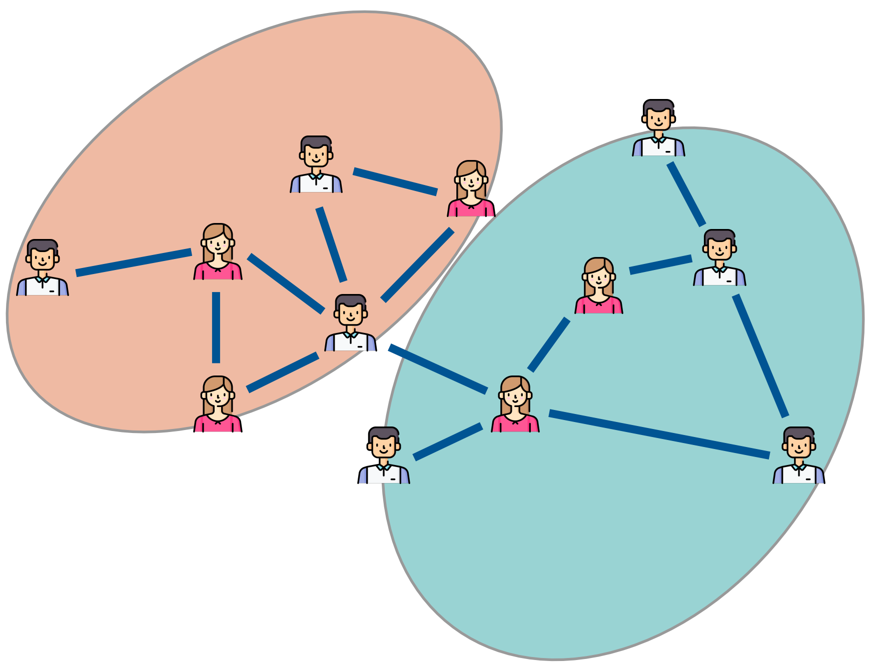
(supervised) graph-level classification

Learning with graph-structured data



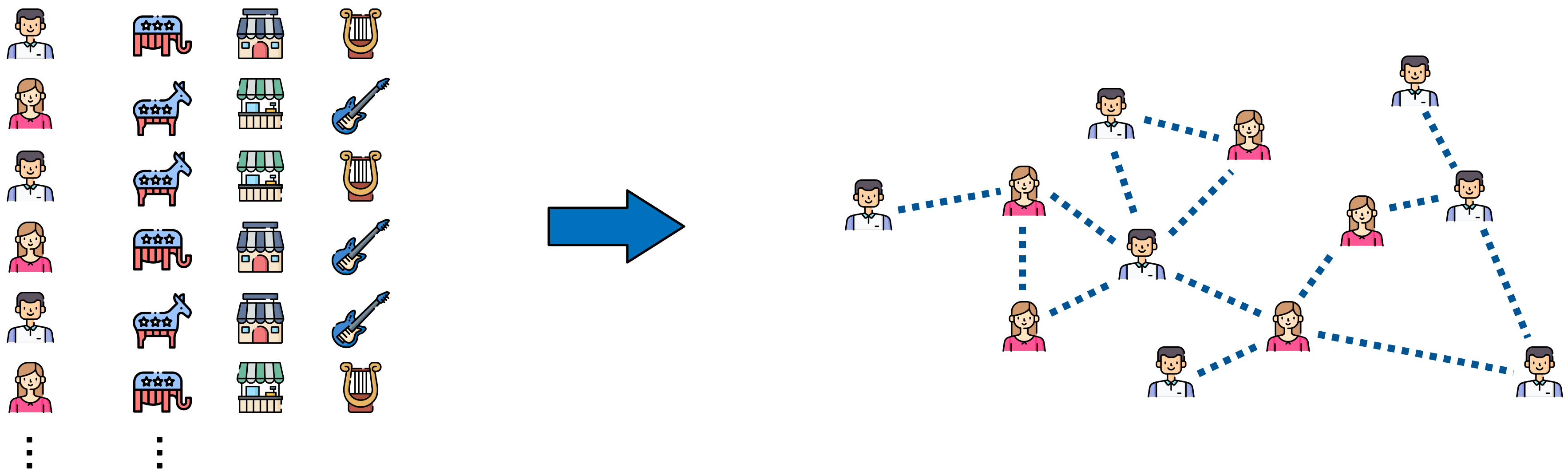
(semi-supervised) node-wise classification

Learning with graph-structured data



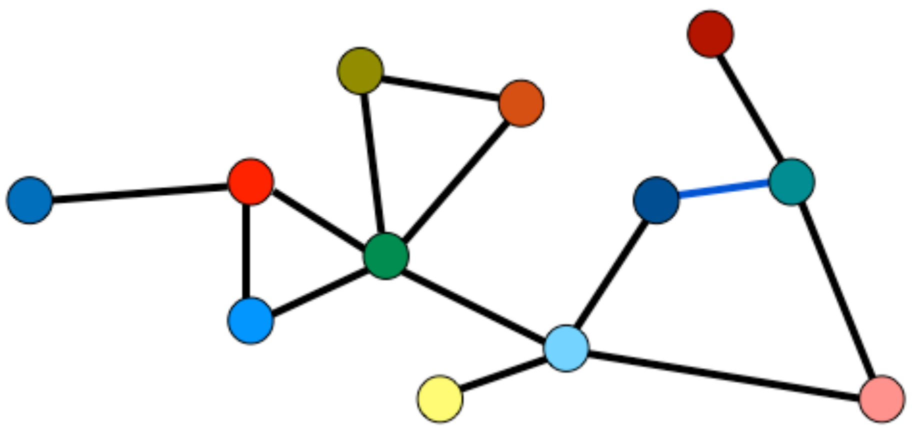
(unsupervised) clustering

Learning with graph-structured data



inferring graph topology from data

Graph-based machine learning

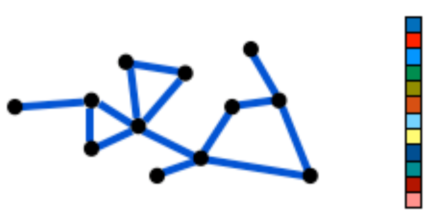


This tutorial:

- graph-structured data are **graph signals**
- how **graph signal processing** brings unique contribution to (graph-based) ML?

Graph-based ML

$f(\mathcal{G}, \mathbf{x})$



$\mathcal{G}$ $\mathbf{x}$

Tasks

Supervised learning

Unsupervised learning

Reinforcement learning

Challenges

Exploiting structure

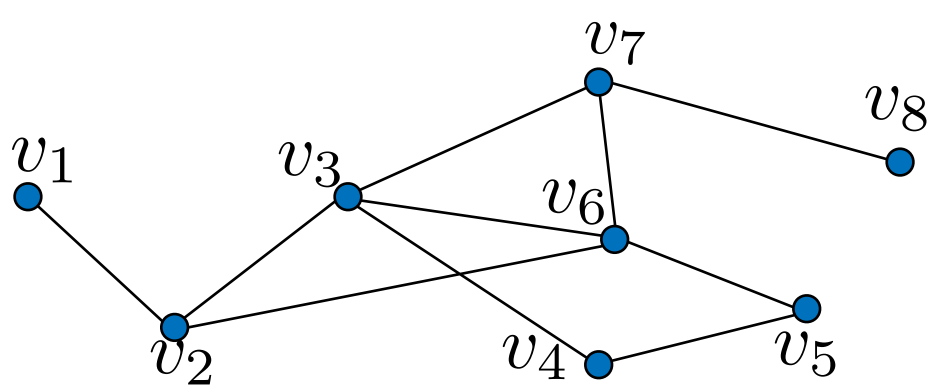
Efficiency & Robustness

Model Interpretability

Outline

- Brief introduction to graph signal processing (GSP)
- Challenge I: GSP for exploiting data structure
- Challenge II: GSP for improving efficiency and robustness
- Challenge III: GSP for enhancing model interpretability
- Applications
- Summary, open challenges, and new perspectives

Graphs and graph Laplacian



weighted and undirected graph:

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$$

$$D = \mathrm{diag}(d(v_1), \cdots, d(v_N))$$

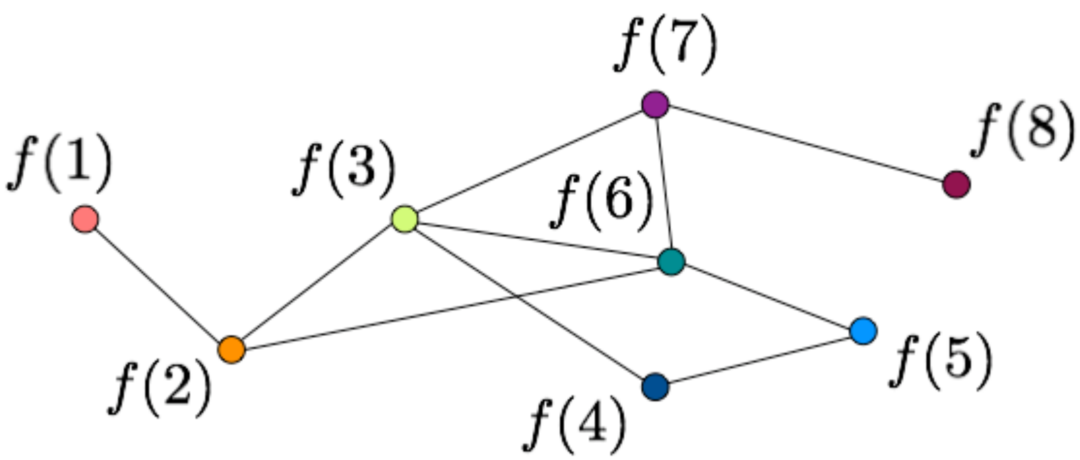
$$L = D - W$$ **equivalent to G!**

$$L_{\mathrm{norm}} = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{D} - \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}}_{W} = \underbrace{\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}}_{L}$$

- symmetric
- off-diagonal entries non-positive
- rows sum up to zero

Graphs and graph Laplacian



graph signal $f : \mathcal{V} \rightarrow \mathbb{R}$

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}$$

$$Lf(i) = \sum_{j=1}^{N} W_{ij}(f(i) - f(j))$$

$$\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}^T \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}$$

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

a measure of "smoothness"

Zhou and Schölkopf, "A regularization framework for learning from graph data," ICML Workshop, 2004.

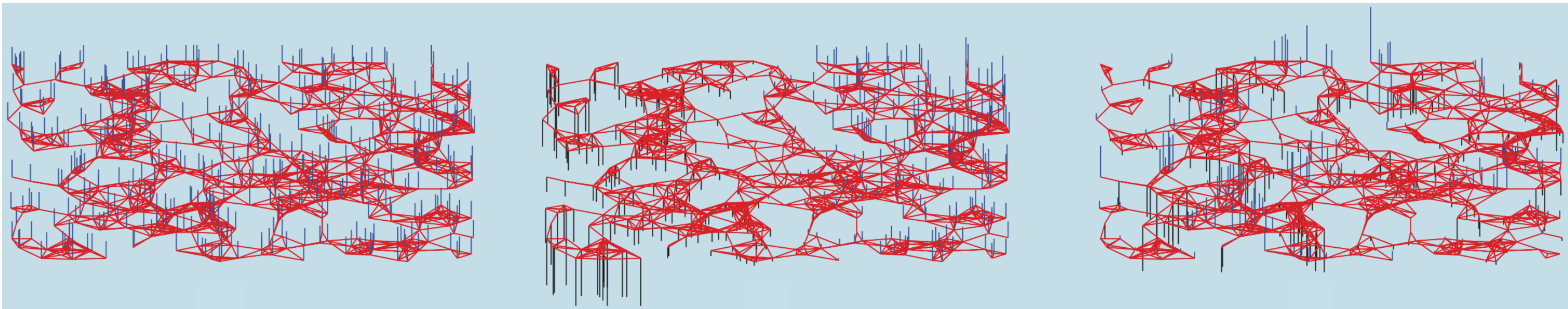
Graphs and graph Laplacian

- L has a complete set of orthonormal eigenvectors: $L = \chi \Lambda \chi^T$

$$L = \underbrace{\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}}_{\chi} \underbrace{\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} \text{---} & \chi_0^T & \text{---} \\ & \cdots & \\ \text{---} & \chi_{N\text{-}1}^T & \text{---} \end{bmatrix}}_{\chi^T}$$

- Eigenvalues are usually sorted increasingly: $0 = \lambda_0 < \lambda_1 \leq \ldots \leq \lambda_{N-1}$

Graph Fourier transform



χ_0 χ_1 χ_{50}

low frequency → high frequency

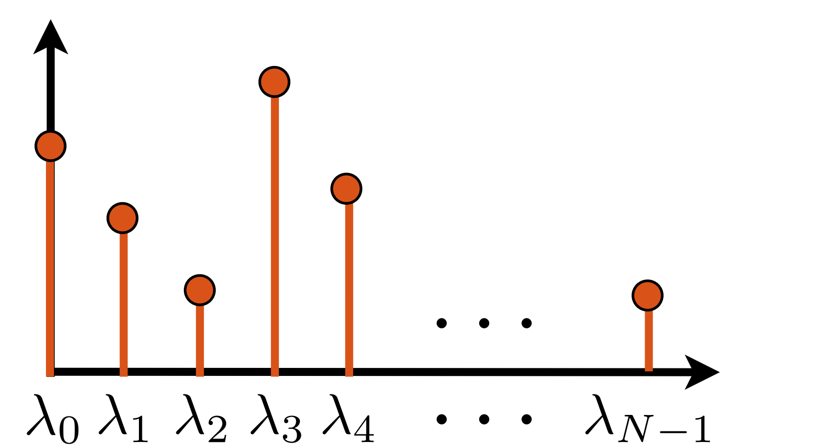
$L = \chi \Lambda \chi^T$

$\chi_0^T L \chi_0 = \lambda_0 = 0$

$\chi_{50}^T L \chi_{50} = \lambda_{50}$

graph Fourier transform:

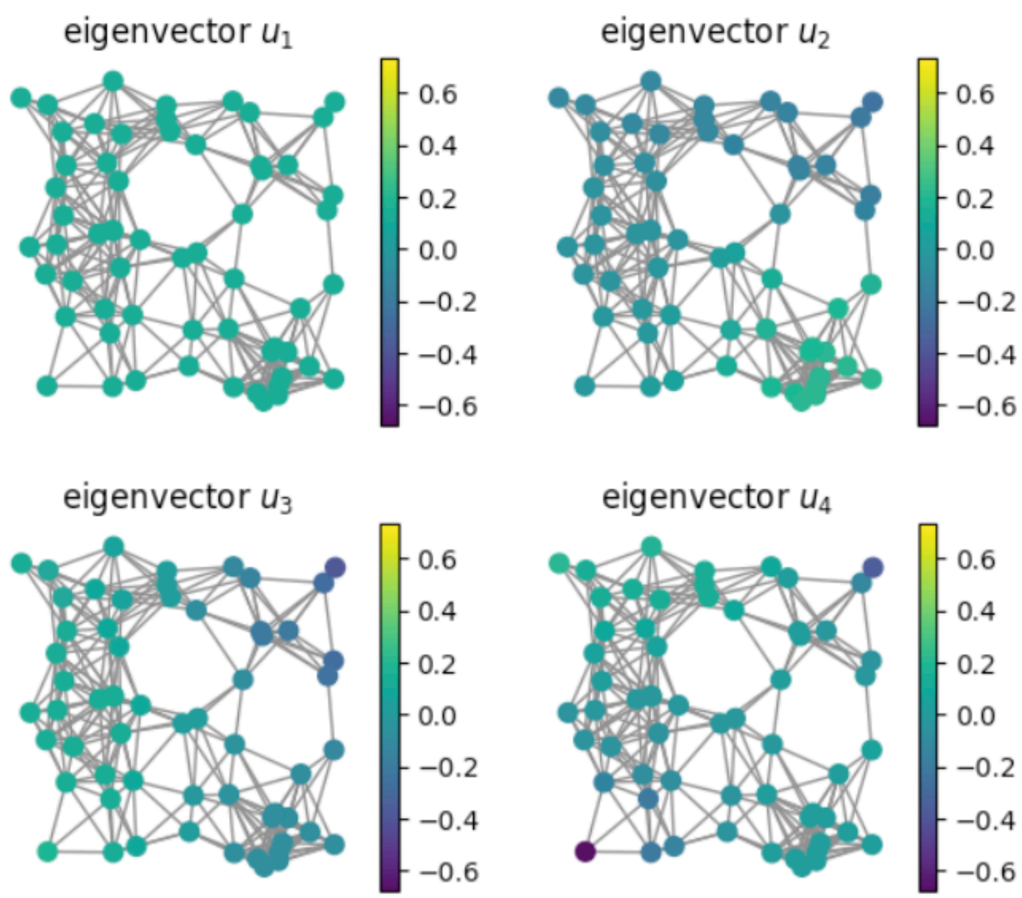
$$\hat{f}(\ell) = \langle \chi_\ell, f \rangle : \quad \begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}^T \begin{matrix} | \\ f \\ | \end{matrix}$$

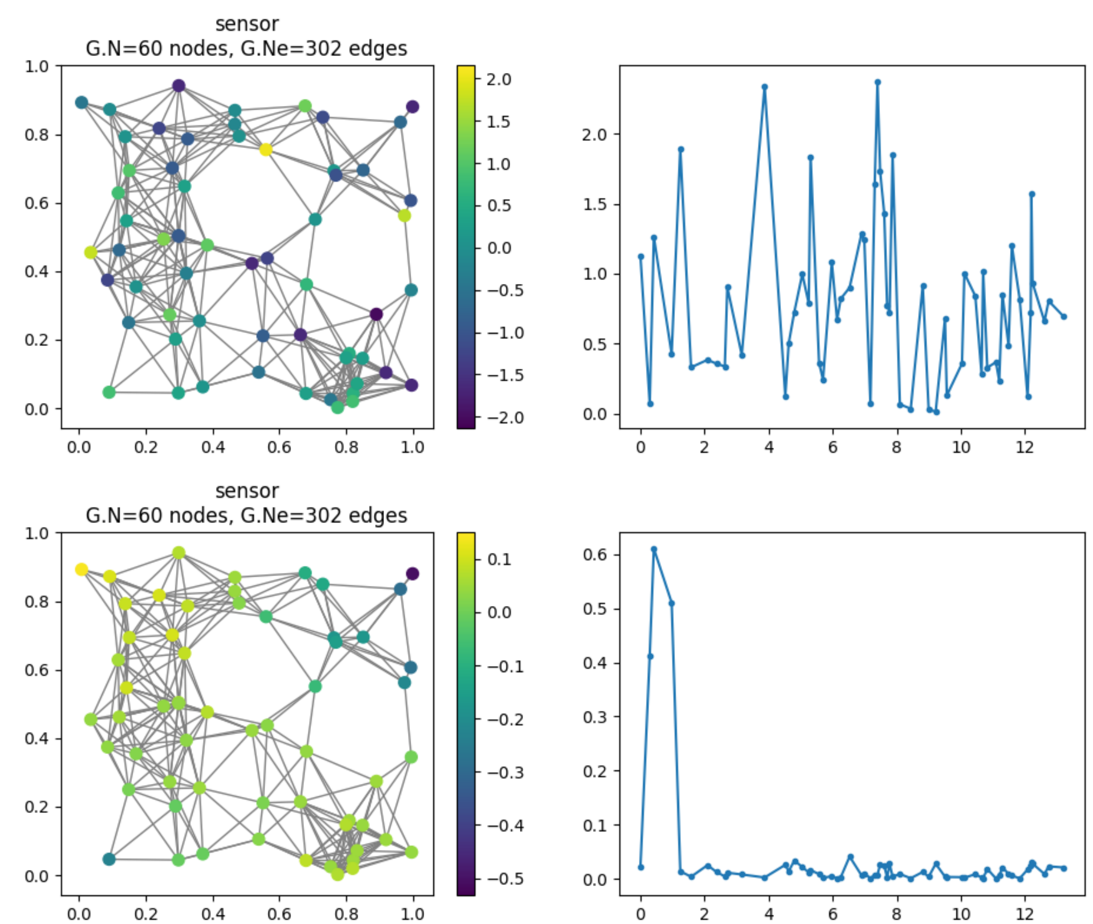


Graph Fourier transform

- Graph Fourier transform

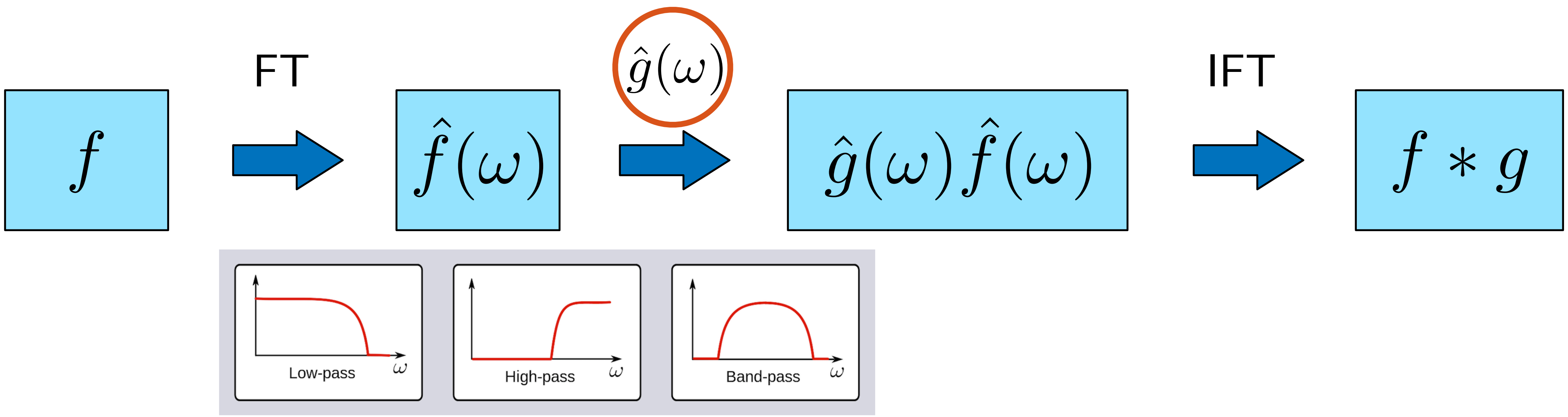
$$\hat{f}(\ell) = \langle \chi_\ell, f \rangle : \begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}^T \begin{matrix} | \\ f \\ | \end{matrix}$$





Classical frequency filtering

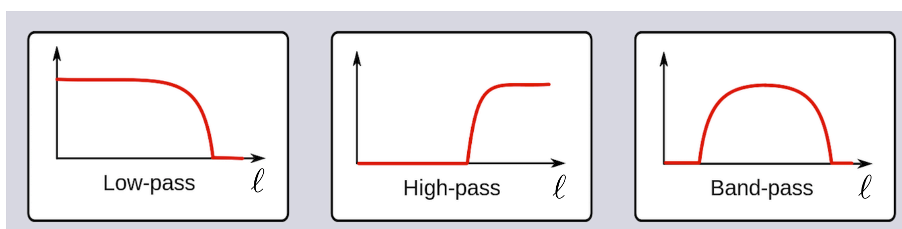
Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$

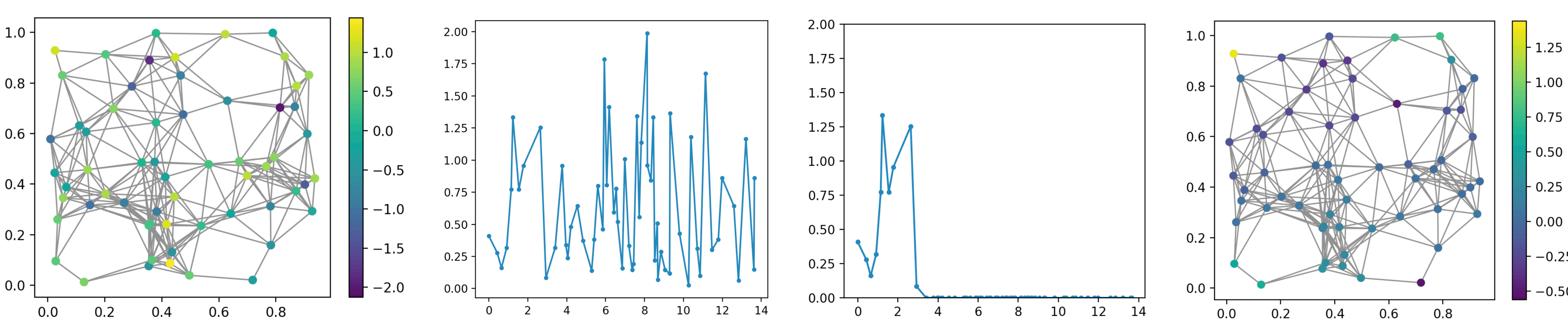


Graph spectral filtering

$$\text{GFT:}\quad \hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$$

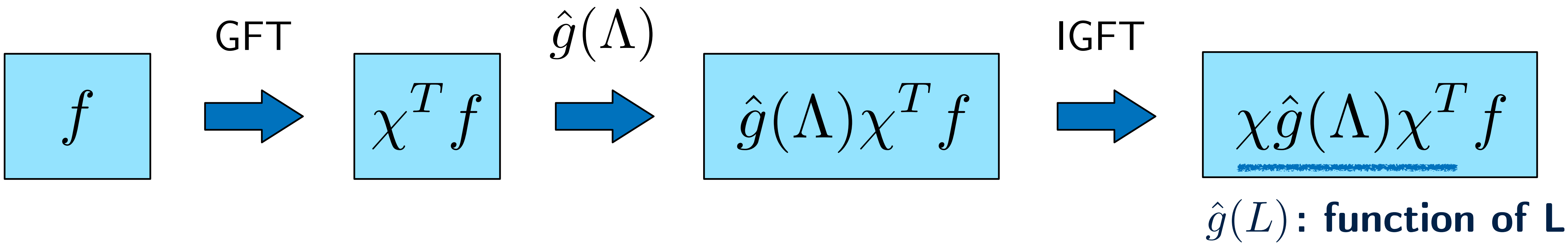
f → GFT → $\hat{f}(\ell)$ → $\hat{g}(\lambda_\ell)$ → $\hat{g}(\lambda_\ell)\hat{f}(\ell)$ → IGFT → $f(i) = \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\hat{f}(\ell)\chi_\ell(i)$





Graph transform/dictionary design

- Transforms and dictionaries can be designed through graph spectral filtering: Functions of graph Laplacian!

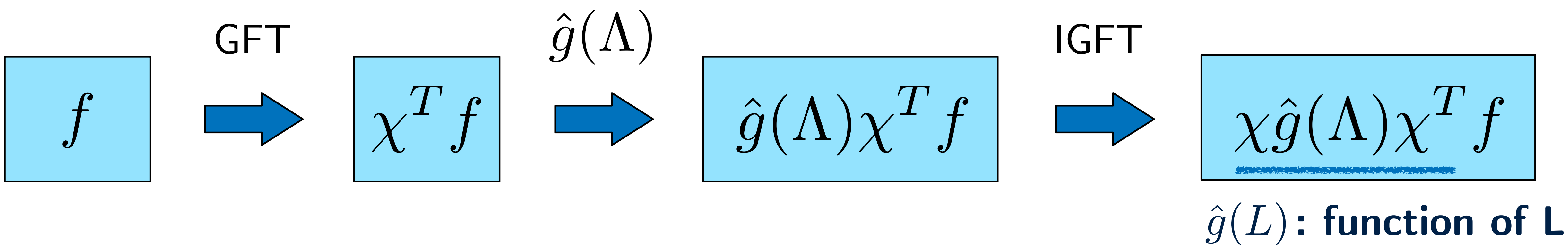


- Important properties can be achieved by properly defining $\hat{g}(L)$, such as localisation of atoms (more on this later)

- Closely related to kernels and regularisation on graphs

Smola and Kondor, "Kernels and regularization on graphs," COLT, 2003.

A practical example



problem: we observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

$$y^* = \arg\min_y \{||y - f||_2^2 + \gamma y^T L y\}$$

data fitting term

"smoothness" assumption

$$y^* = \underbrace{(I + \gamma L)^{-1}}_{\hat{g}(L)} f = \chi(1 + \gamma\Lambda)^{-1}\chi^T f$$

remove noise by low-pass filtering in graph spectral domain!

Graph transform/dictionary design

f $\xrightarrow{\text{GFT}}$ $\chi^T f$ $\xrightarrow{\hat{g}(\Lambda)}$ $\hat{g}(\Lambda)\chi^T f$ $\xrightarrow{\text{IGFT}}$ $\chi\hat{g}(\Lambda)\chi^T f$

$\hat{g}(L)$ **: function of L!**

smoothing/low-pass filtering: $\hat{g}(L) = (I + \gamma L)^{-1} = \chi(I + \gamma\Lambda)^{-1}\chi^T$

Graph-based regularisation

windowed kernel: windowed graph Fourier transform

shifted and dilated band-pass filters: spectral graph wavelets $\hat{g}(sL)$

Graph filters & transforms

adapted kernels: learn values of $\hat{g}(L)$ directly from data

parametric kernel: $\hat{g}(L) = \sum_{k=0}^{K} \theta_j L^k = \chi(\sum_{k=0}^{K} \theta_j \Lambda^k)\chi^T$

Learning models on graphs

Shuman et al., "Vertex-frequency analysis on graphs," ACHA, 2016.

GSP for machine learning

	Graph-based regularisation	**Graph filters & transforms**	**GSP-related learning models**

Exploiting structure

Efficiency & Robustness

Model Interpretability

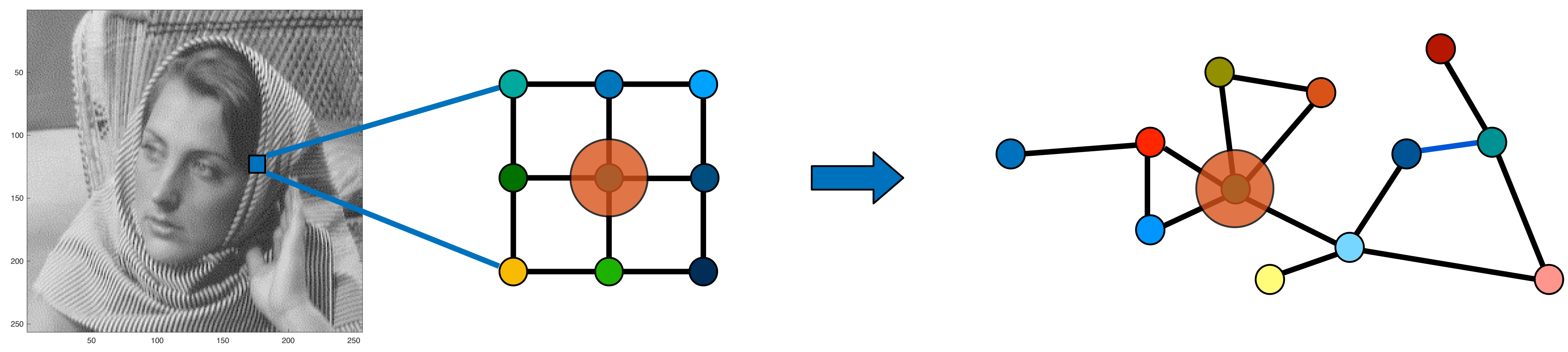
- enable convolution & hierarchical modelling on graphs
- improve efficiency & robustness of (graph-based) ML models
- interpret data structure & learning models on graphs

Dong et al., "Graph signal processing for machine learning," IEEE SPM, 2020.

Outline

- Brief introduction to graph signal processing (GSP)
- Challenge I: GSP for exploiting data structure
- Challenge II: GSP for improving efficiency and robustness
- Challenge III: GSP for enhancing model interpretability
- Applications
- Summary, open challenges, and new perspectives

GSP for exploiting data structure



- GSP enables definition of graph convolution
- GSP enriches design of graph convolutional models
- GSP facilitates hierarchical modelling on graphs

GSP for defining convolution on graphs

Convolution on graphs

classical convolution

convolution on graphs

time domain

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau$$

frequency domain

$$\widehat{(f * g)}(\omega) = \hat{f}(\omega) \cdot \hat{g}(\omega)$$

spatial (node) domain

$$f * g = \chi \hat{g}(\Lambda)\chi^T f = \hat{g}(L) f$$

graph spectral domain

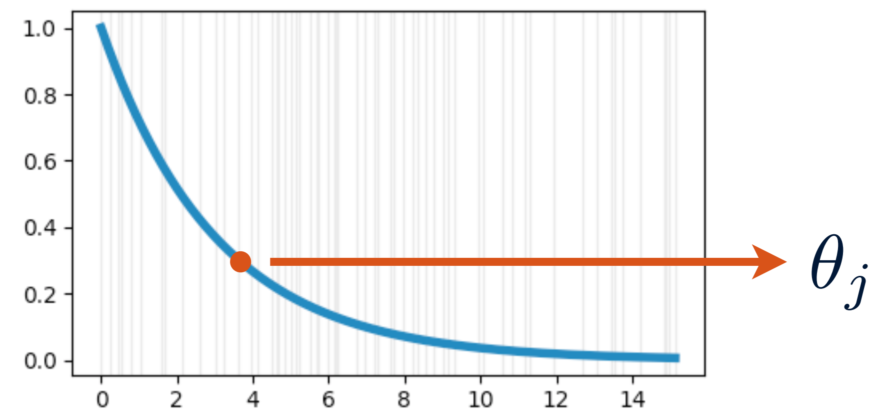
$$\widehat{(f * g)}(\lambda) = \big((\chi^T f) \circ \hat{g}\big)(\lambda)$$

Convolution on graphs

$$f * g = \chi \hat{g}(\Lambda)\chi^T f = \hat{g}(L) f$$

learning a non-parametric filter:

$$\hat{g}_\theta(\Lambda) = \mathrm{diag}(\theta),\ \theta \in \mathbb{R}^N$$



- convolution expressed in the graph spectral domain
- no localisation in the spatial (node) domain

Bruna et al., "Spectral networks and deep locally connected networks on graphs," ICLR, 2014.

Convolution on graphs

$$f * g = \chi \hat{g}(\Lambda) \chi^T f = \hat{g}(L) f$$

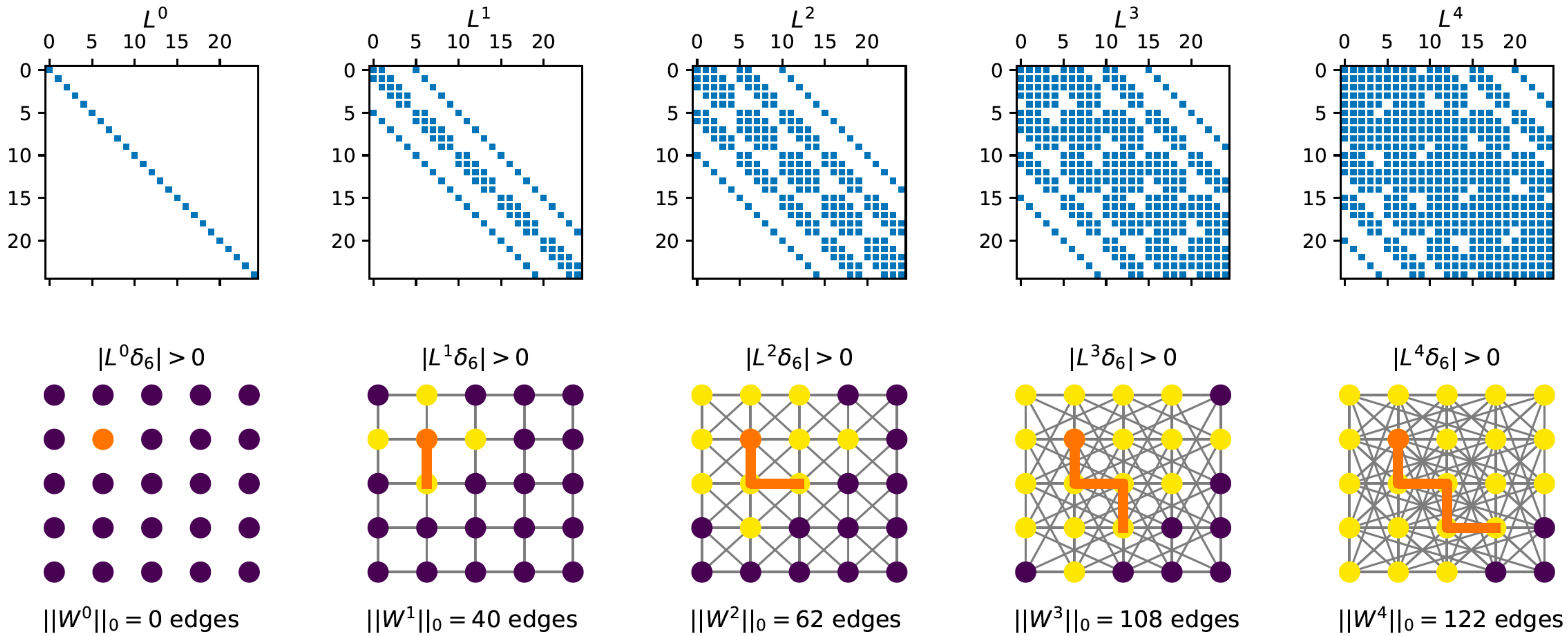
parametric filter as polynomial of Laplacian

$$\hat{g}_\theta(\lambda) = \sum_{j=0}^{K} \theta_j \lambda^j, \ \theta \in \mathbb{R}^{K+1} \quad \Rightarrow \quad \hat{g}_\theta(L) = \sum_{j=0}^{K} \theta_j L^j$$

what do powers of graph Laplacian capture?

Powers of graph Laplacian

L^k defines the k-neighborhood



Localization: $d_{\mathcal{G}}(v_i, v_j) > K$ implies $(L^K)_{ij} = 0$

(slide by Michaël Deferrard)

Defferrard et al., "Convolutional neural networks on graphs with fast localized spectral filtering," NIPS, 2016.

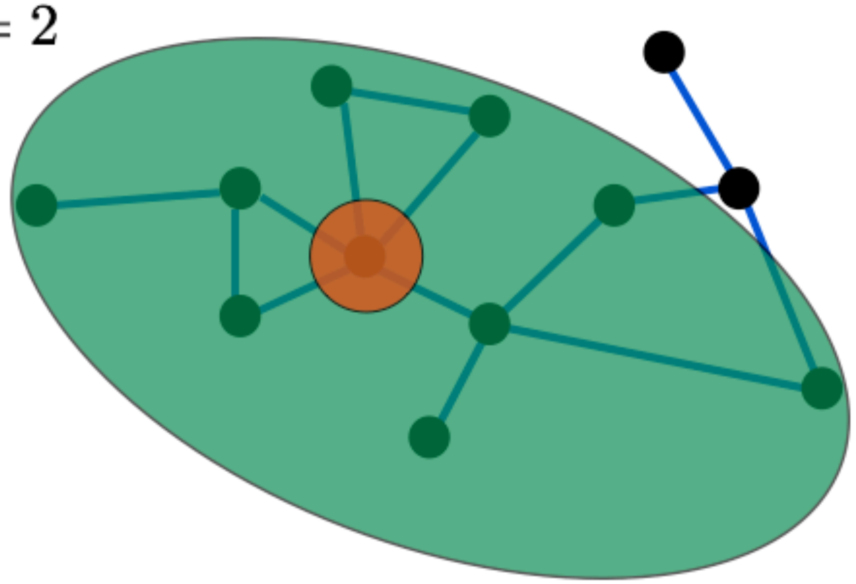
Convolution on graphs

$$f * g = \chi \hat{g}(\Lambda) \chi^T f = \hat{g}(L) f$$

parametric filter as polynomial of Laplacian

$$\hat{g}_\theta(\lambda) = \sum_{j=0}^{K} \theta_j \lambda^j, \; \theta \in \mathbb{R}^{K+1} \quad \Rightarrow \quad \hat{g}_\theta(L) = \sum_{j=0}^{K} \theta_j L^j \Rightarrow \sum_{j=0}^{K} \theta_j T_j(\tilde{L})$$

$K = 2$



- convolution is expressed in the graph spectral domain
- localisation within *K*-hop neighbourhood
- Chebyshev approximation using $\tilde{L} = 2L/\lambda_{N-1} - I$

Defferrard et al., "Convolutional neural networks on graphs with fast localized spectral filtering," NIPS, 2016.

Convolution on graphs

$$f * g = \chi \hat{g}(\Lambda) \chi^T f = \hat{g}(L) f$$

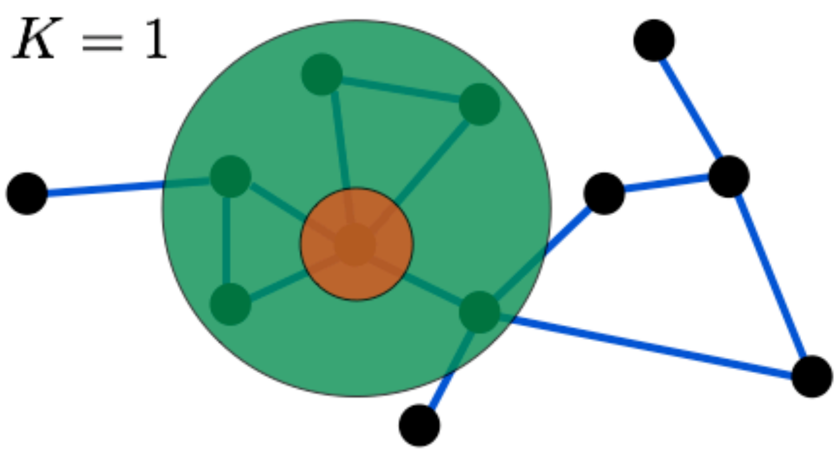
simplified polynomial

$$\hat{g}_\theta(L) = \sum_{j=0}^{K} \theta_j L^j \Rightarrow \sum_{j=0}^{K} \theta_j T_j(\tilde{L})$$

$K = 1$, $\lambda_{N-1} \approx 2$

$$= \theta_0 I - \theta_1 (D^{-\frac{1}{2}} W D^{-\frac{1}{2}})$$

(localisation within 1-hop neighbourhood)



$\alpha = \theta_0 = -\theta_1$

$$= \alpha (I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}})$$

renormalisation

$$\Rightarrow \alpha (\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}})$$

Kipf and Welling, "Semi-supervised classification with graph convolutional networks," ICLR, 2017.

Convolution on graphs - Remarks

- Convolution is defined via the **graph spectral** domain...

$$f * g = \chi \hat{g}(\Lambda) \chi^T f = \hat{g}(L) f$$

- ...but can be implemented in the **spatial (node)** domain

$$y = \hat{g}_\theta(L) f = \alpha(\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}}) f$$

→ simple neighbourhood averaging in Kipf and Welling 2017

Convolution on graphs - Remarks

- Convolution in classical signal processing relies on the shift operator

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-\tau) g(\tau) d\tau$$

- Notion of shift by a graph shift operator (e.g., adjacency/Laplacian matrix)

$$Sf \Rightarrow g(S)f = \sum_{k=0}^{K} \theta_k S^k f$$

  - spatial definition of convolution that resembles an FIR filter (on graphs)
  - motivated from a **spatial** perspective, but has a **spectral** interpretation via eigendecomposition of S

Ortega et al., "Graph signal processing," Proceedings of the IEEE, 2018.
Cheung et al., "Graph signal processing and deep learning," IEEE SPM, 2020.
Gama et al., "Graphs, convolutions, and neural networks," IEEE SPM, 2020.

Convolution on graphs - Remarks

- Convolution can also be interpreted as a weighted summation

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau$$

- Spatial generalisation of convolution in non-Euclidean domain

$$D(x)f = \int_{\mathcal{X}} f(x') u(x, x') dx' \quad \Rightarrow \quad D(v)f = \sum_{\mathcal{V}} f(v') u(v, v')$$

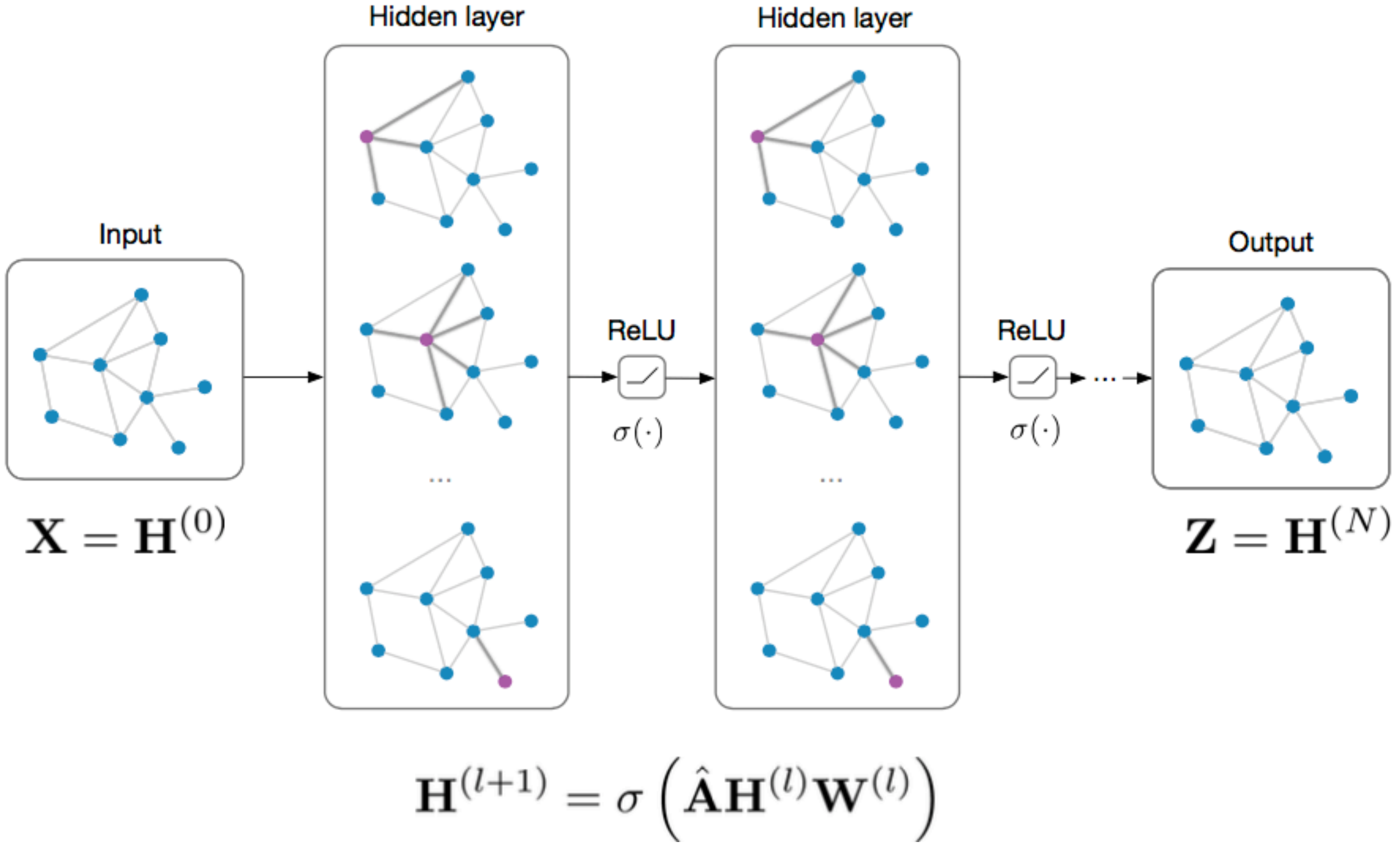
- weighting function $u(v, v')$ determines relative importance of neighbours

Monti et al., "Geometric deep learning on graphs and manifolds using mixture model CNNs," CVPR, 2017.
Walker and Glocker, "Graph convolutional Gaussian processes," ICML, 2019.

Graph convolutional networks

- Convolution on graphs leads to graph convolutional networks (GCNs)...

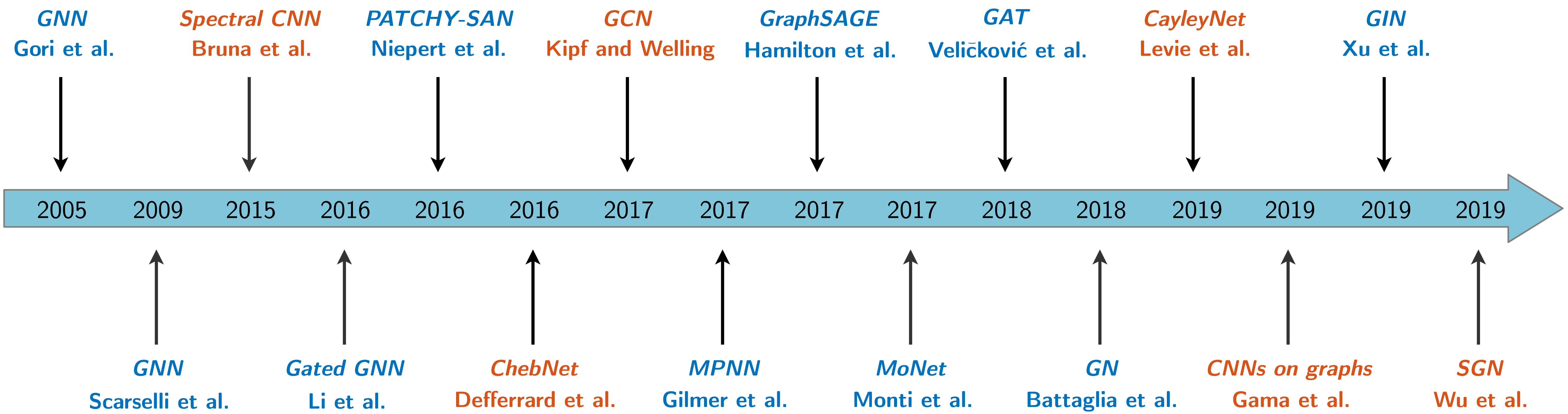
$$\hat{g}_{\theta^{(k+1)}}(L)\Big(\mathrm{ReLU}(\hat{g}_{\theta^{(k)}}(L)f)\Big)$$



Kipf and Welling, "Semi-supervised classification with graph convolutional networks," ICLR, 2017.

Graph neural networks

- ...and more generally graph neural networks (GNNs)

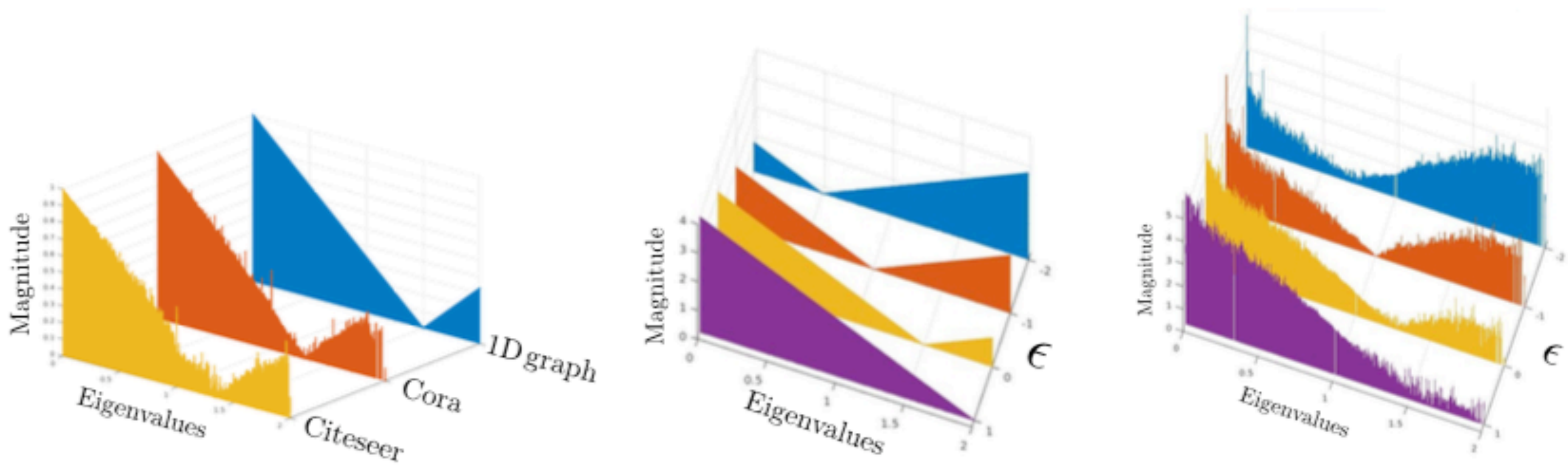


  - **spatial** vs **spectral** designs

Balcilar et al., "Analyzing the expressive power of graph neural networks in a spectral perspective," ICLR, 2021.
Veličković, "Theoretical foundations of graph neural networks," https://petar-v.com/talks/GNN-Wednesday.pdf, 2021.

GSP for enriching graph convolutional models

Expressive power of GNNs



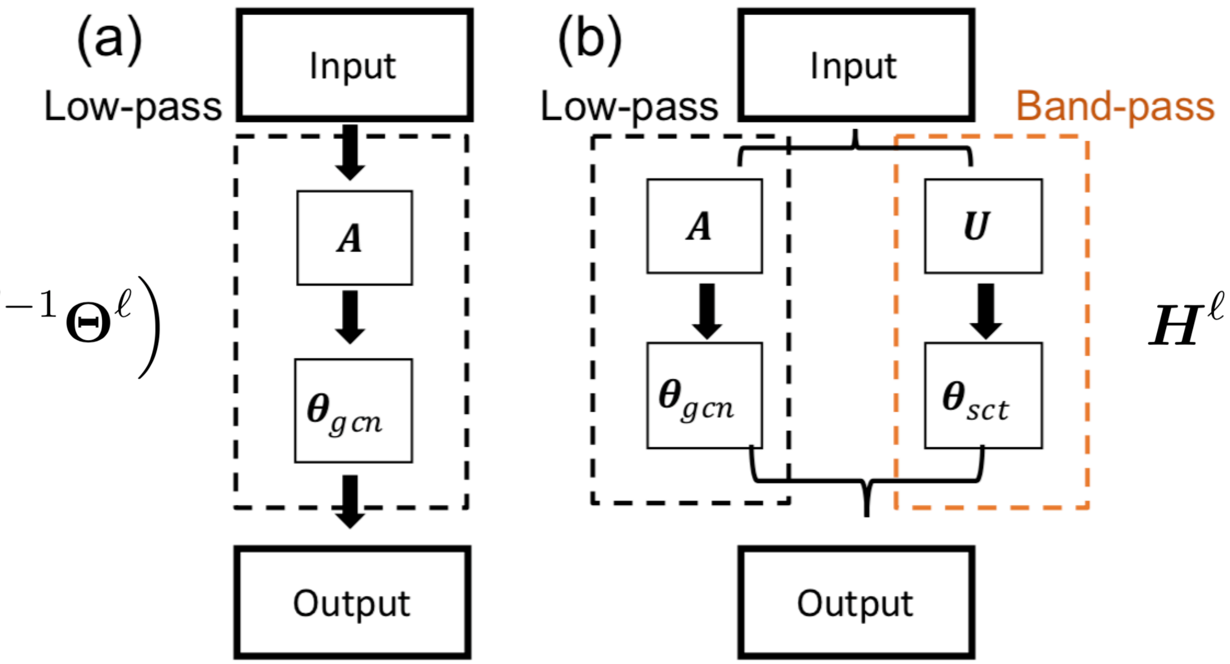
(a) GCN frequency profiles (b) GIN on 1D (c) GIN on CiteSeer

- convolutional layers in various GNN models can be understood as graph filters of different spectral profiles
- focusing on low-frequency information may lead to over-smoothing

Balcilar et al., "Analyzing the expressive power of graph neural networks in a spectral perspective," ICLR, 2021.

Beyond low-frequency information

$$H^{\ell} = \sigma\left(AH^{\ell-1}\Theta^{\ell}\right)$$



GCN

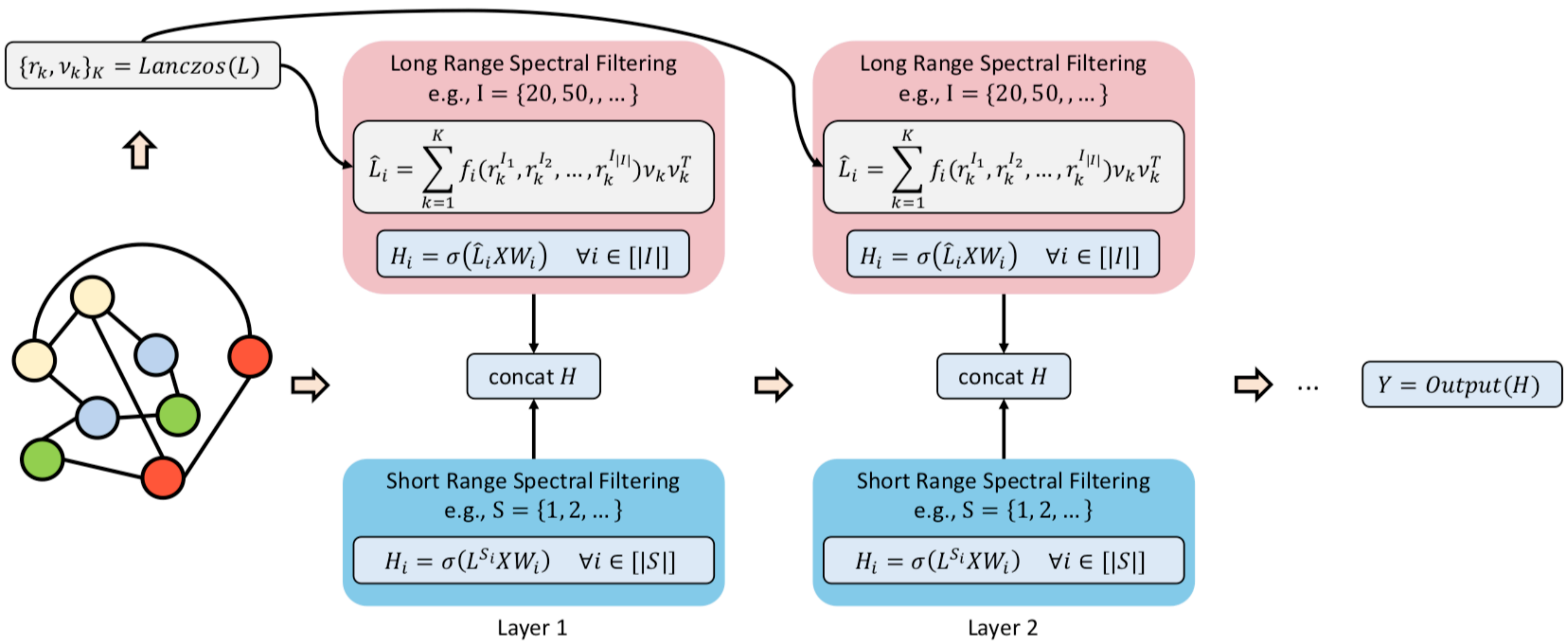
Scattering GCN

$$H^{\ell} = \sigma\left(U_p H^{\ell-1}\Theta^{\ell}\right)$$

- combine low-pass operations based on GCN with band-pass operations based on geometric scattering (Min et al. 2020)
- combine low-pass and high-pass filtering (Bo et al. 2021)

Min et al., "Scattering GCN: Overcoming oversmoothness in graph convolutional networks," NeurIPS, 2020.
Bo et al., "Beyond low-frequency information in graph convolutional networks," AAAI, 2021.

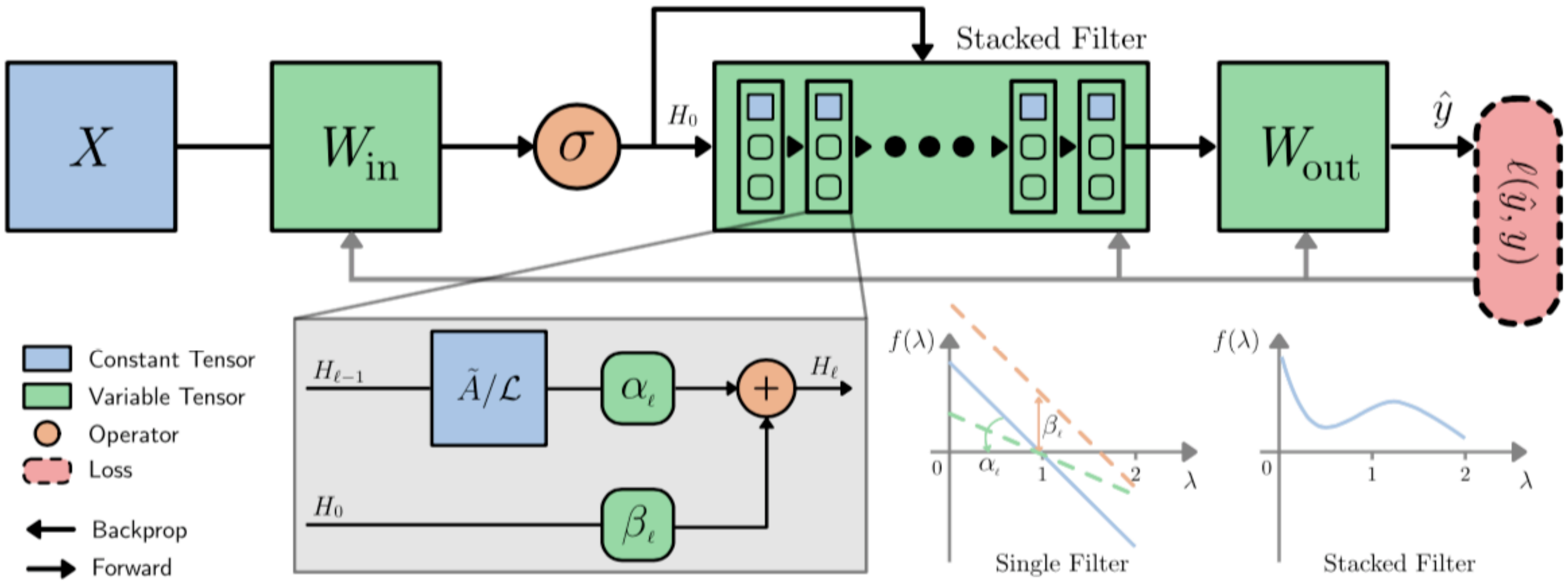
Beyond low-frequency information



- combine short-range and long-range filtering
- long-range filtering facilitated by low-rank approximation to affinity matrix based on Lanczos algorithm
- learnable spectral filters based on the approximation

Liao et al., "LanczosNet: Multi-scale deep graph convolutional networks," ICLR, 2019.

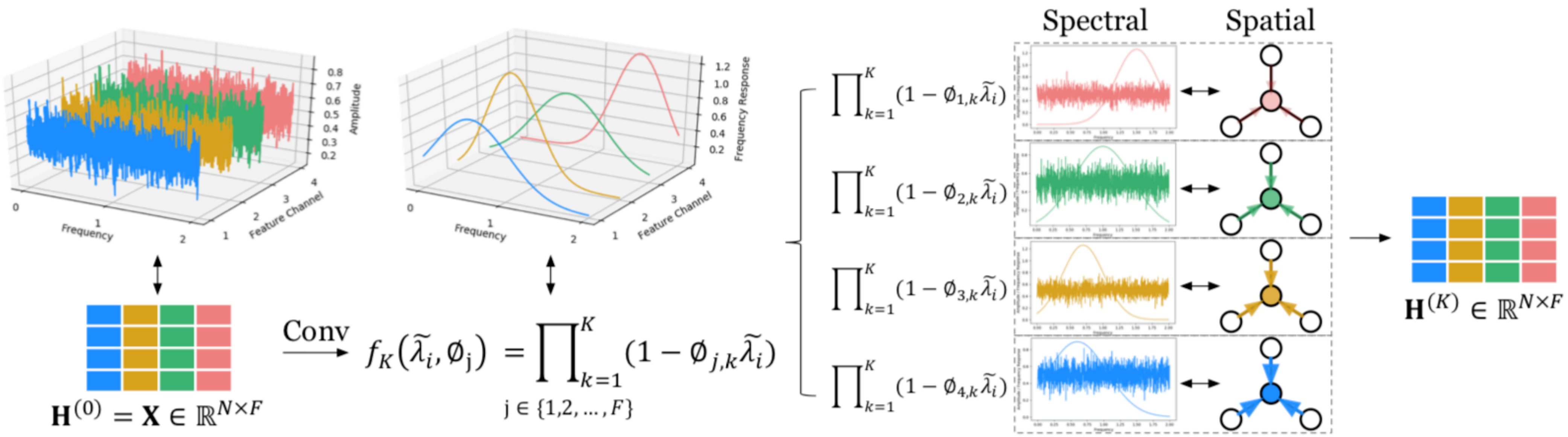
Adaptive filters



- stack graph filters with learnable filter parameters to build highly adaptive model

NT et al., "Adaptive stacked graph filter," https://openreview.net/forum?id=6VPl9khIMz, 2020.

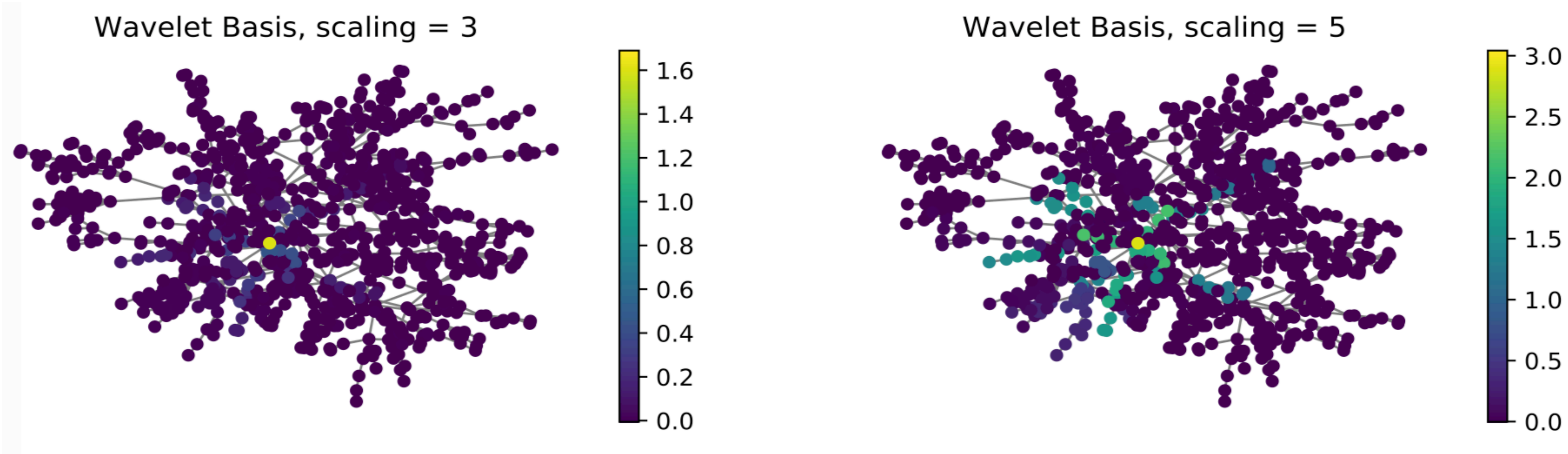
Adaptive filters



- stack learnable filters across network layers (removing nonlinearities at second and subsequence layers)
- learn separate adaptive filter for each feature channel

Dong et al., "Graph neural networks with adaptive frequency response filter," arXiv, 2021.

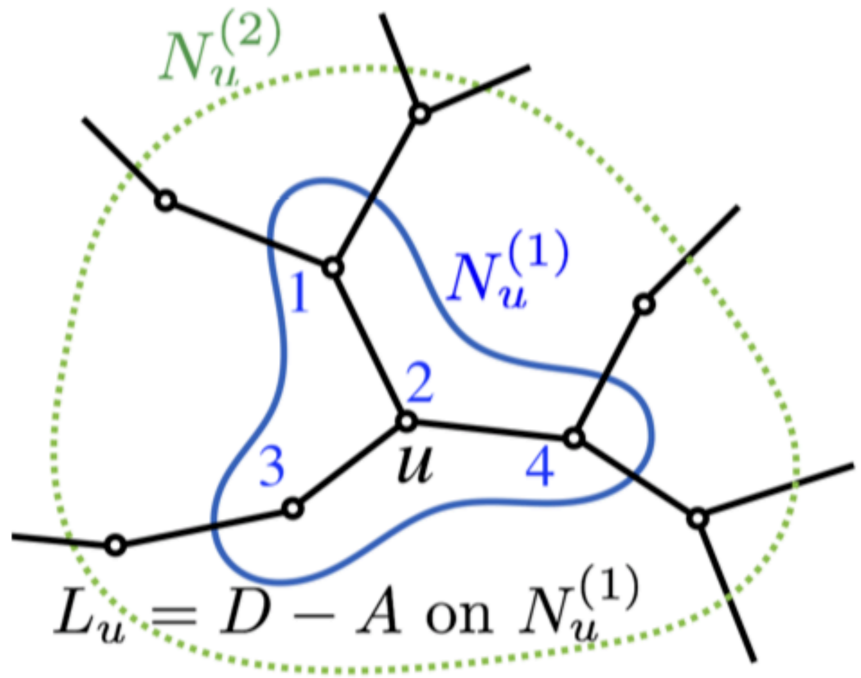
local basis filters



- replace Fourier basis with spectral graph wavelet basis to achieve localised convolution

Xu et al., "Graph wavelet neural network," ICLR, 2019.

local basis filters



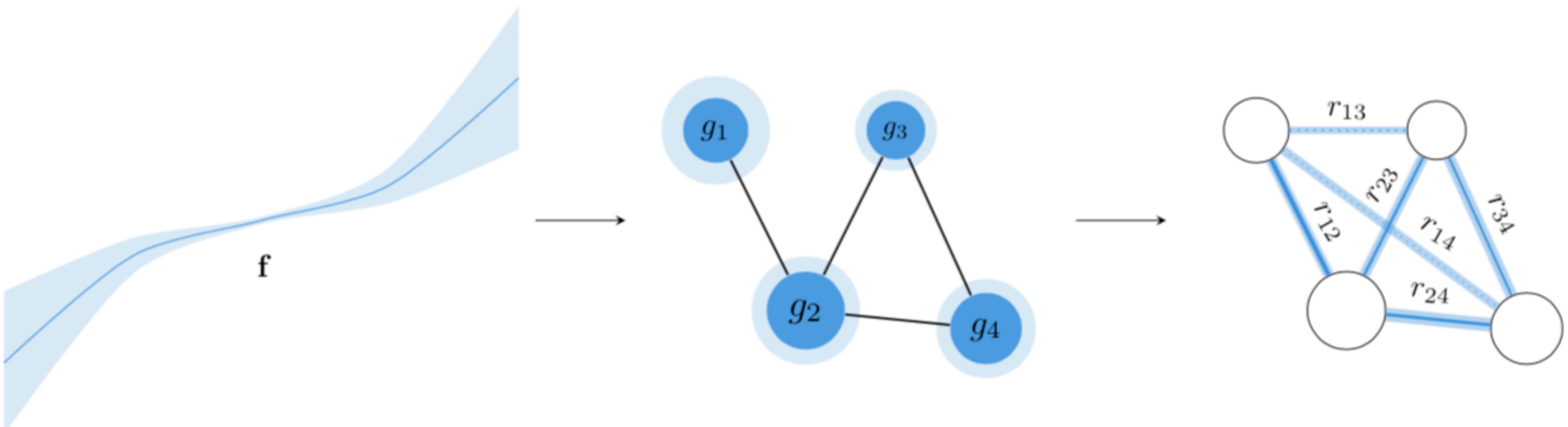
$$D = \begin{bmatrix} 3 & & & \\ & 3 & & \\ & & 2 & \\ & & & 3 \end{bmatrix}$$

$$A = \begin{bmatrix} & 1 & & \\ 1 & & 1 & 1 \\ & 1 & & \\ & 1 & & \end{bmatrix}$$

- learnable local filters where localisation is imposed in spatial domain
- regularisation by local graph Laplacian to improve robustness

Cheng et al., "Graph convolution with low-rank learnable local filters," ICLR, 2021.
Isufi et al., "EdgeNets: Edge varying graph neural networks," arXiv, 2020.

Graph convolutional Gaussian processes



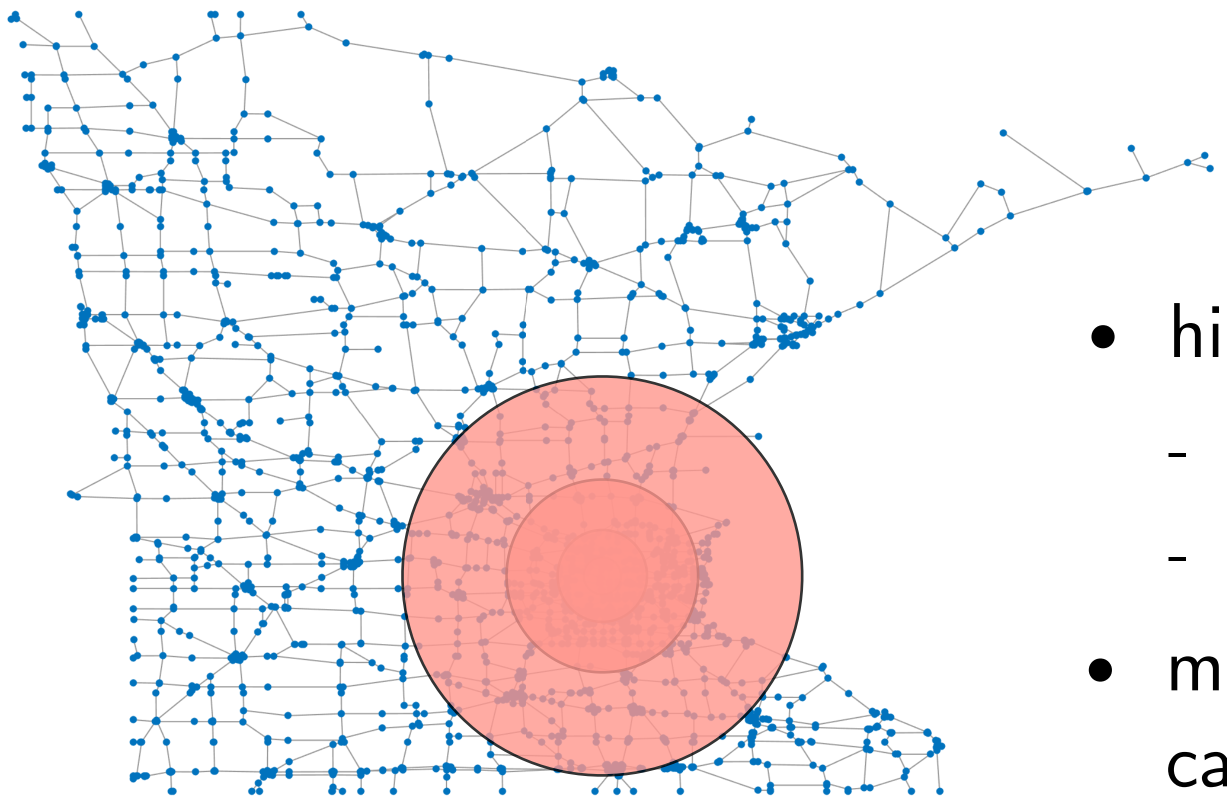
(a) Gaussian process on $\mathbb{R}^D$ (b) Gaussian process over $\mathcal{V}$ (c) Gaussian process over $\mathcal{E}$

- graph convolution enriches design of kernels associated with Gaussian processes (GPs)
- different formulations of convolution lead to different GP designs

Opolka and Liò, "Graph convolutional Gaussian processes for link prediction," ICML Workshop, 2020.
Walker and Glocker, "Graph convolutional Gaussian processes," ICML, 2019.

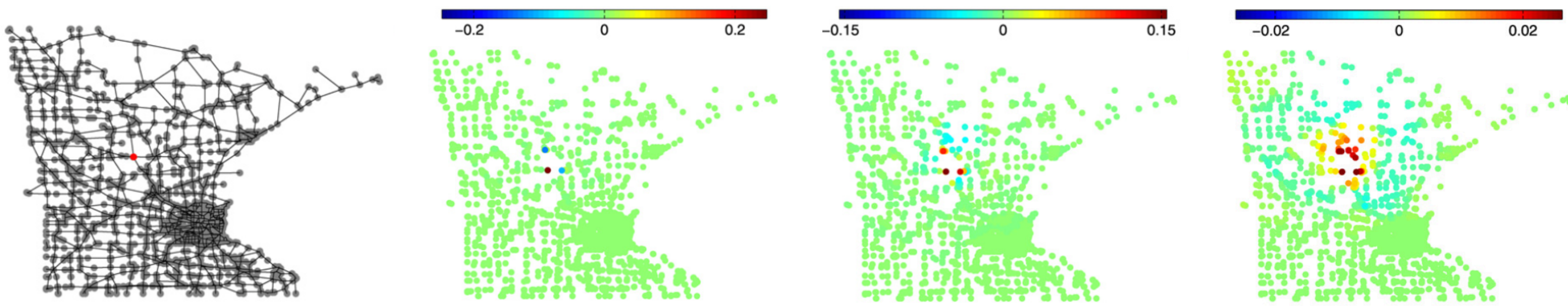
GSP for hierarchical modelling on graphs

Hierarchical modelling on graphs

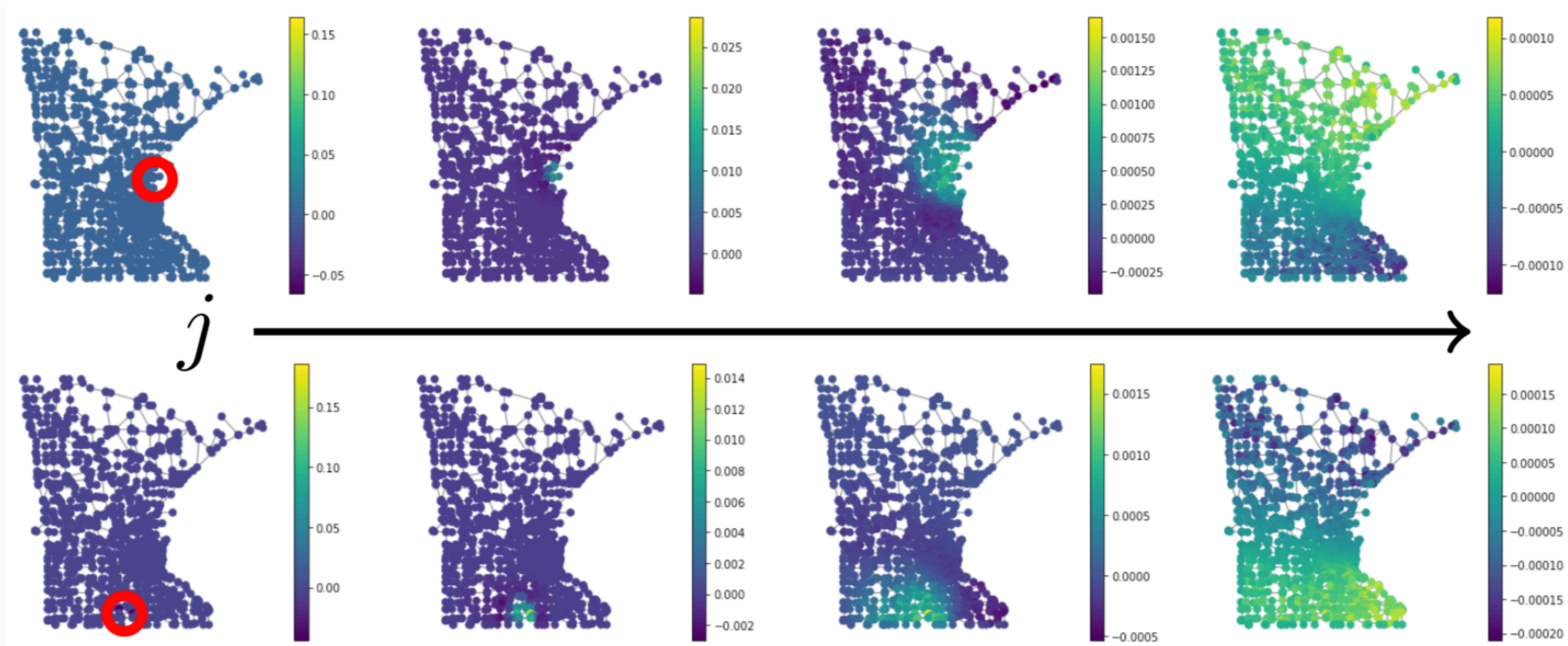


- hierarchical modelling in GNNs
  - increase size of filter or number of layers
  - graph pooling (downsampling)
- multiscale transforms (e.g., wavelets) can be natural tools for this

Wavelets on graphs



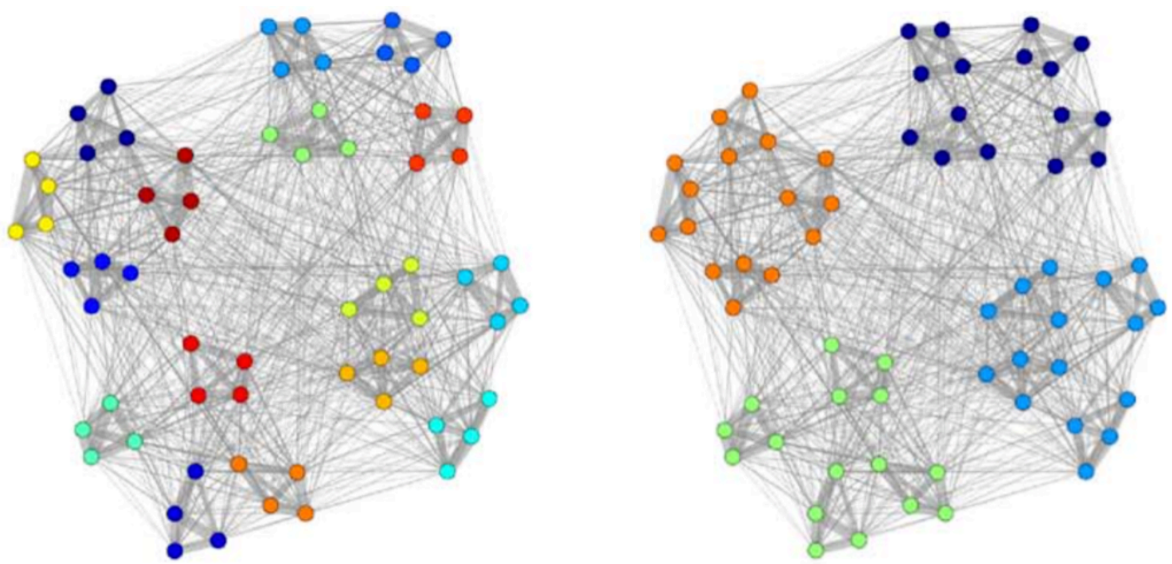
spectral graph wavelets



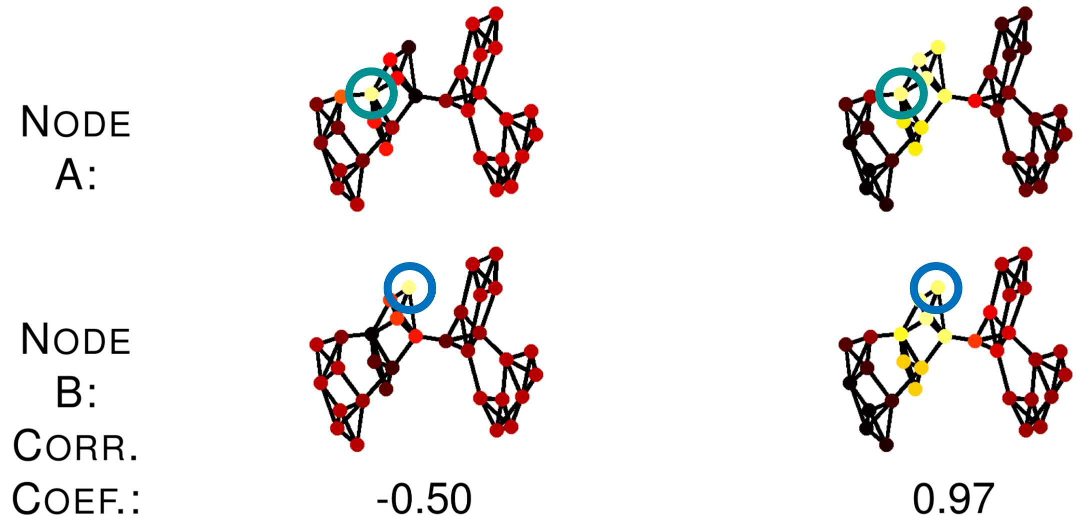
spatial graph wavelets

Hammond et al., "Wavelets on graphs via spectral graph theory," ACHA, 2011.
Gao et al., "Geometric scattering for graph data analysis," ICML, 2019.

Spectral graph wavelets for multiscale clustering



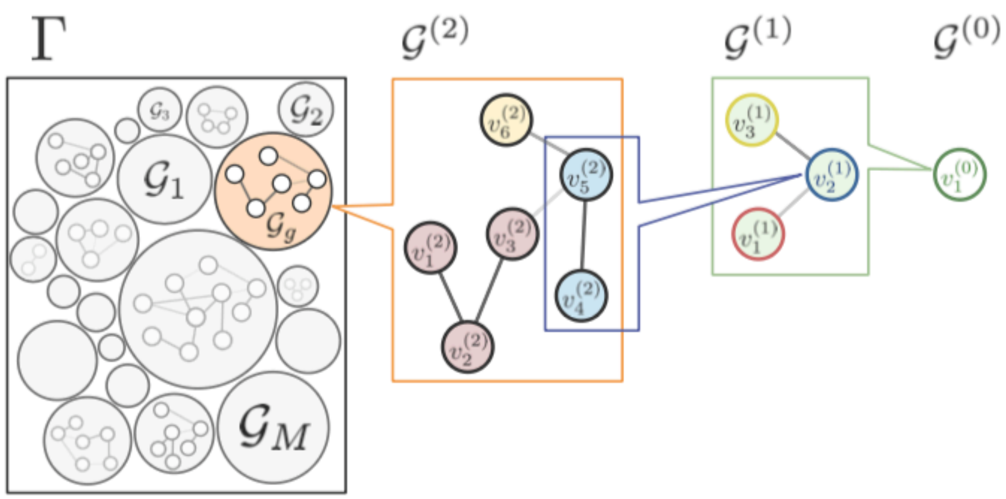
$$\mathbf{D}_s(a,b) = 1 - \frac{\psi_{s,a}^{\top}\psi_{s,b}}{||\psi_{s,a}||_2||\psi_{s,b}||_2}.$$



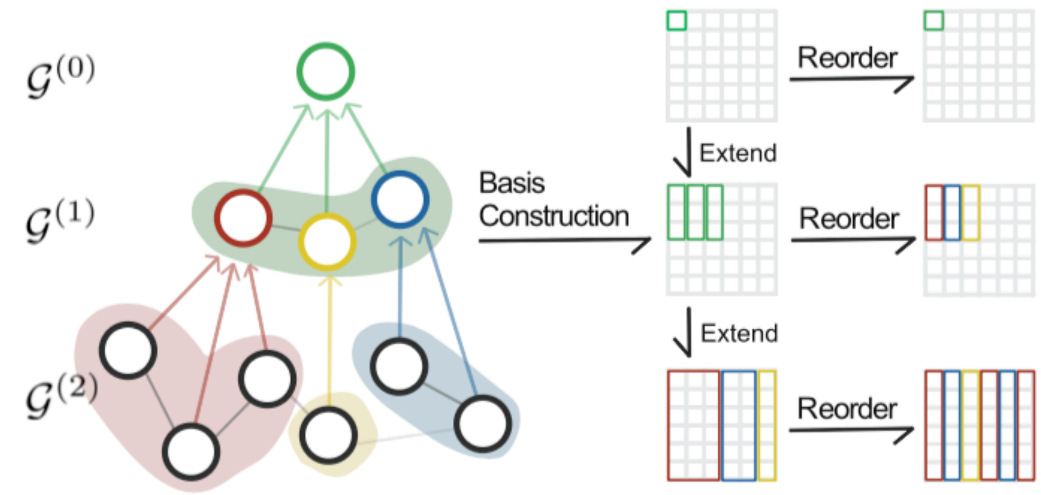
- spectral graph wavelets with different scales centred at node **u** provides an "egocentered view" of the graph seen from **u**
- similarity between nodes can be built at different levels to facilitate multiscale clustering

Tremblay and Borgnat, "Graph wavelets for multiscale community mining," IEEE TSP, 2014.

Haar-like wavelets for graph learning



coarse-grained graph chain

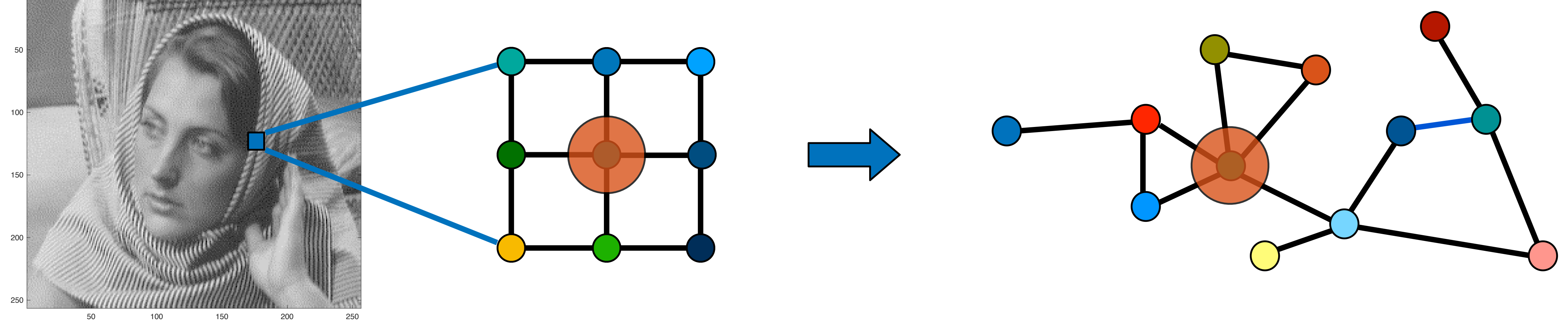


orthogonal basis construction

- given a coarse-grained graph chain an orthogonal basis is constructed
- this can then be used for both graph convolution and hierarchical graph pooling

Wang et al., "Haar graph pooling," ICML, 2020.
Zheng et al., "MathNet: Haar-like wavelet multiresolution analysis for graph representation and learning," arXiv, 2021.

GSP for exploiting data structure - Summary



- GSP enables various definitions of convolution on graphs
- graph filters enrich design of convolutional learning models on graphs (both GNNs and graph-based GPs)
- multiscale transforms (in particular wavelets) facilitate hierarchical modelling on graphs

References

- Shuman et al., "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.
- Sandryhaila and Moura, "Discrete signal processing on graphs," IEEE Transactions on Signal Processing, vol. 61, no. 7, pp. 1644-1656, April 2013.
- Ortega et al., "Graph signal processing: Overview, challenges, and applications," Proceedings of the IEEE, vol. 106, no. 5, pp. 808-828, May 2018.
- Dong et al., "Graph signal processing for machine learning: A review and new perspectives," IEEE Signal Processing Magazine, vol. 37, no. 6, pp. 117-127, November 2020.
- Mateos et al., "Connecting the dots: Identifying network structure via graph signal processing," IEEE Signal Processing Magazine, vol. 36, no. 3, pp. 16-43, May 2019.
- Dong et al., "Learning graphs from data: A signal representation perspective," IEEE Signal Processing Magazine, vol. 36, no. 3, pp. 44-63, May 2019.
- Gama et al., "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," IEEE Signal Processing Magazine, vol. 37, no. 6, pp. 128-138, November 2020.
- Cheung et al., "Graph signal processing and deep learning: Convolution, pooling, and topology," IEEE Signal Processing Magazine, vol. 37, no. 6, pp. 139-149, November 2020.
- Ruiz et al., "Graph neural networks: Architectures, stability, and transferability," Proceedings of the IEEE, vol. 109, no. 5, pp. 660-682, May 2021.

Image credit

- icons made by Freepik from www.flaticon.com
- https://cityu-bioinformatics.netlify.app/too2/new_pheno/brain/
- https://stock.adobe.com/images/polygonal-mesh-map-of-switzerland-in-black-color-abstract-mesh-lines-triangles-and-points-with-map-of-switzerland-wire-frame-2d-polygonal-line-network-in-vector-format/236496556
- https://commons.wikimedia.org/wiki/File:6_centrality_measures.png
- https://en.wikipedia.org/wiki/File:Zachary%27s_karate_club.png
- https://www.meteoswiss.admin.ch/home/climate/swiss-climate-in-detail/monthly-and-annual-maps.html
- https://commons.wikimedia.org/wiki/File:Chebyshev_Polynomials_of_the_First_Kind.svg