AIMS CDT - Signal Processing

Michaelmas Term 2025

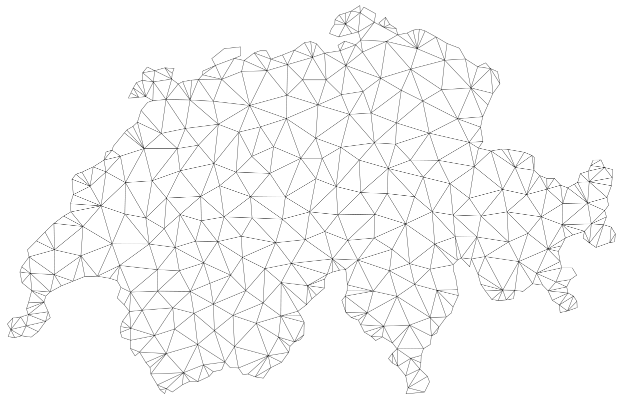
Xiaowen Dong

Department of Engineering Science

UNIVERSITY OF OXFORD

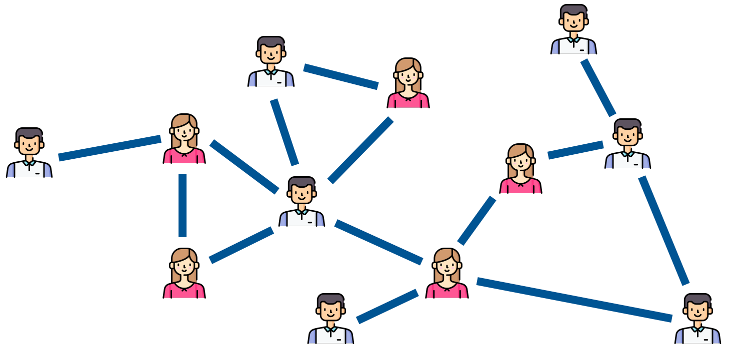
Introduction to
Graphs Signal Processing

Networks are pervasive

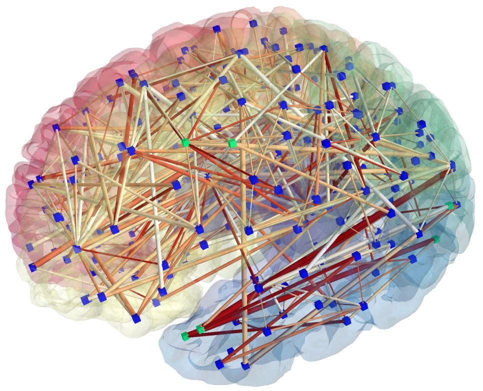


geographical network

traffic network



social network

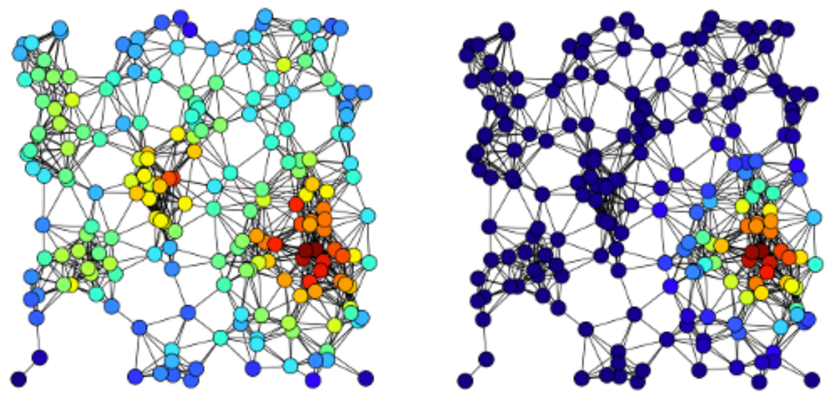


brain network

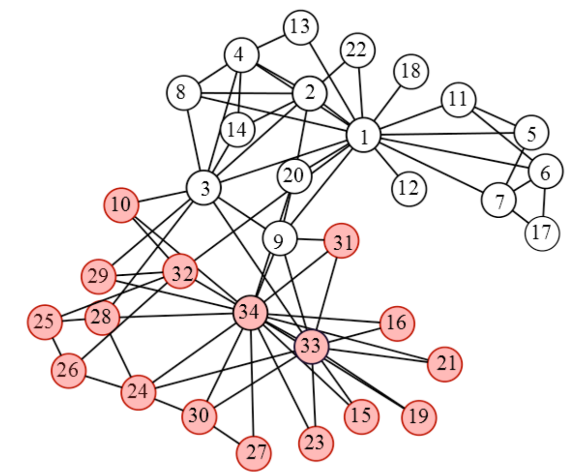
graphs provide mathematical representation of networks

Network science

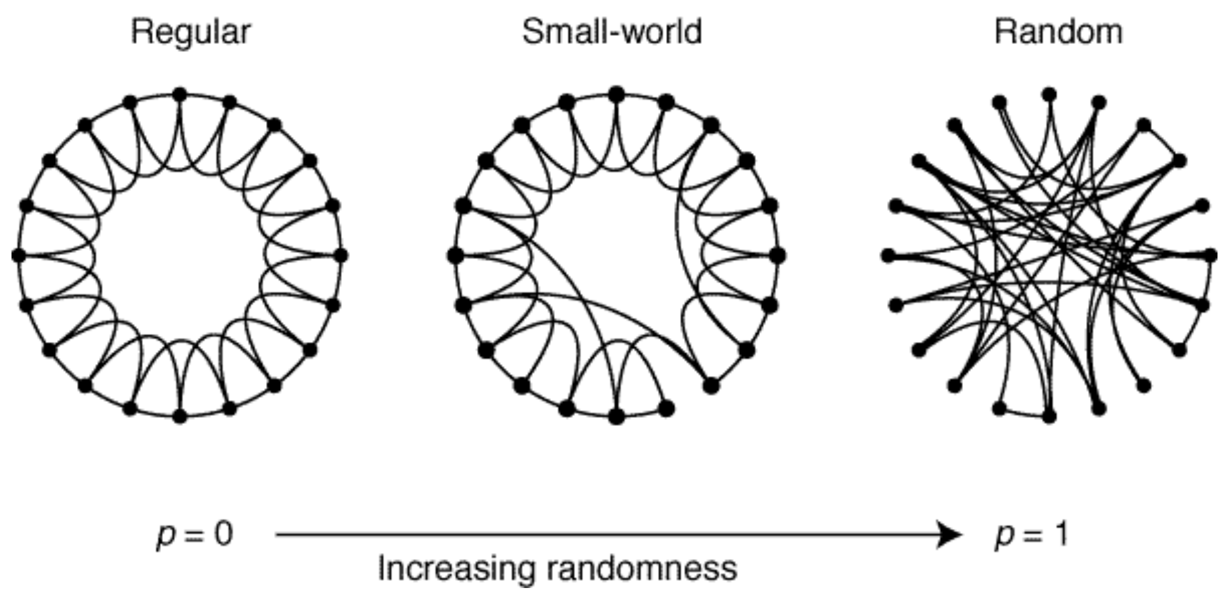
network centrality



community detection



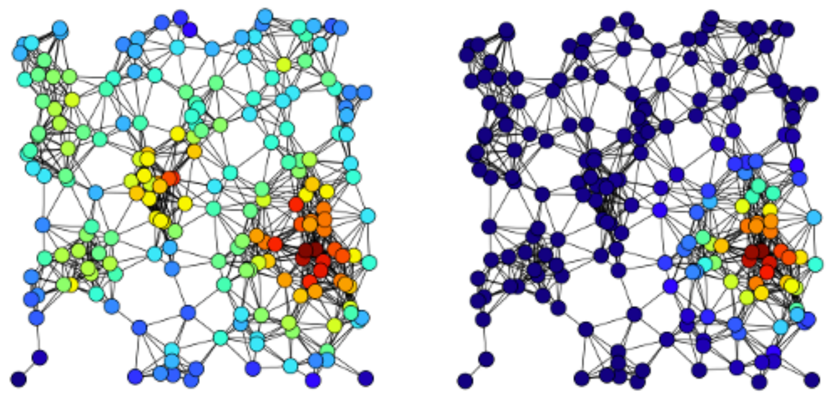
random graph models



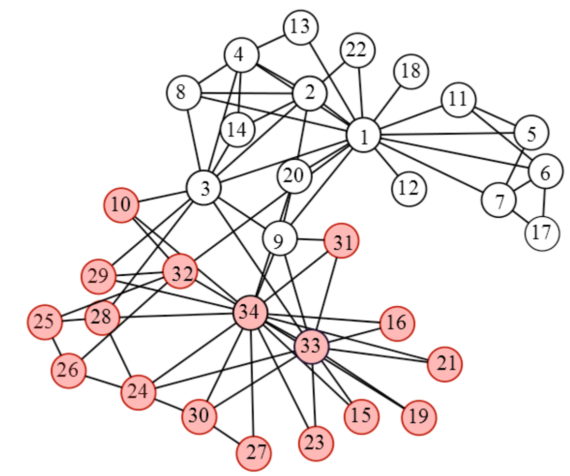
Watts and Strogatz, "Collective dynamics of 'small-world' networks," Nature, 1998.
Newman, "Networks: An introduction," Oxford University Press, 2010.

Network science

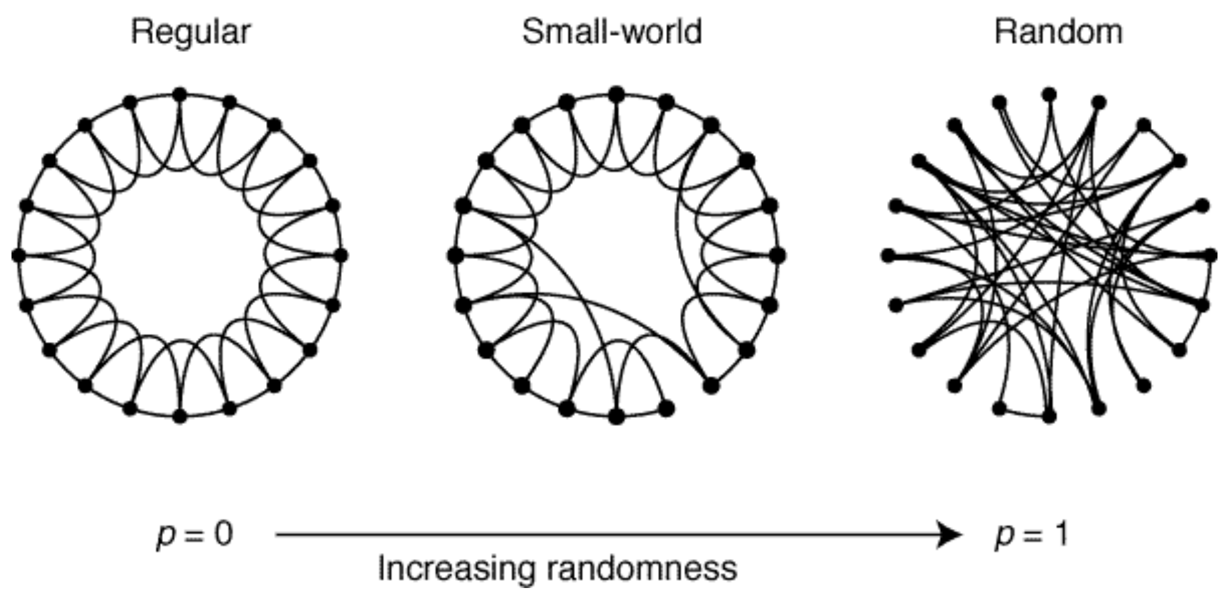
network centrality



community detection



random graph models

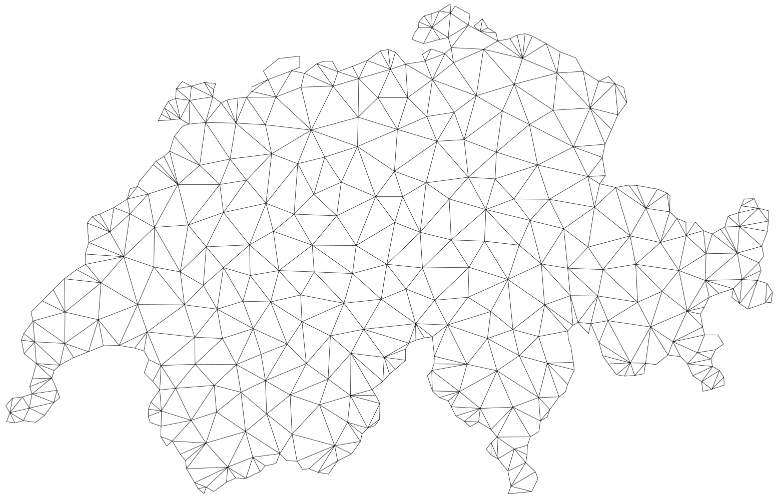


from edge attributes to node attributes

from graphs to graph-structured data

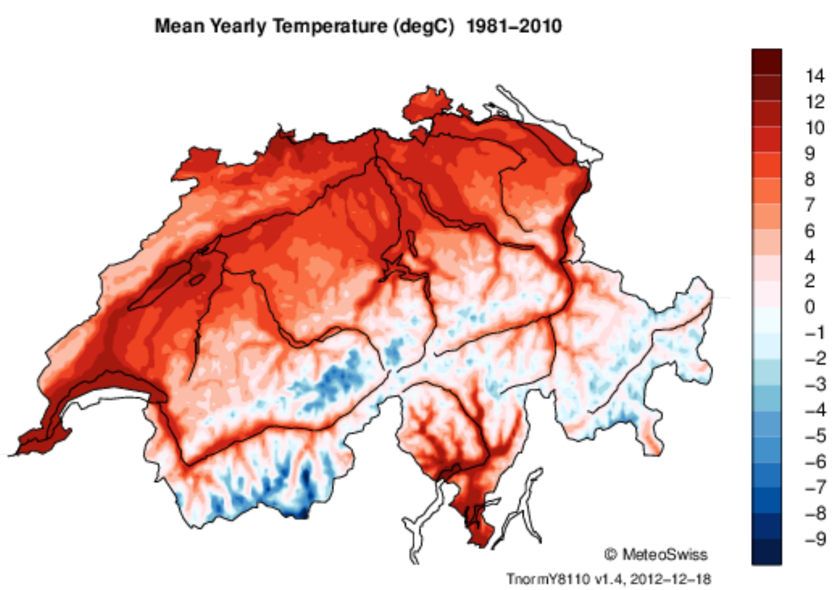
Watts and Strogatz, "Collective dynamics of 'small-world' networks," Nature, 1998.
Newman, "Networks: An introduction," Oxford University Press, 2010.

Graph-structured data are pervasive



- nodes
  - geographical regions
- edges
  - geographical proximity between regions

Graph-structured data are pervasive

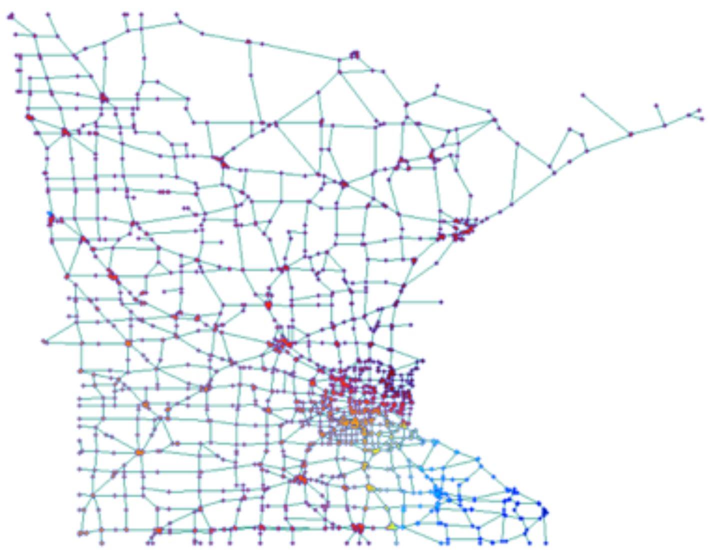


- nodes
  - geographical regions
- edges
  - geographical proximity between regions
- signal
  - temperature records in these regions

Graph-structured data are pervasive

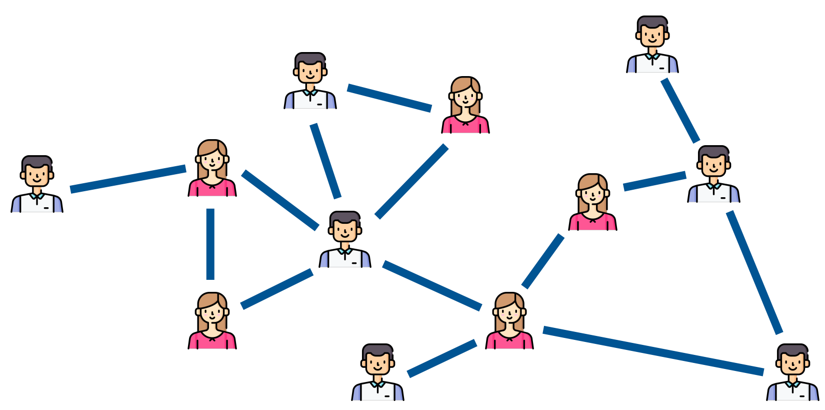
- nodes
  - road junctions
- edges
  - road connections

Graph-structured data are pervasive



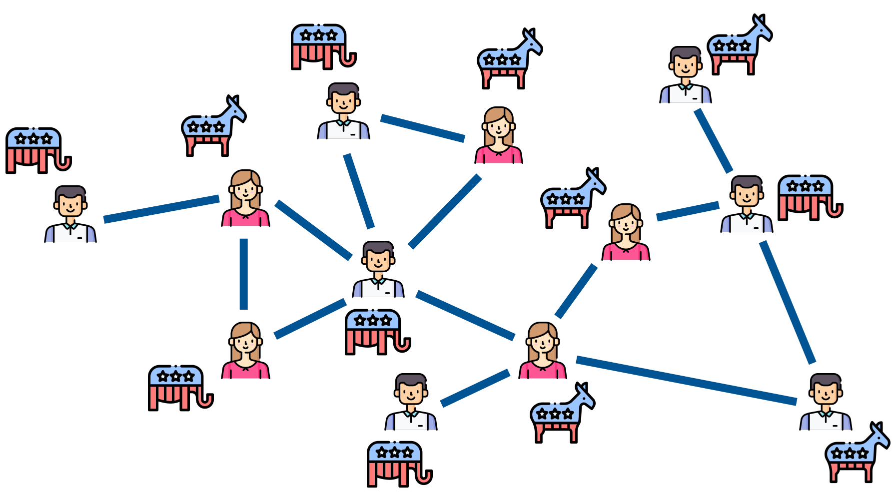
- nodes
  - road junctions
- edges
  - road connections
- signal
  - traffic congestion at junctions

Graph-structured data are pervasive



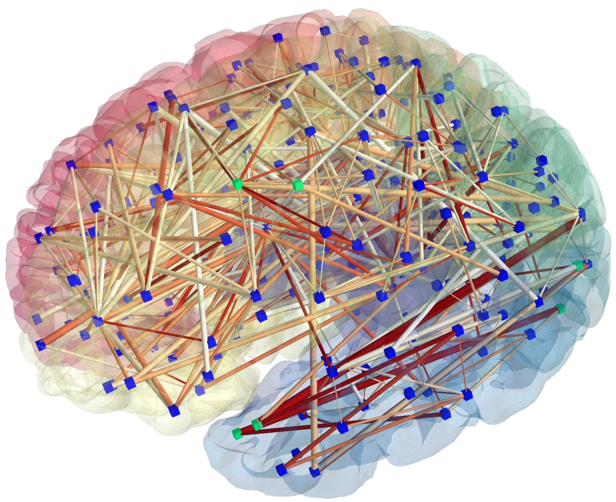
- nodes
  - individuals
- edges
  - friendship between individuals

Graph-structured data are pervasive



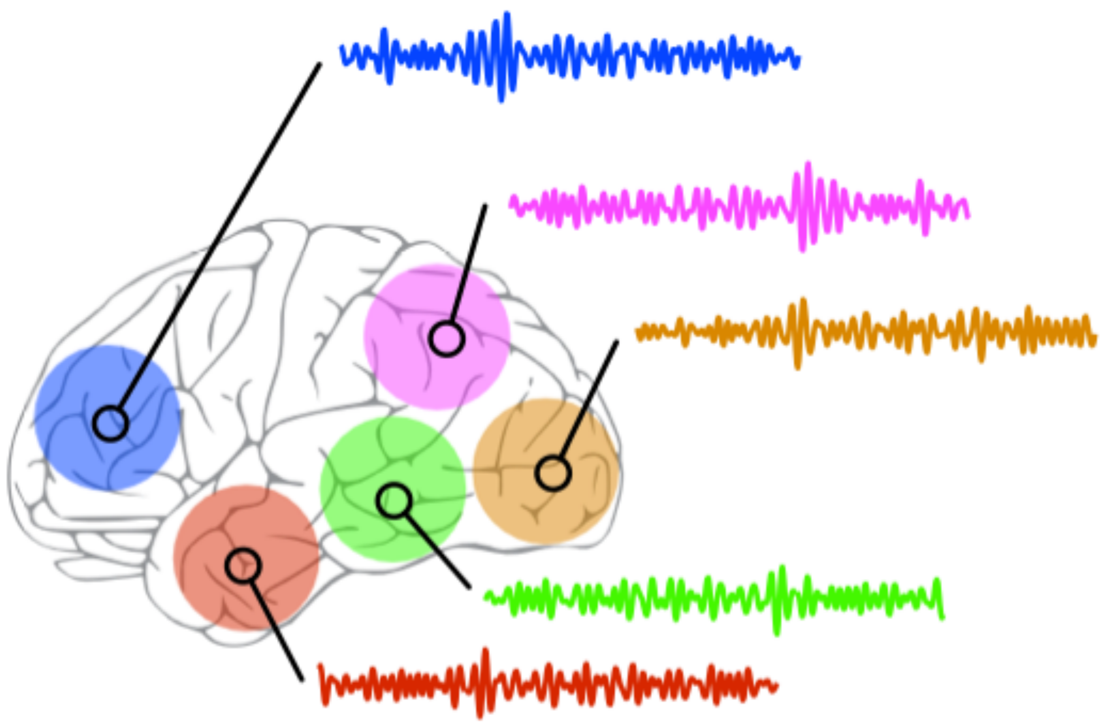
- nodes
  - individuals
- edges
  - friendship between individuals
- signal
  - political view

Graph-structured data are pervasive



- nodes
  - brain regions
- edges
  - structural connectivity between brain regions

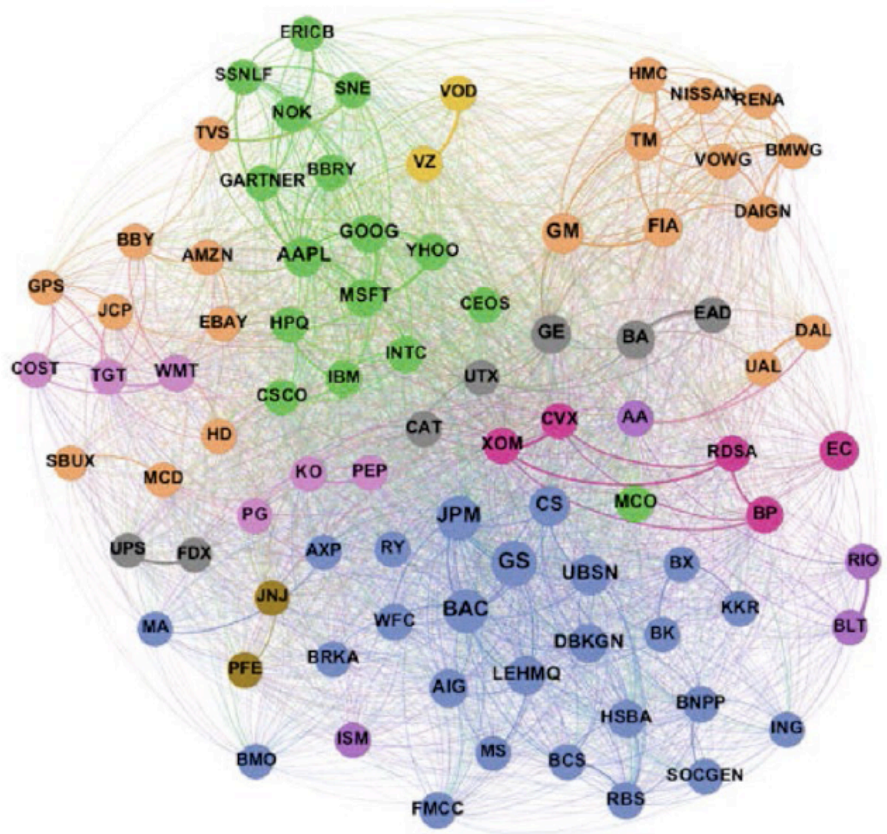
Graph-structured data are pervasive



- 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.

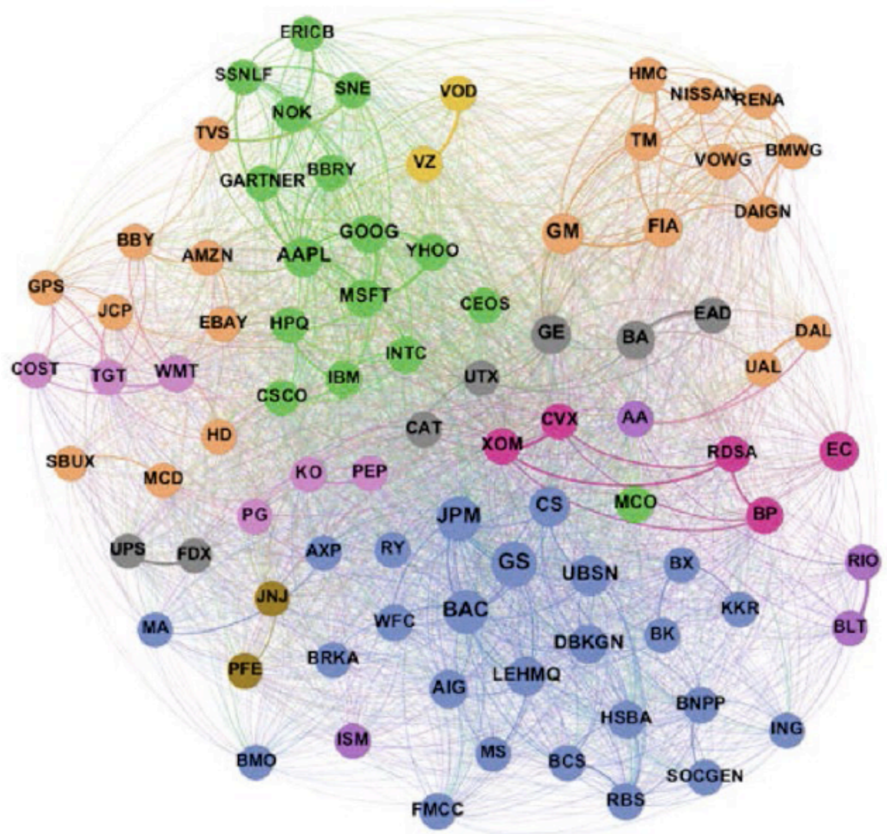
Graph-structured data are pervasive



- nodes
  - companies
- edges
  - co-occurrence of companies in financial news

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

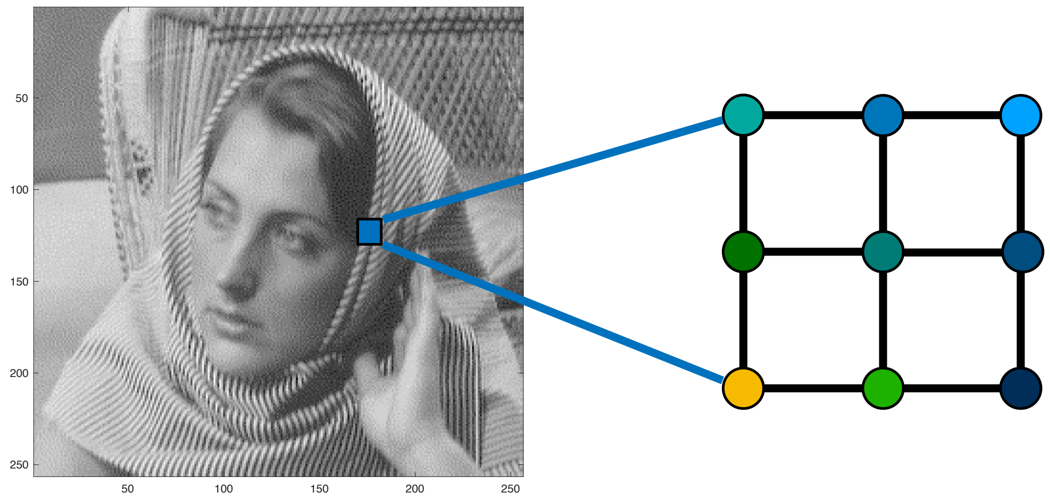
Graph-structured data are pervasive



- 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.

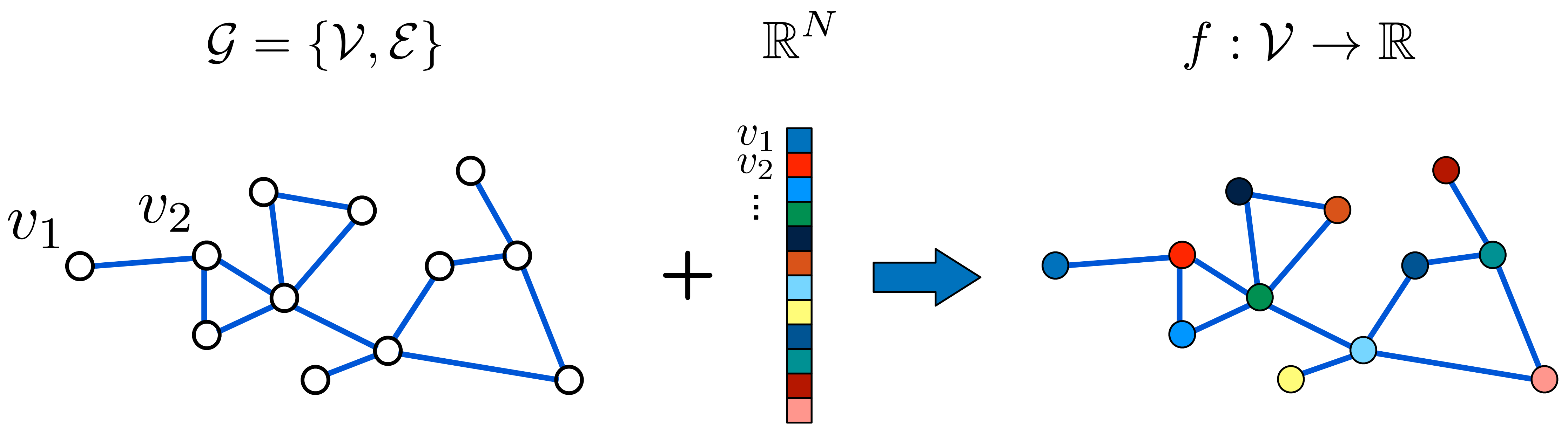
Graph-structured data are pervasive



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

Graph signal processing

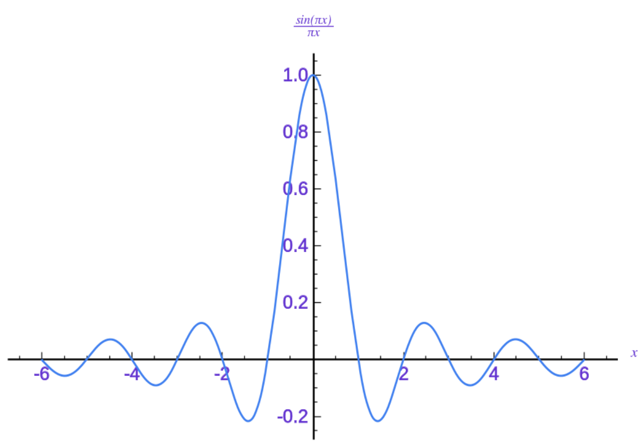
- Graph-structured data can be represented by graph signals



takes into account both structure (edges) and data (values at nodes)

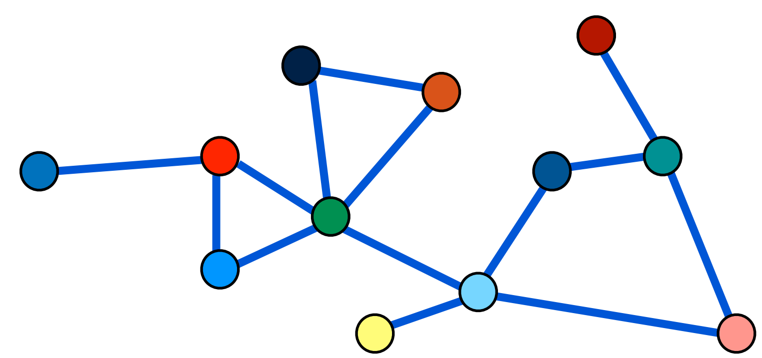
Graph signal processing

1D signal



2D signal

$f : \mathcal{V} \to \mathbb{R}$



how to generalise classical signal processing tools on irregular domains such as graphs?

Lecture 2

- Graph signal processing: Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Representation of graph signals
- Applications

Graph signal processing

- Main GSP approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs

Graph signal processing

- Main GSP approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs

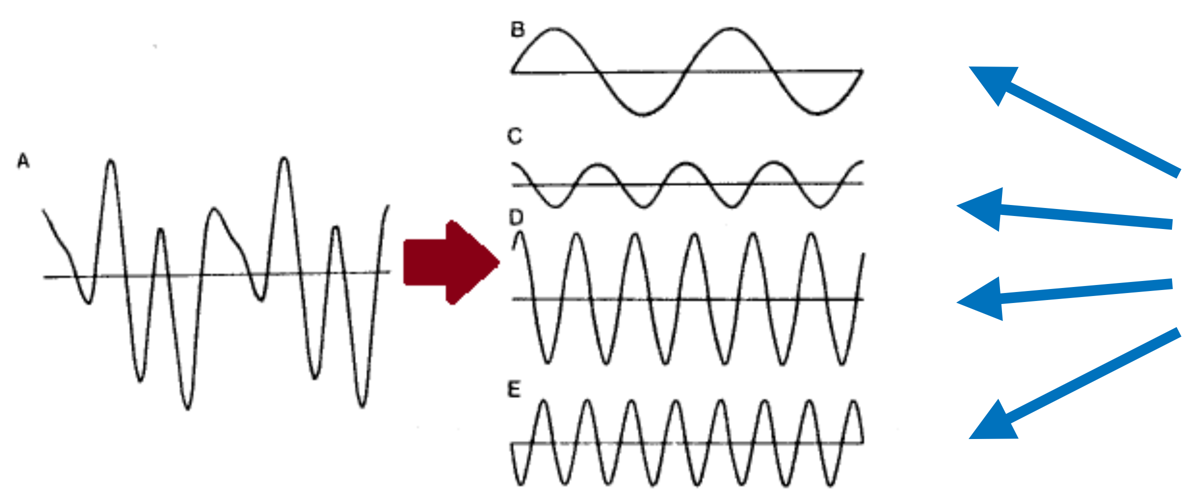
important for signal analysis

Graph signal processing

- Main GSP approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs

important for signal analysis

- Classical Fourier transform provides frequency domain representation of signals



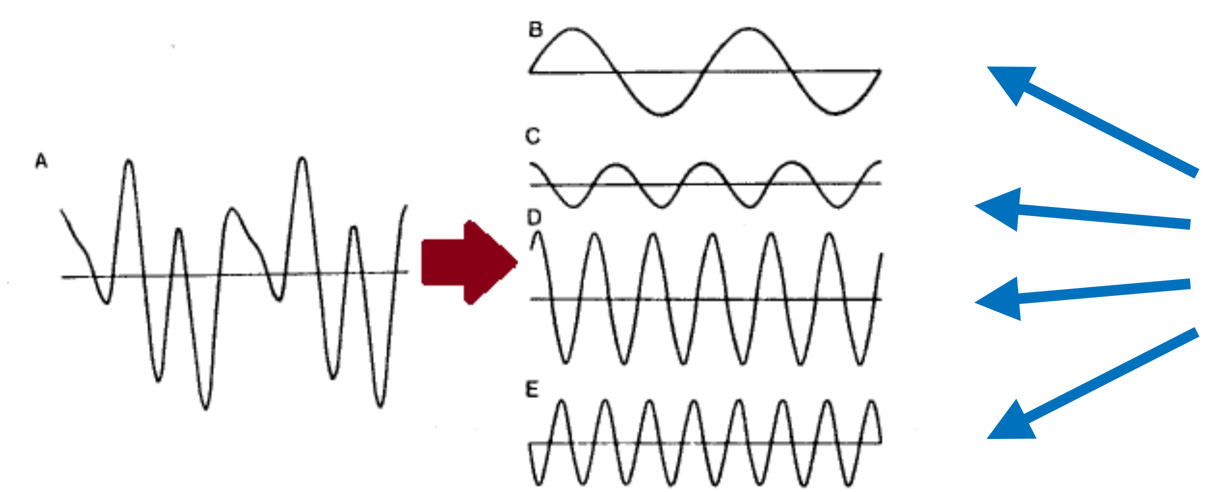
- “building blocks” of signal
- different frequency (oscillation)

Graph signal processing

- Main GSP approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs

important for signal analysis

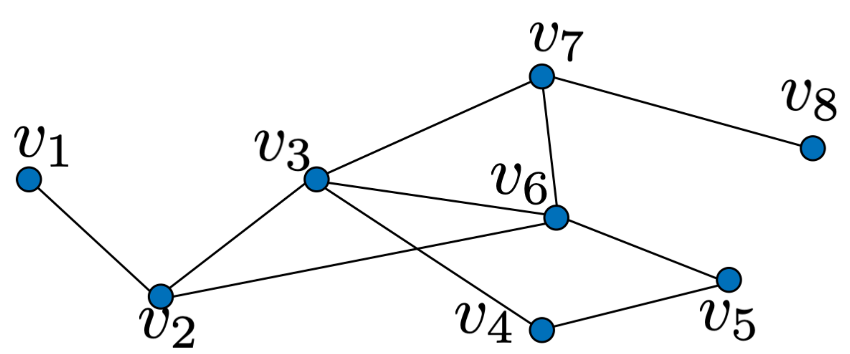
- Classical Fourier transform provides frequency domain representation of signals



- "building blocks" of signal
- different frequency (oscillation)

- What about a notion of frequency for graph signals?

Graph Laplacian



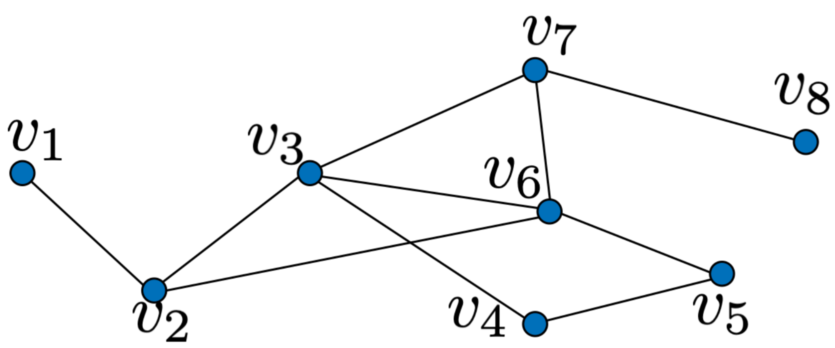
weighted and undirected graph:

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

Graph Laplacian

weighted and undirected graph:

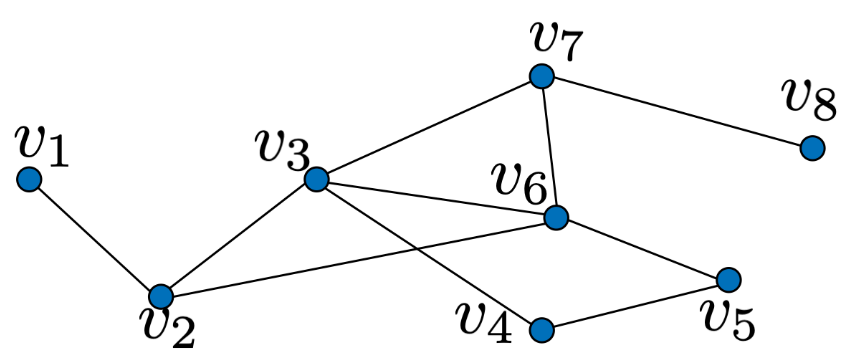
$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$



$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$W$$

Graph Laplacian



weighted and undirected graph:

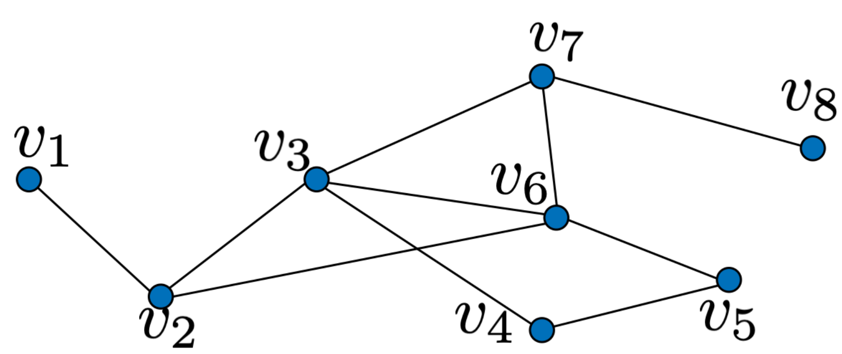
$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$$

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

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$D \qquad\qquad\qquad W$$

Graph Laplacian



weighted and undirected graph:

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

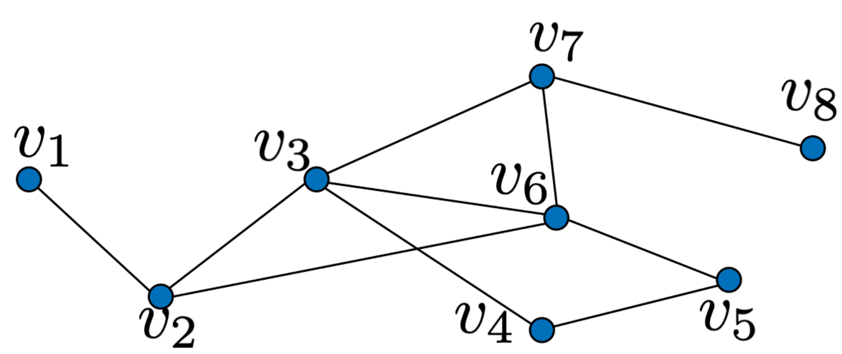
$$D = \mathrm{diag}(d(v_1), \cdots, d(v_N))$$

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

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

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

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 W!**

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

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

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

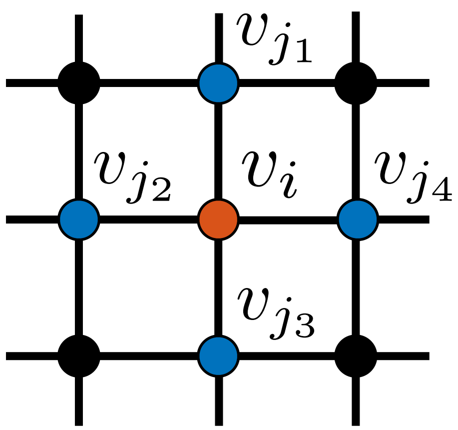
Graph Laplacian

Why graph Laplacian?

Graph Laplacian

Why graph Laplacian?

- provides an approximation of the Laplace operator



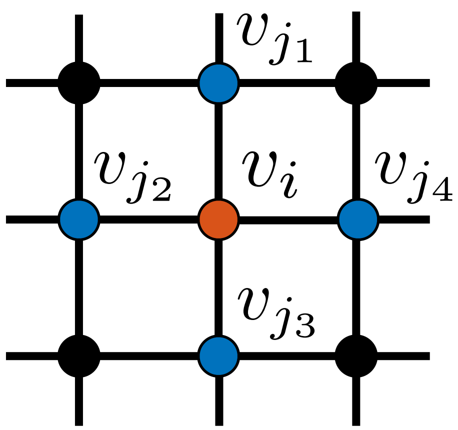
$$(Lf)(i) = \big(4f(i) - f(j_1) - f(j_2) - f(j_3) - f(j_4)\big)/(\delta x)^2$$

standard 5-point stencil for approximating $-\nabla^2 f$

Graph Laplacian

Why graph Laplacian?

- provides an approximation of the Laplace operator

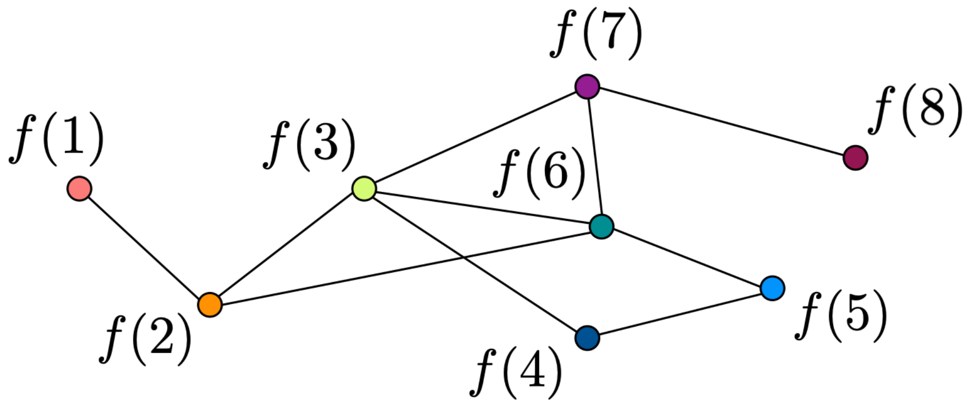


$$(Lf)(i) = \big(4f(i) - f(j_1) - f(j_2) - f(j_3) - f(j_4)\big)/(\delta x)^2$$

standard 5-point stencil for approximating $-\nabla^2 f$

- converges to the Laplace-Beltrami operator (given certain conditions)
- provides a notion of "frequency" on graphs

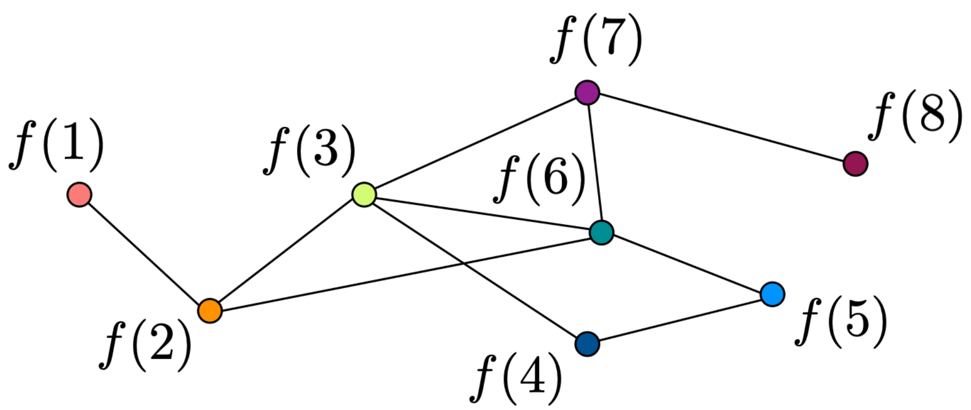
Graph Laplacian



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

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

Graph Laplacian



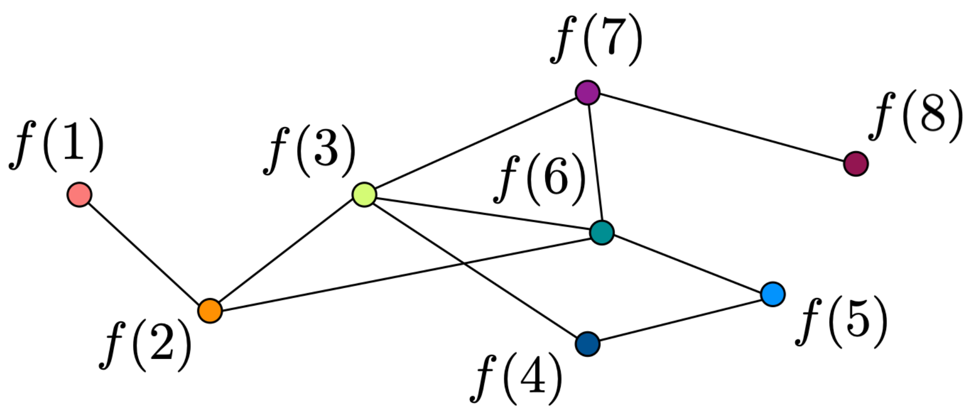
graph signal $f : \mathcal{V} \to \mathbb{R}$

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

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

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

Graph Laplacian



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

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

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

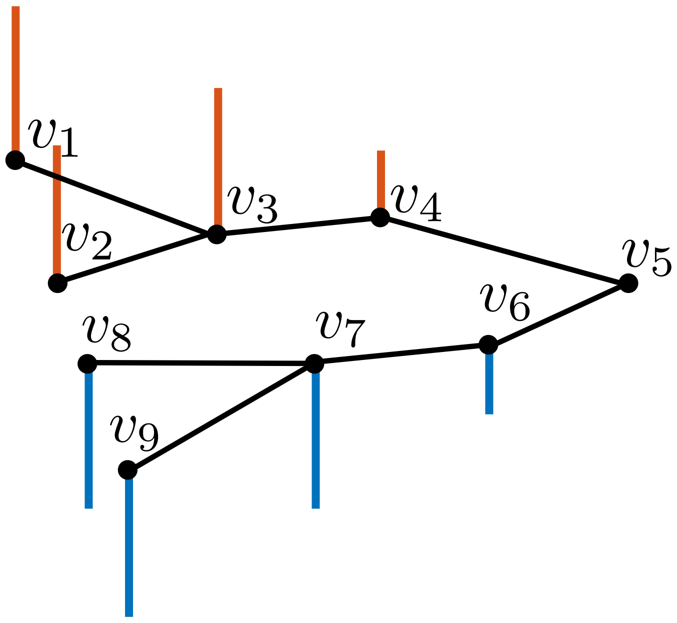
$$Lf(i) = \sum_{j=1}^{N} W_{ij}(f(i) - f(j))$$

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

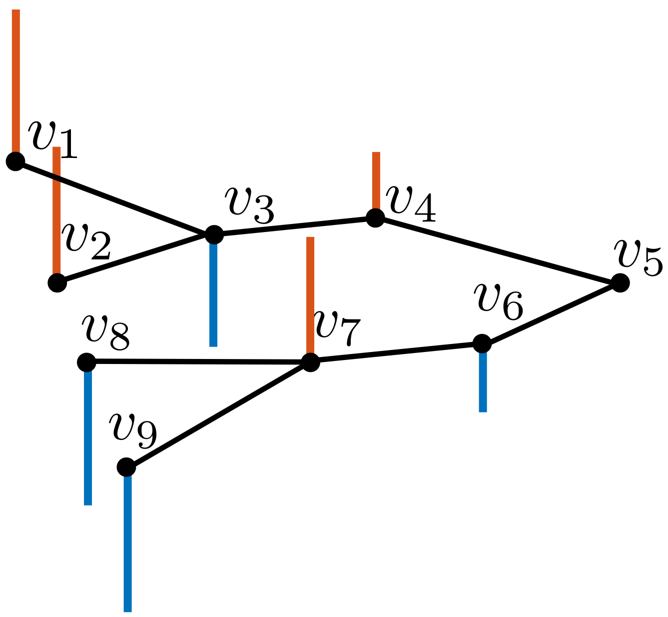
a measure of "smoothness"

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

Graph Laplacian



$$f^T L f = 1$$



$$f^T L f = 21$$

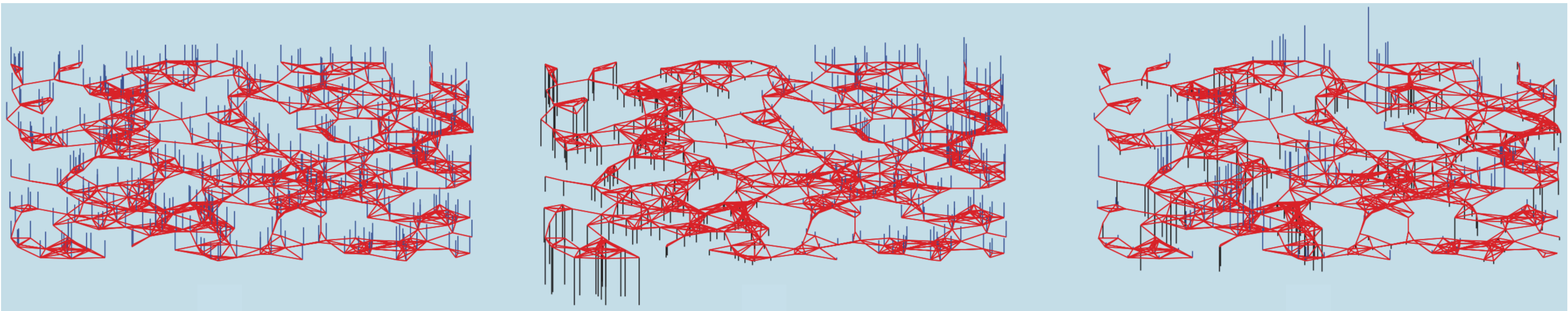
Graph Laplacian

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

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

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

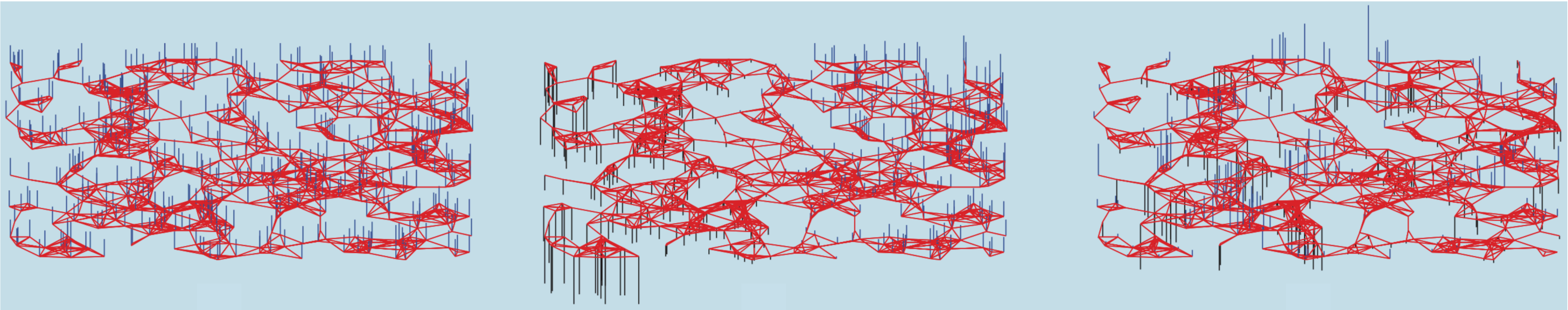
Graph Fourier transform



χ_0 χ_1 χ_{50}

Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

Graph Fourier transform



χ_0 χ_1 χ_{50}

low frequency → high frequency

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

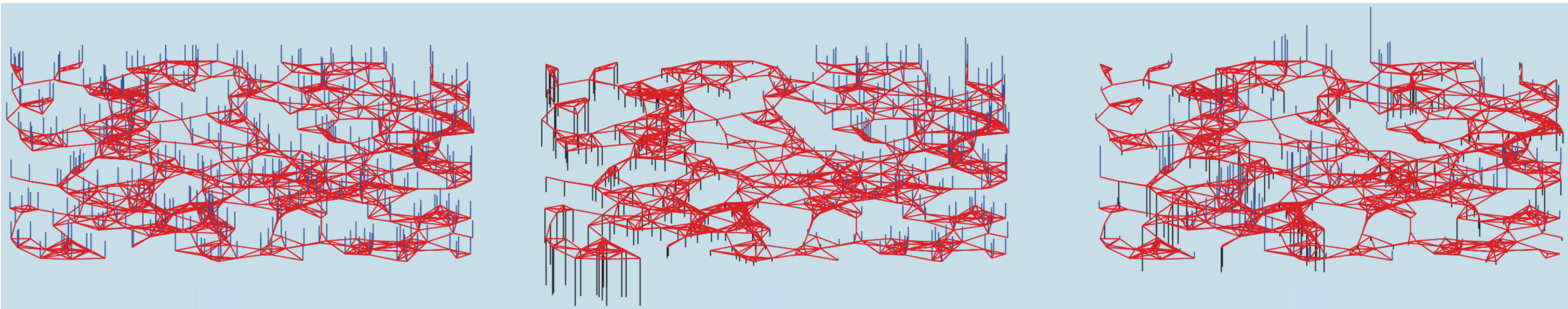
$\chi_0^T L \chi_0 = \lambda_0 = 0$

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

- Eigenvectors associated with smaller eigenvalues have values that vary less rapidly along the edges

Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

Graph Fourier transform



χ_0 χ_1 χ_{50}

low frequency → high frequency

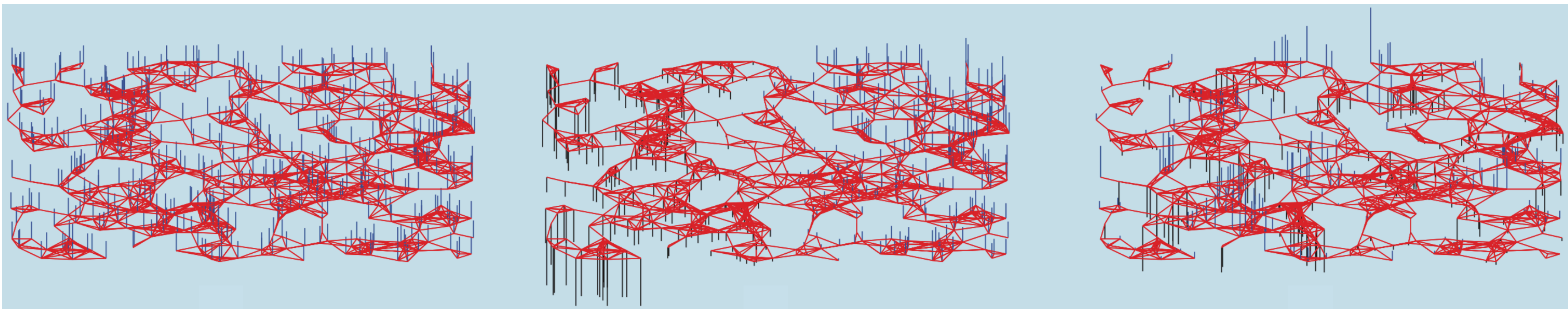
$$L = \chi \Lambda \chi^T$$

$\chi_0^T L \chi_0 = \lambda_0 = 0$ $\chi_{50}^T L \chi_{50} = \lambda_{50}$

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



χ_0 χ_1 χ_{50}

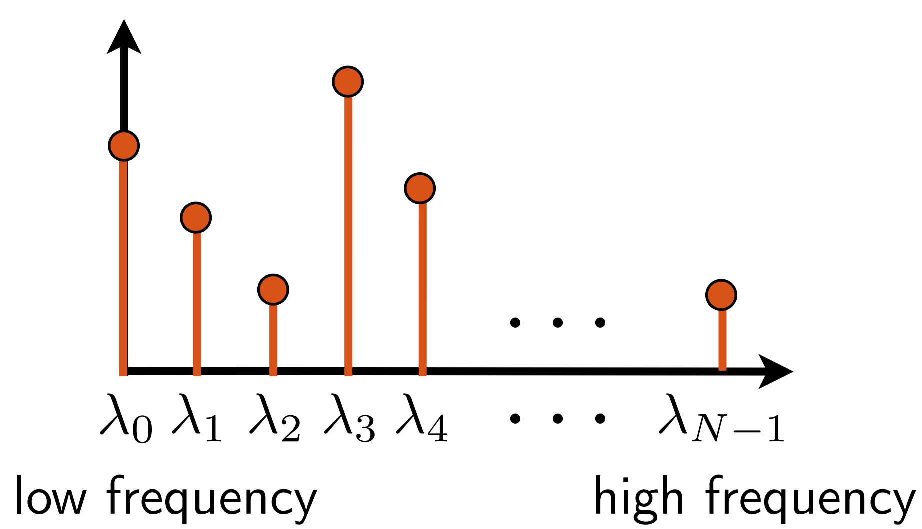
low frequency → high frequency

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

$\chi_0^T L \chi_0 = \lambda_0 = 0$ $\chi_{50}^T L \chi_{50} = \lambda_{50}$

graph Fourier transform:

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



Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $-\nabla^2$

$\Downarrow$

eigenfunctions: $e^{j\omega x}$

$\Downarrow$

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $-\nabla^2$

eigenfunctions: $e^{j\omega x}$

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

graph Laplacian: L

eigenvectors: χ_ℓ

$f : V \to \mathbb{R}^N$

Graph FT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i)$

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $-\nabla^2$

$\Downarrow$

eigenfunctions: $e^{j\omega x}$

$\Downarrow$

Classical FT: $\hat{f}(\omega) = \boxed{\int} \boxed{(e^{j\omega x})^*} \boxed{f(x)} dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

graph Laplacian: L

$\Downarrow$

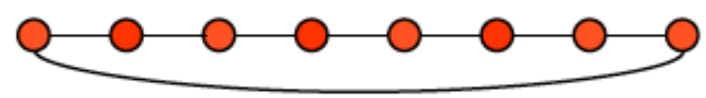
eigenvectors: χ_ℓ

$\Downarrow$ $f : V \to \mathbb{R}^N$

Graph FT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \boxed{\sum_{i=1}^{N}} \boxed{\chi_\ell^*(i)} \boxed{f(i)}$

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$$

Graph Fourier transform: Special case 1



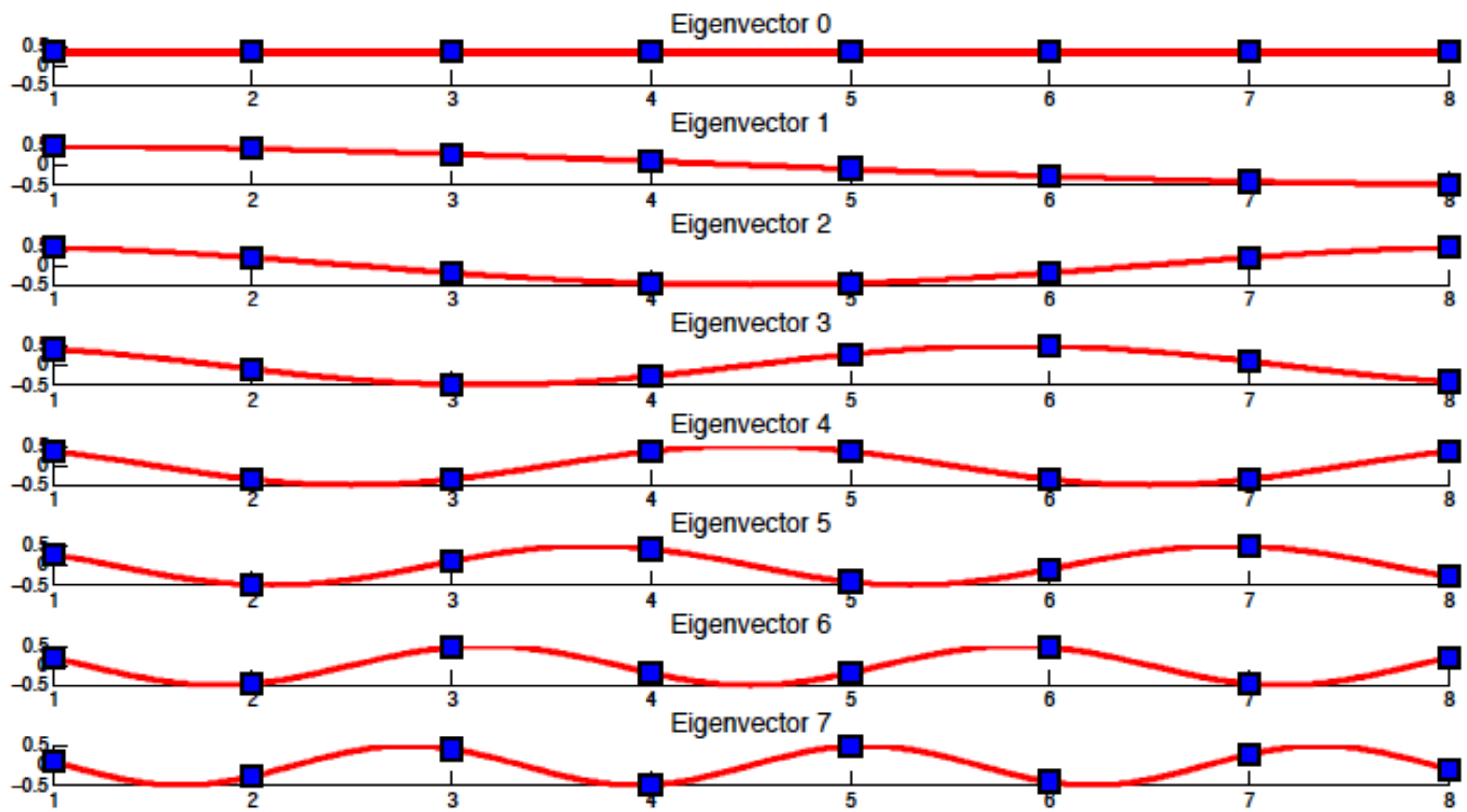
- (Unordered) Laplacian eigenvalues: $\lambda_\ell = 2 - 2\cos\left(\frac{2\ell\pi}{N}\right)$
- One possible choice of orthogonal Laplacian eigenvectors:

$$\chi_\ell = \left[1, \omega^\ell, \omega^{2\ell}, \ldots, \omega^{(N-1)\ell}\right], \text{ where } \omega = e^{\frac{2\pi j}{N}}$$

- $\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N-1} \\ | & & | \end{bmatrix}$ is the Discrete Fourier Transform (DFT) matrix

Graph Fourier transform: Special case 2

- $\lambda_\ell = 2 - 2\cos\left(\frac{\pi\ell}{N}\right)$
- $\chi_0(i) = \frac{1}{\sqrt{N}}, \; \chi_\ell(i) = \sqrt{\frac{2}{N}}\cos\left(\frac{\pi\ell(i-0.5)}{N}\right), \; \ell = 1, 2, \ldots, N-1$

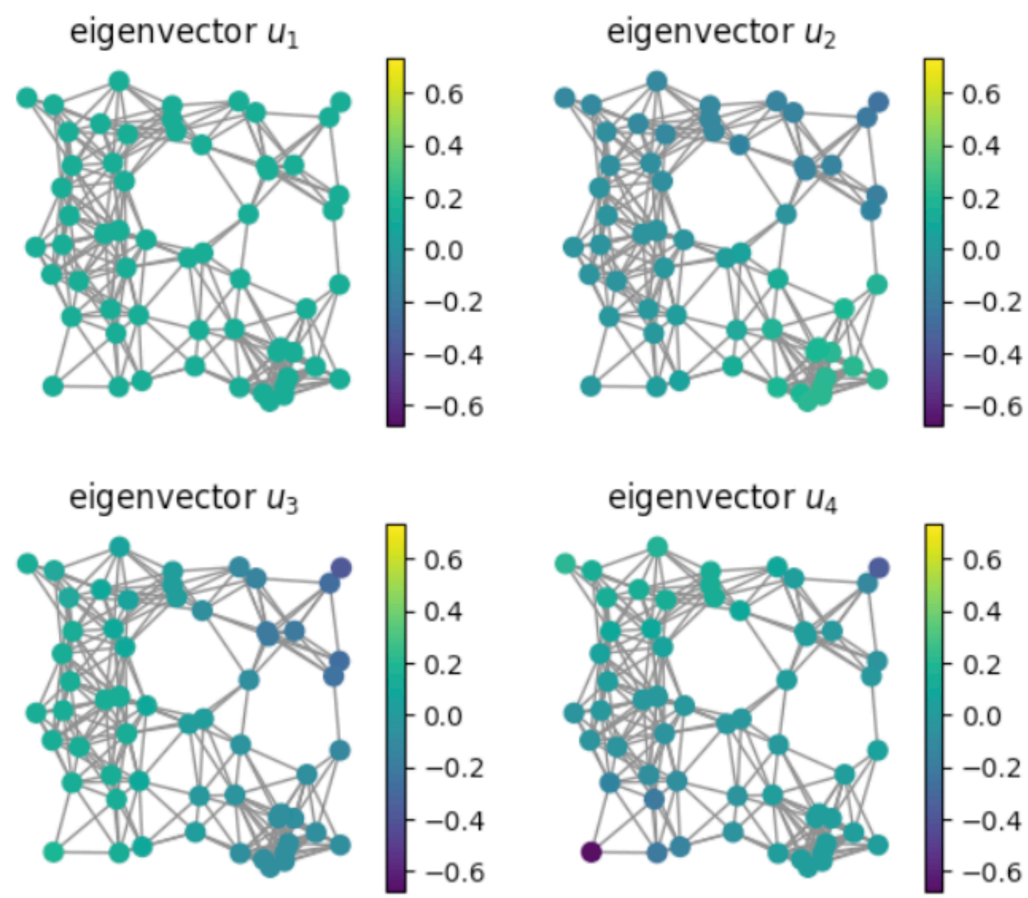


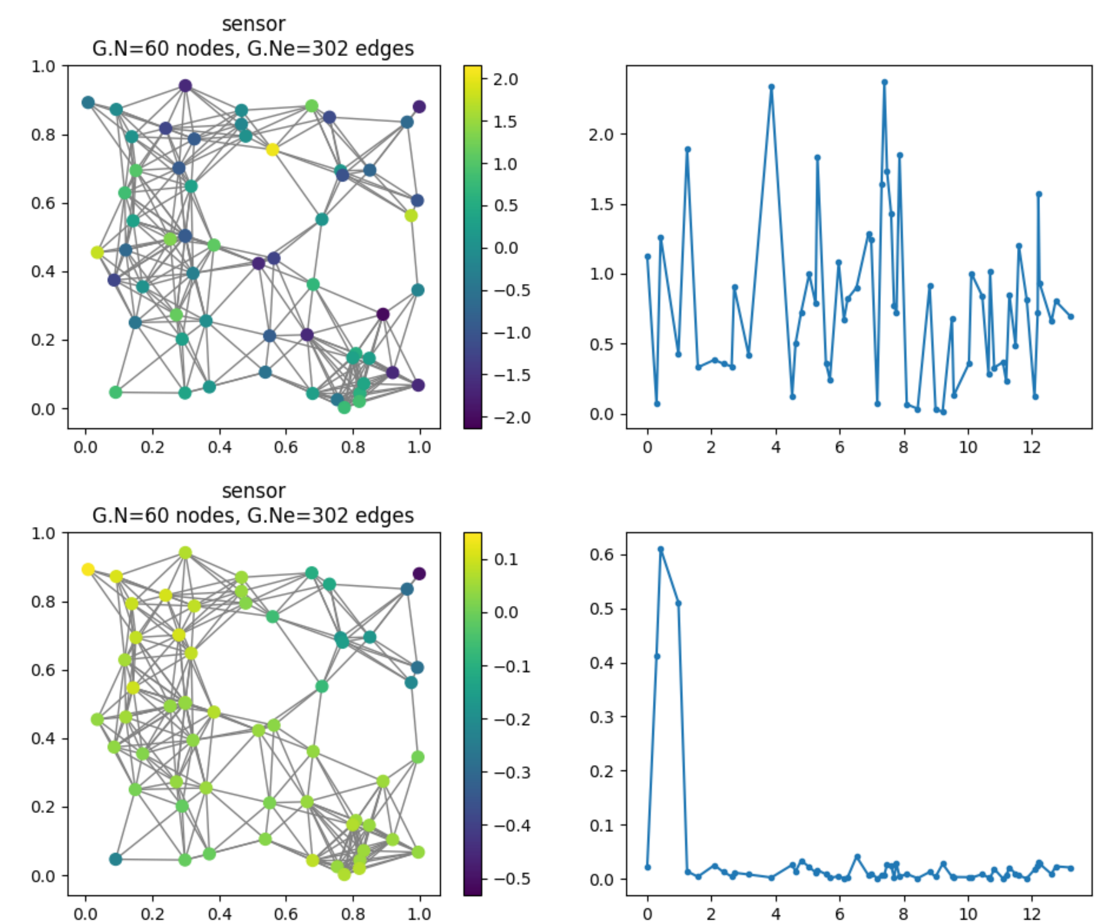
$\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N-1} \\ | & & | \end{bmatrix}$ is the Discrete Cosine Transform matrix (DCT-II, Strang, 1999), which is used in JPEG image compression

Example on a generic graph

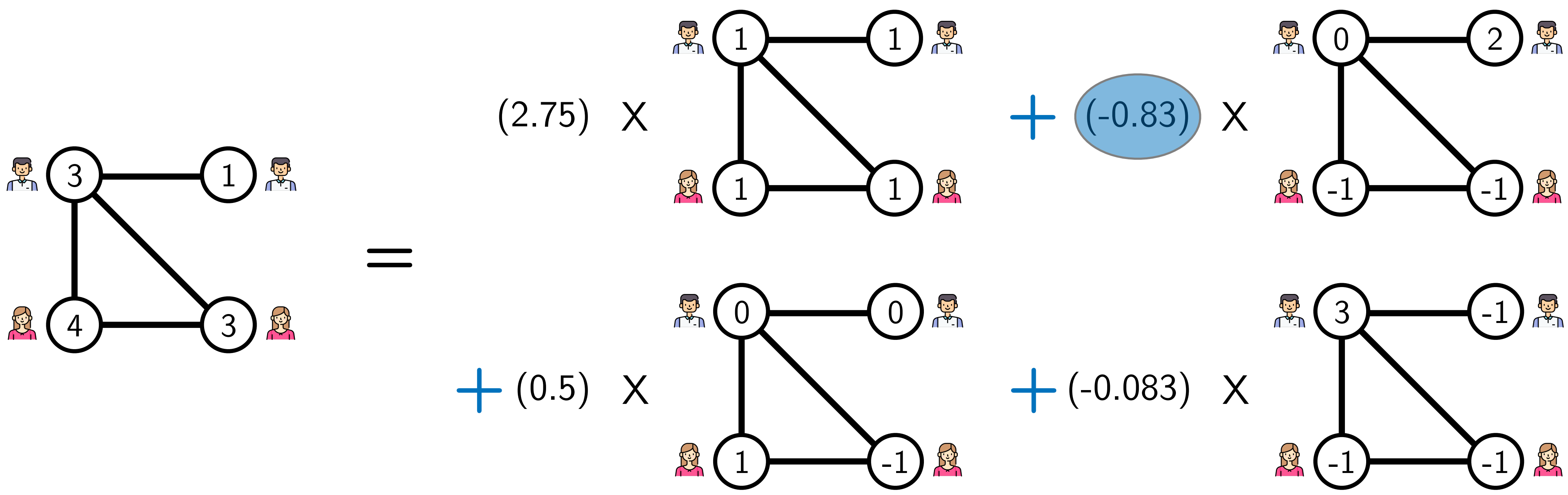
GFT:

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

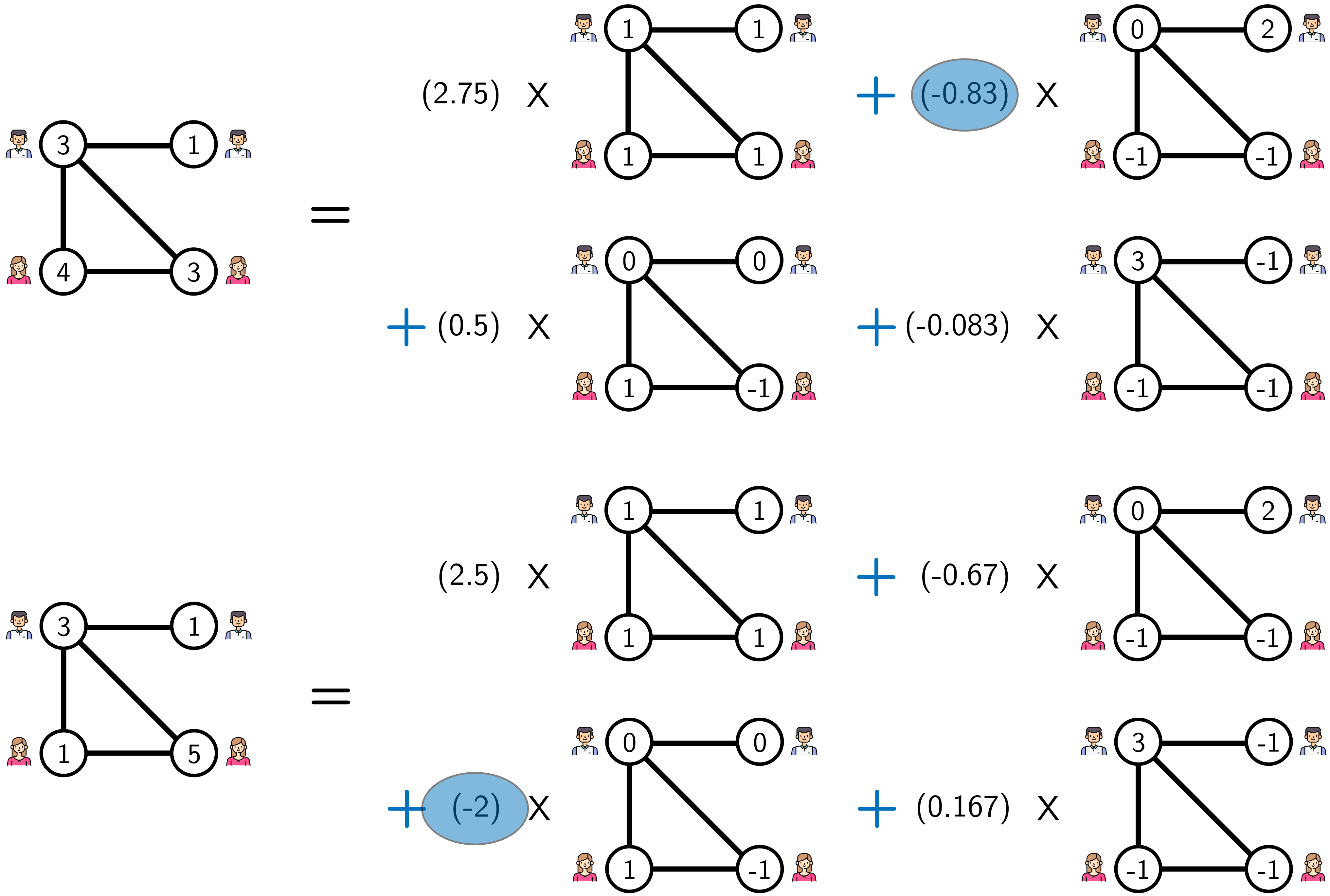




Example on movie rating



Example on movie rating



Lecture 2

- Graph signal processing: Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Representation of graph signals
- Applications

Classical frequency filtering

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

Classical frequency filtering

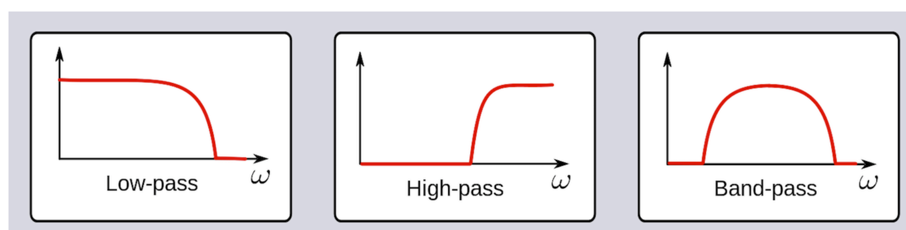
Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x)dx \qquad f(x) = \frac{1}{2\pi}\int \hat{f}(\omega)e^{j\omega x}d\omega$

$$\boxed{f} \xrightarrow{\text{FT}} \boxed{\hat{f}(\omega)} \xrightarrow{\hat{g}(\omega)} \boxed{\hat{g}(\omega)\hat{f}(\omega)} \xrightarrow{\text{IFT}} \boxed{f * g}$$

Classical frequency filtering

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

$$f \xrightarrow{\text{FT}} \hat{f}(\omega) \xrightarrow{\hat{g}(\omega)} \hat{g}(\omega)\hat{f}(\omega) \xrightarrow{\text{IFT}} f * g$$



Graph spectral filtering

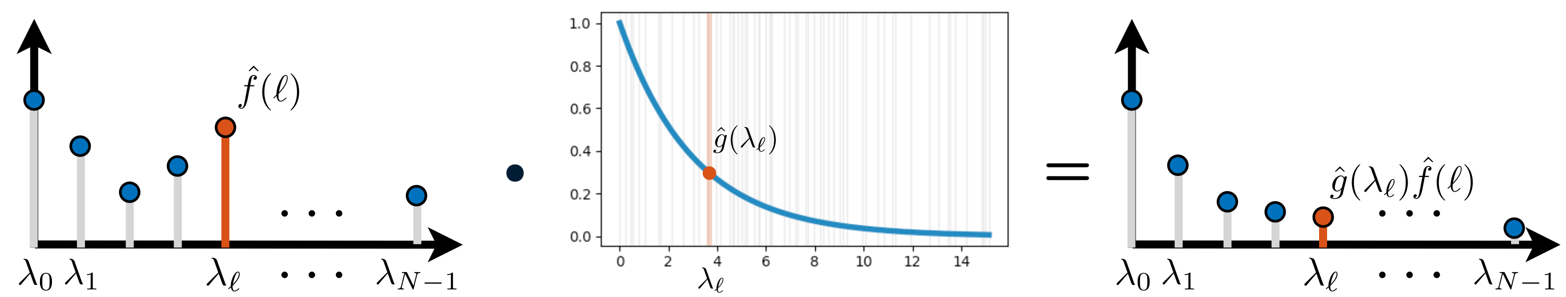
GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$

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

Graph spectral filtering

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

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



Graph spectral filtering

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

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

$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

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

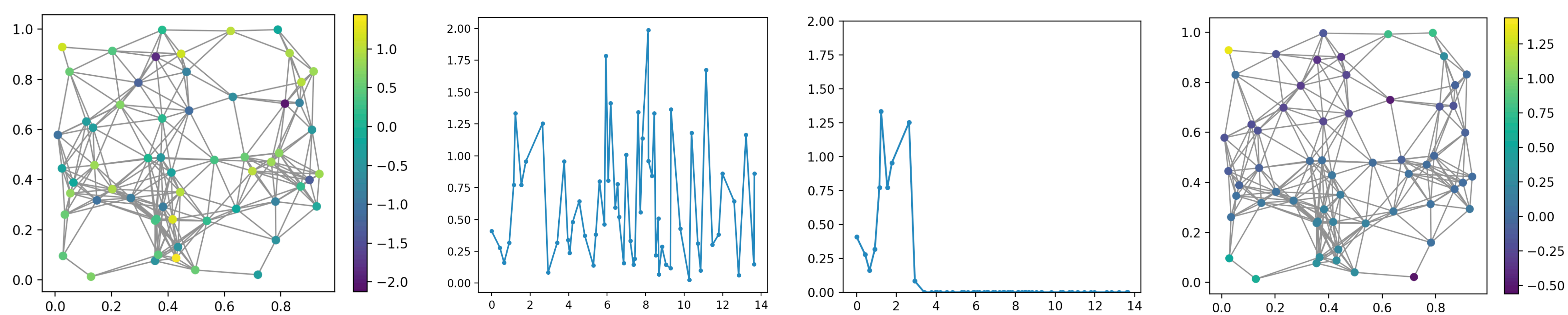
Graph spectral filtering

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

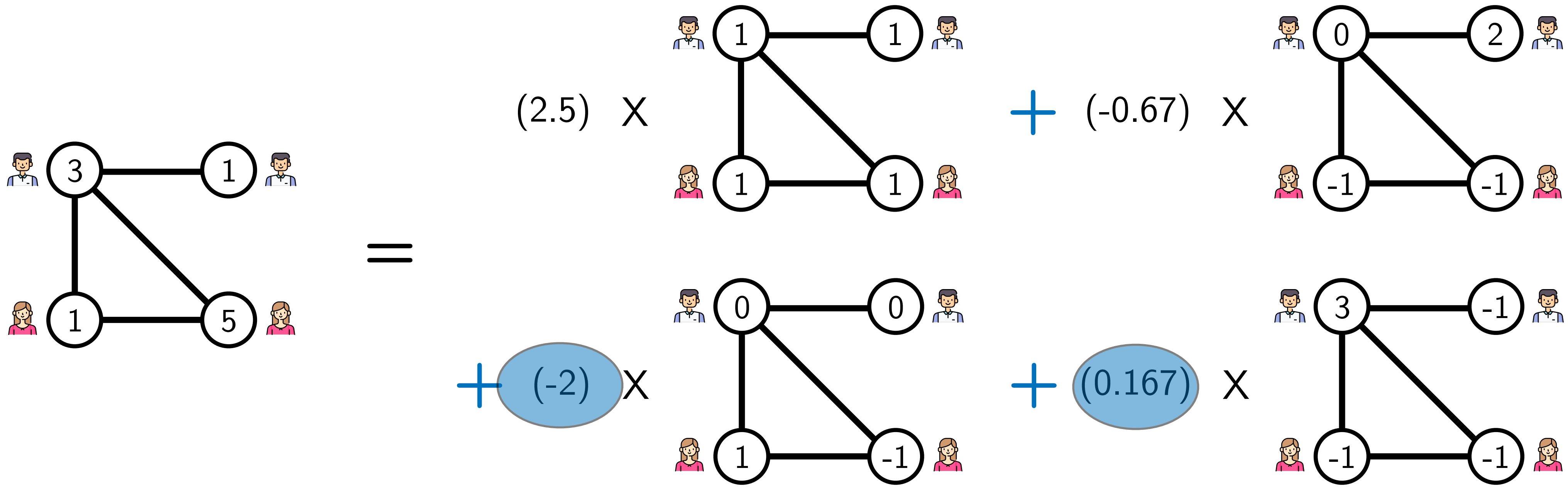
$$\boxed{f} \xrightarrow{\text{GFT}} \boxed{\chi^T f} \xrightarrow{\hat{g}(\Lambda)} \boxed{\hat{g}(\Lambda)\chi^T f} \xrightarrow{\text{IGFT}} \boxed{\underline{\chi \hat{g}(\Lambda)} \chi^T f}$$

$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

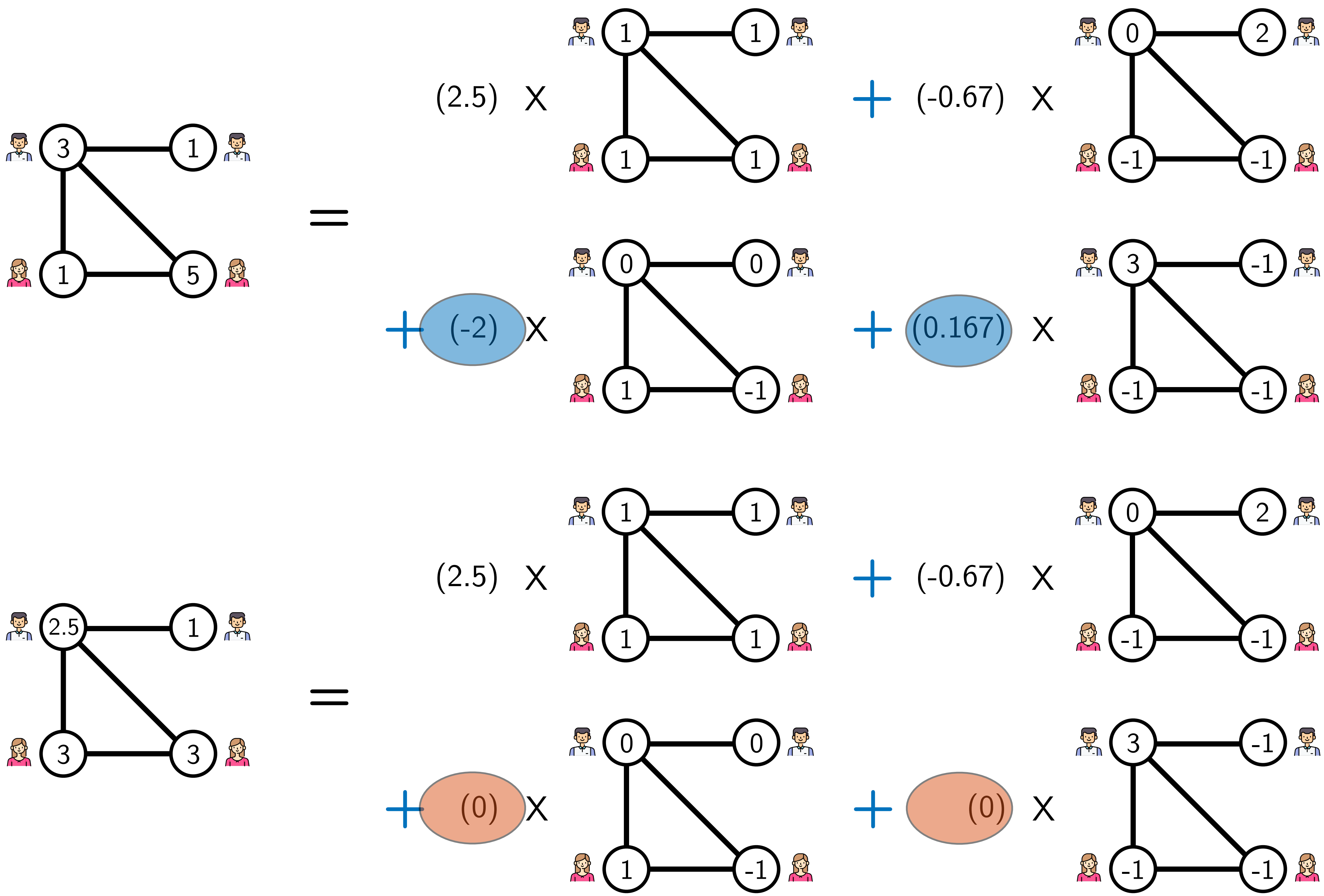
$\hat{g}(L)$: **function of L!**



Example on movie rating



Example on movie rating



Graph spectral filtering

- Filters can be designed as functions of graph Laplacian

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

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

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

Graph spectral filtering

- Filters can be designed as functions of graph Laplacian

$$\boxed{f} \xrightarrow{\text{GFT}} \boxed{\chi^T f} \xrightarrow{\hat{g}(\Lambda)} \boxed{\hat{g}(\Lambda)\chi^T f} \xrightarrow{\text{IGFT}} \boxed{\underbrace{\chi\hat{g}(\Lambda)\chi^T}_{\hat{g}(L)\textbf{: function of L!}} f}$$

- Important properties can be achieved by properly defining $\hat{g}(L)$, such as localisation of filters

- Closely related to kernels and regularisation on graphs

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

Graph Laplacian revisited

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

The Laplacian L is a difference operator: $Lf = \chi \Lambda \chi^T f$

$$
\begin{tikzcd}
\boxed{f} \arrow[r, "\text{GFT}"] & \boxed{\chi^T f} \arrow[r, "\hat{g}(\Lambda) = \Lambda"] & \boxed{\Lambda \chi^T f} \arrow[r, "\text{IGFT}"] & \boxed{\chi \Lambda \chi^T f}
\end{tikzcd}
$$

$$
\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}
$$

Graph Laplacian revisited

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

The Laplacian L is a difference operator: $Lf = \chi \Lambda \chi^T f$

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

$$\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}$$

The Laplacian operator filters the signal in the spectral domain with a high-pass filter

The Laplacian quadratic form: $f^T L f = ||L^{\frac{1}{2}} f||_2^2 = ||\chi \Lambda^{\frac{1}{2}} \chi^T f||_2^2$

Example on signal denoising

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

Example on signal denoising

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

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

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

Example on signal denoising

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

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

Example on signal denoising

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

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

data fitting term

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

"smoothness" assumption

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

Example on signal denoising

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

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

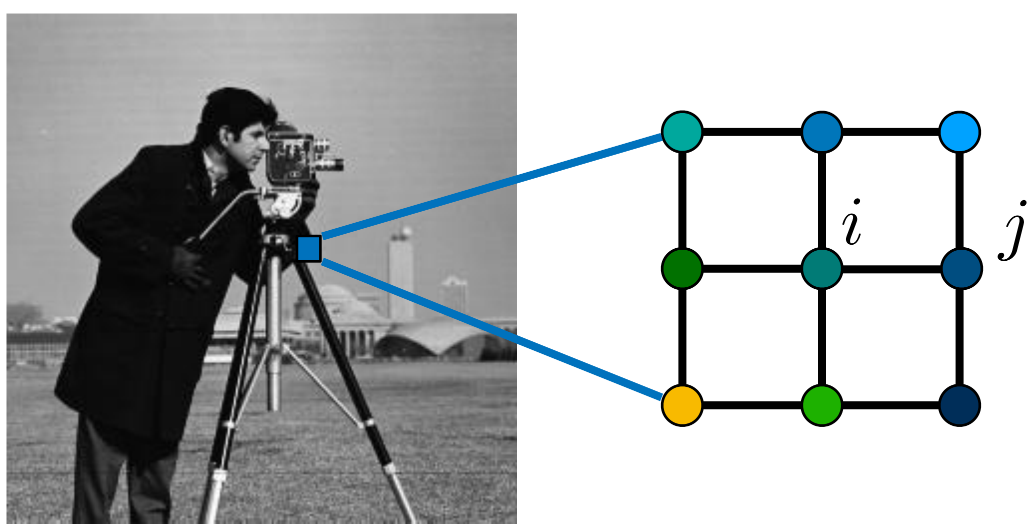
"smoothness" assumption

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

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

Example on signal denoising

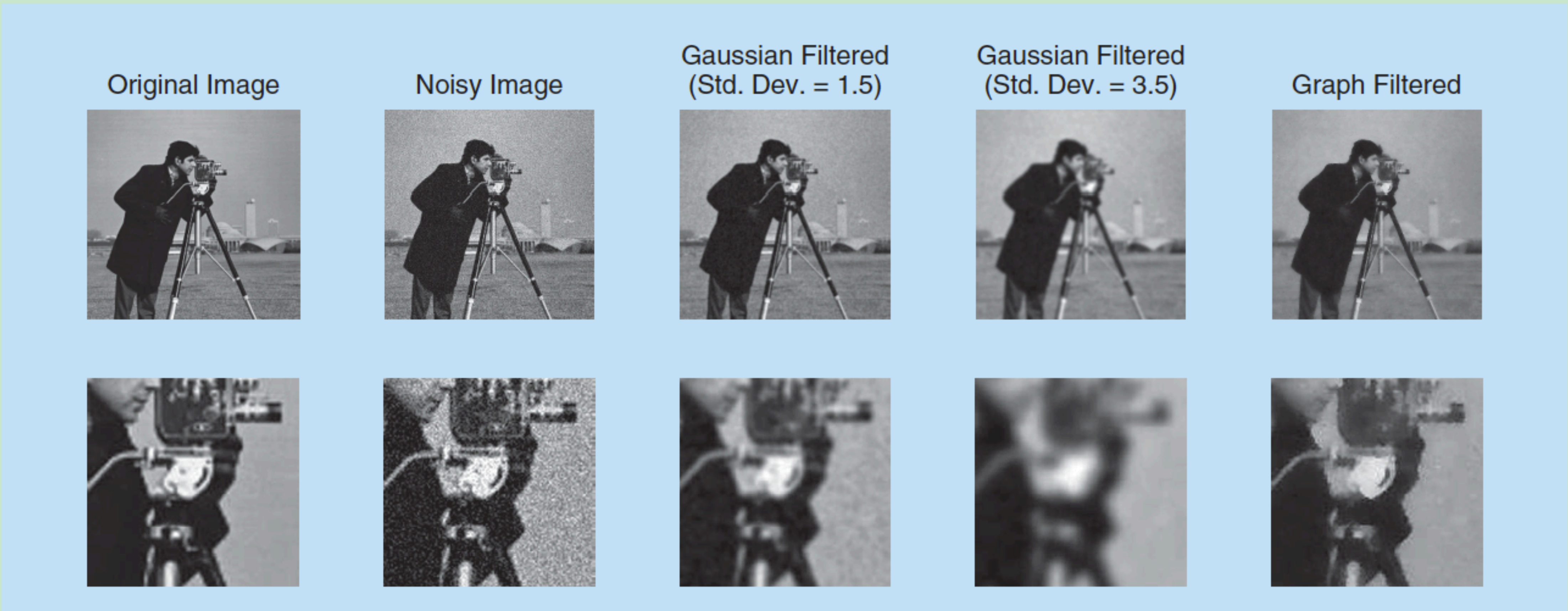
- Noisy image as observed noisy graph signal
- Regular grid graph (weights inversely proportional to pixel difference)



$$w_{ij} = \frac{1}{|f(i) - f(j)| + \epsilon}$$

Example on signal denoising

- Noisy image as observed noisy graph signal
- Regular grid graph (weights inversely proportional to pixel difference)



[FIGS2] **Image denoising via classical Gaussian filtering and graph spectral filtering.**

Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

Graph filter design

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

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

windowed kernel: windowed graph Fourier transform

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

pre-defined filters

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

Graph filter design

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

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

windowed kernel: windowed graph Fourier transform

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

pre-defined filters

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

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

adaptive (learnable) filters

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

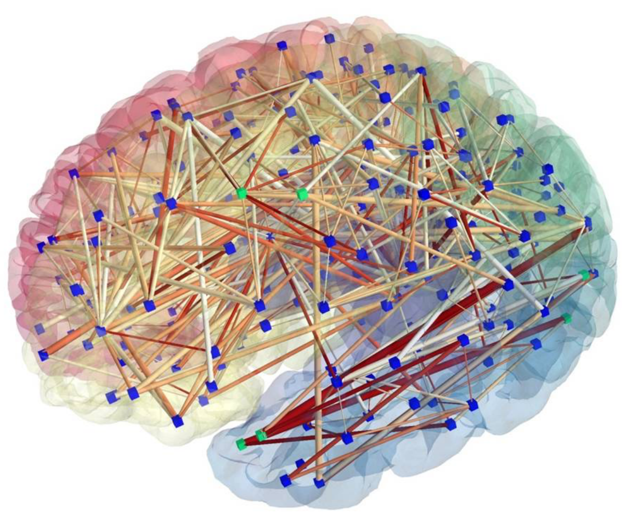
Lecture 2

- Graph signal processing: Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Representation of graph signals
- Applications

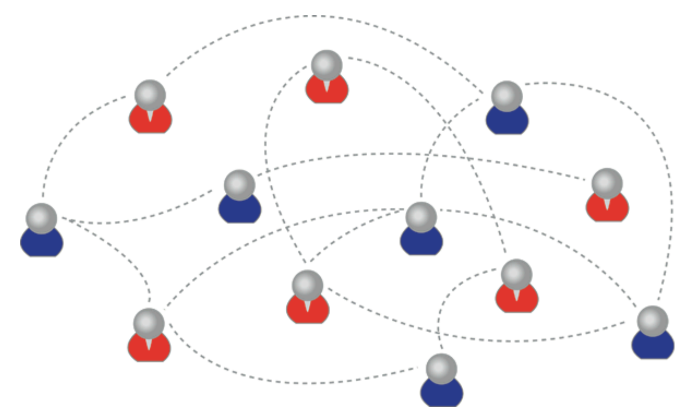
Why representation for graph signals?

image analysis (e.g., denoising, compression)

traffic analysis (e.g., mobility, congestion)



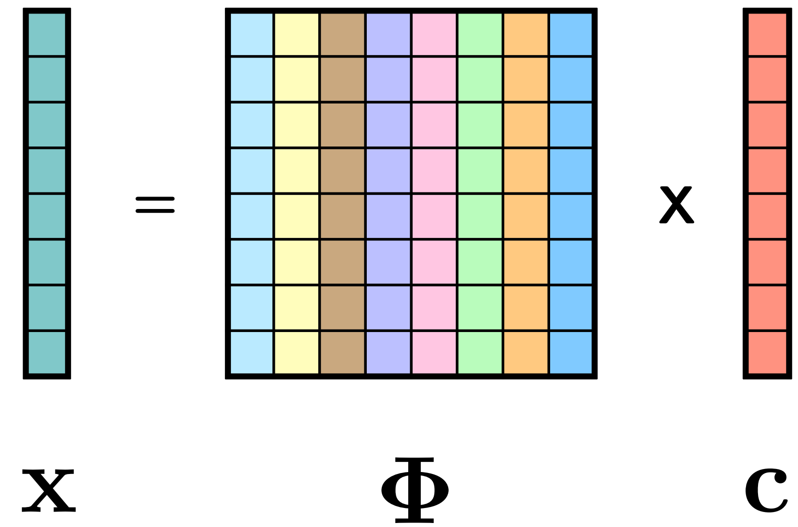
neuroscience (e.g., brain analysis)



social network analysis (e.g., community, recommendation)

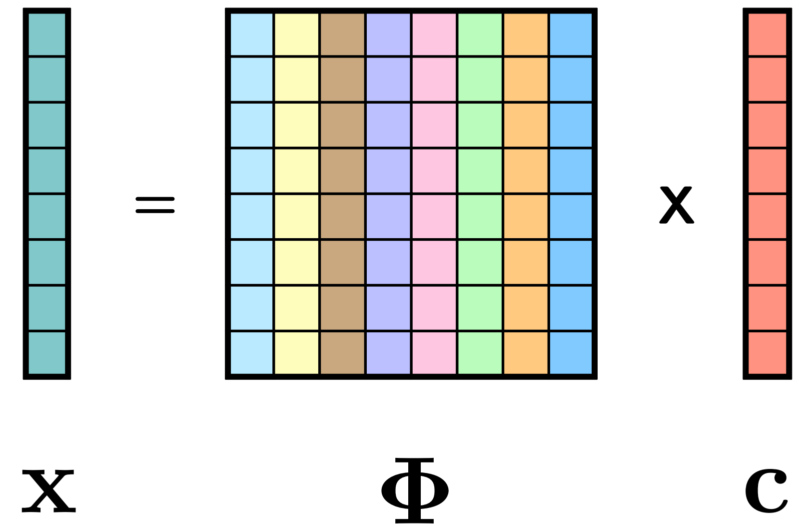
Classical vs Graph dictionaries

classical signal

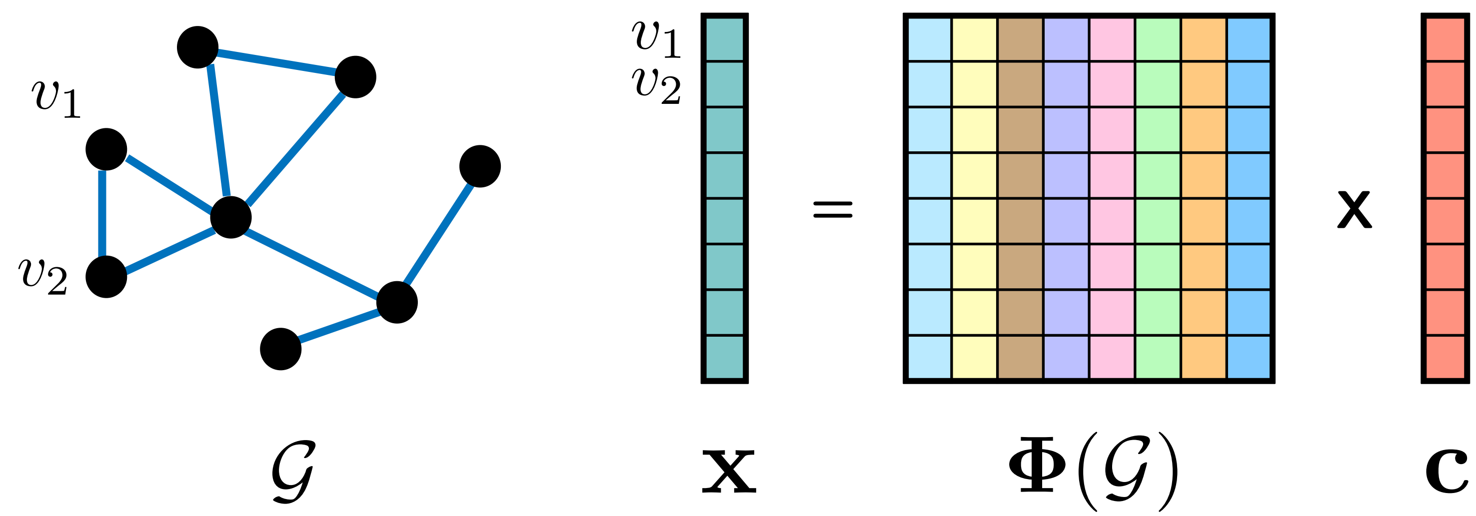


Classical vs Graph dictionaries

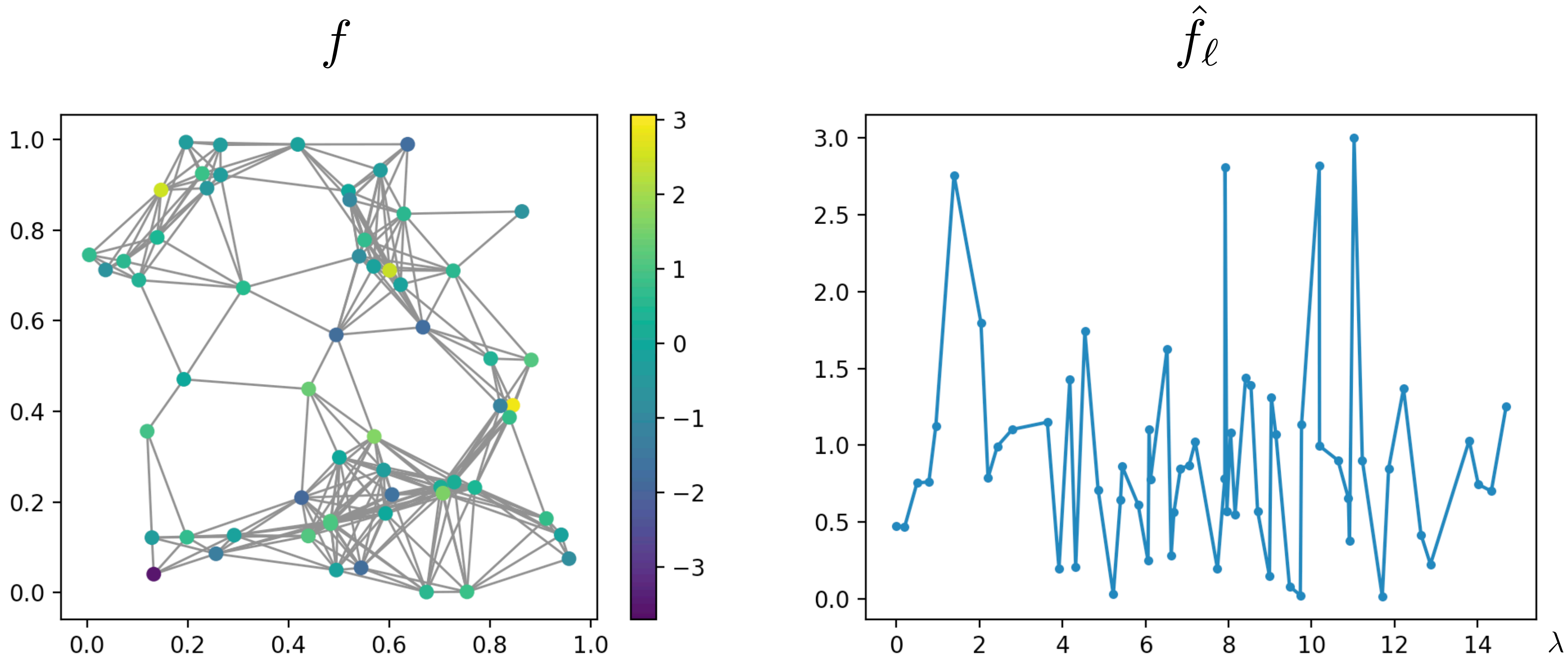
classical signal



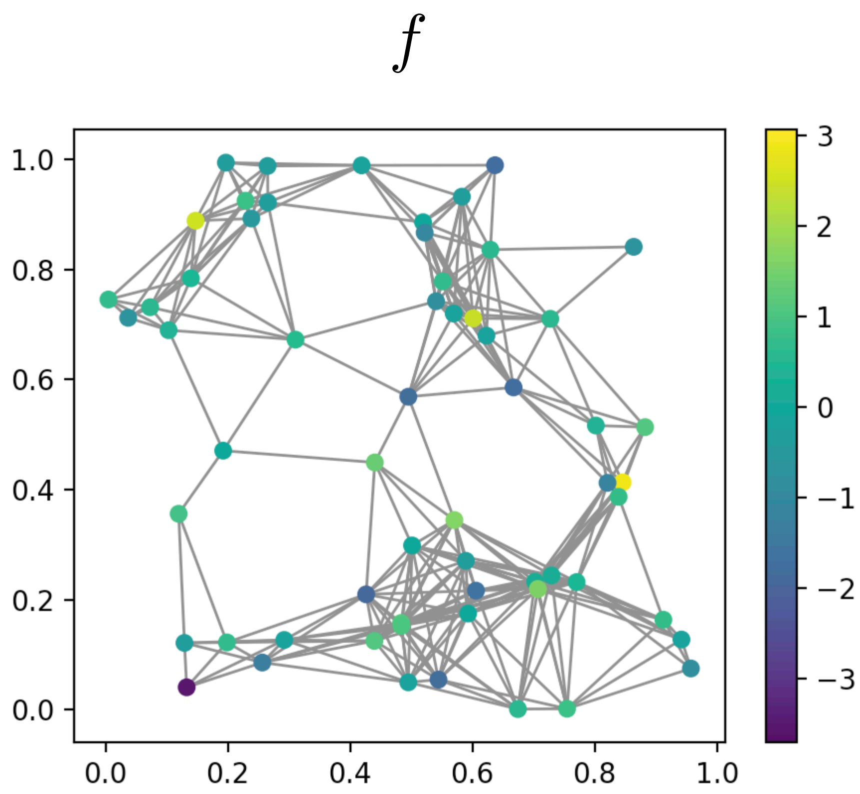
graph signal

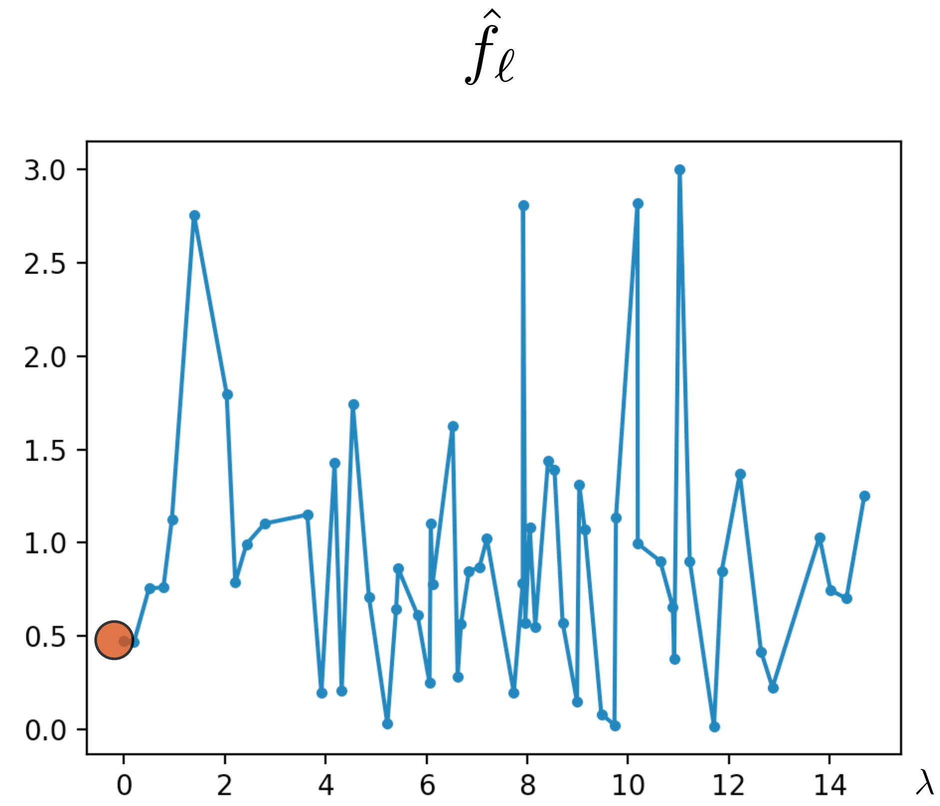


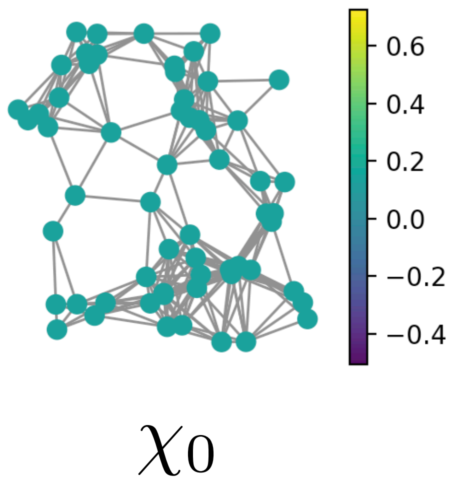
GFT provides a first graph dictionary



GFT provides a first graph dictionary



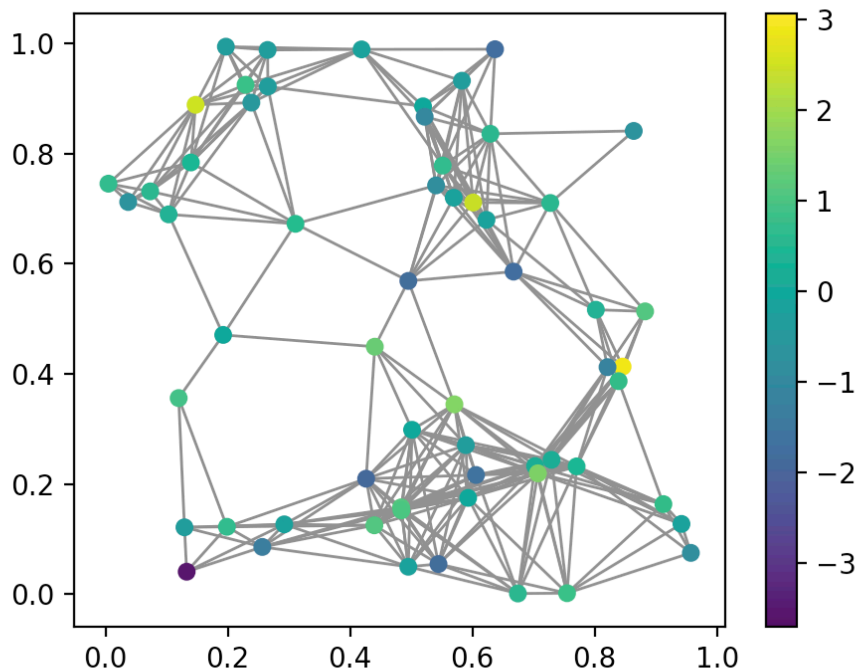




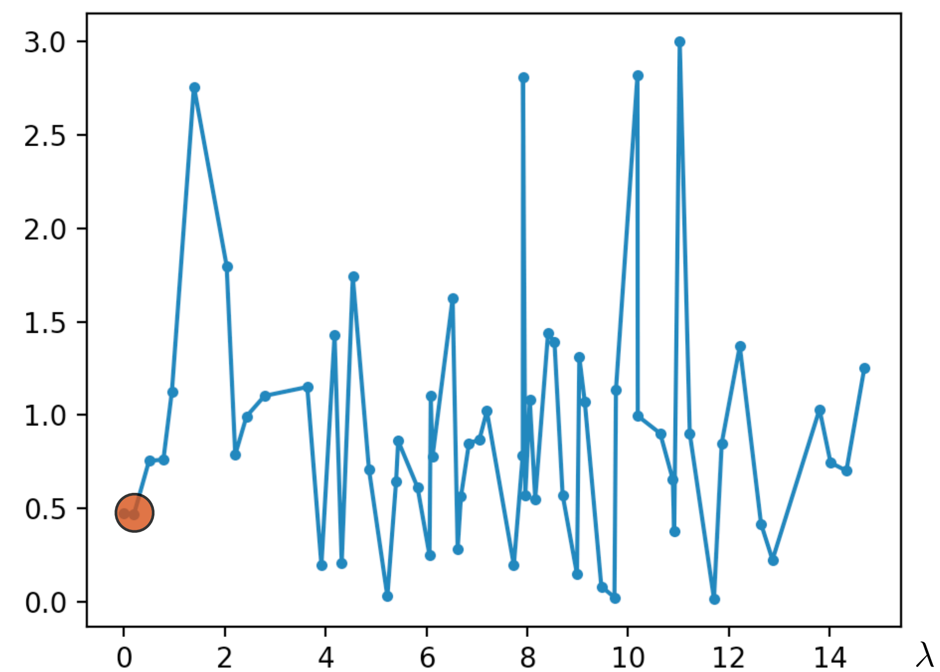
GFT atoms (corresponding to discrete frequencies)

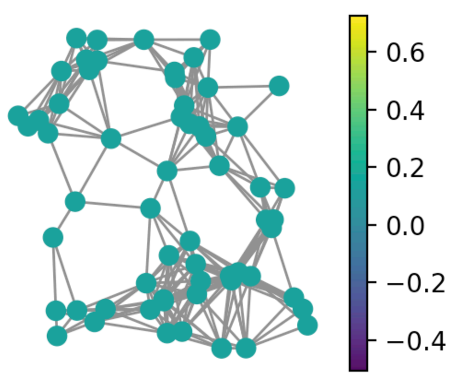
GFT provides a first graph dictionary

f

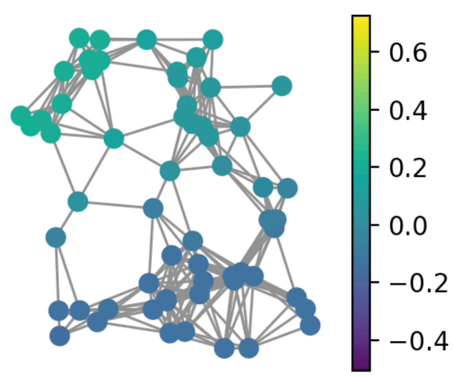


$\hat{f}_\ell$





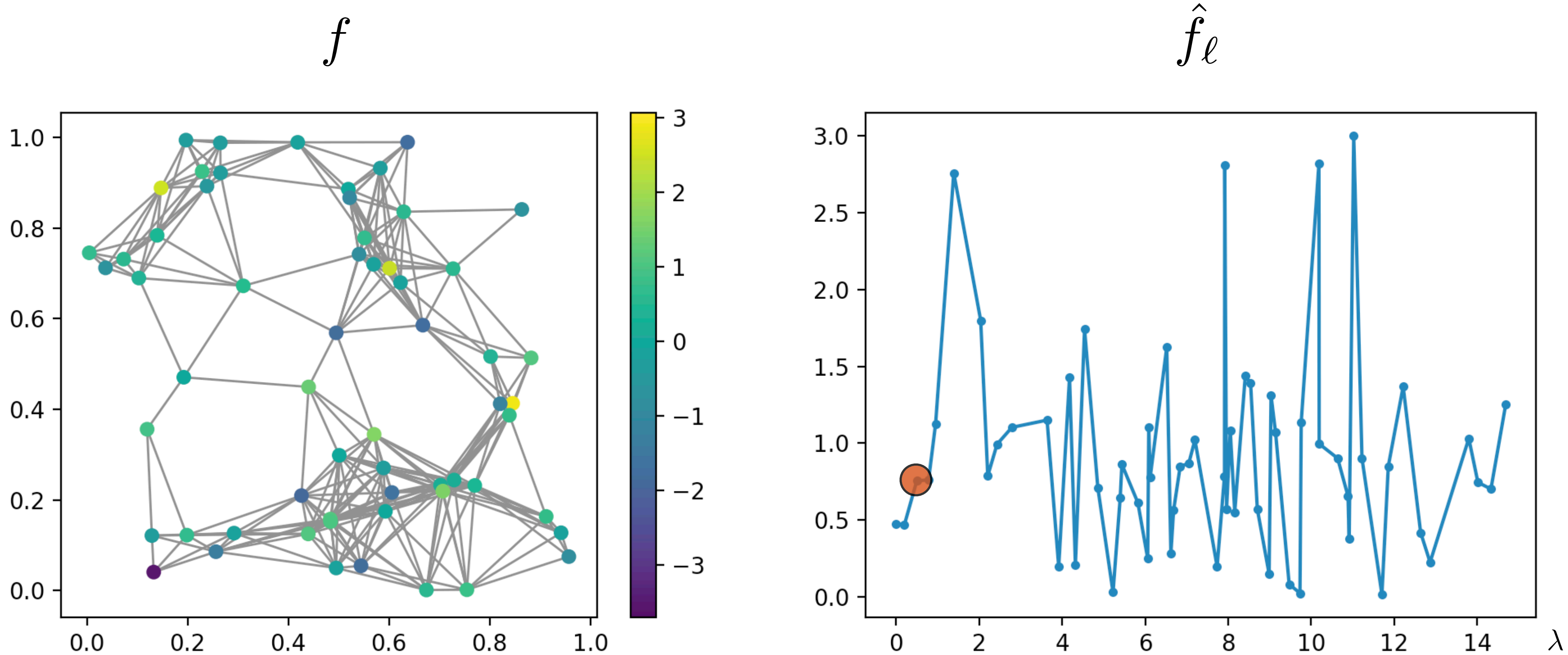
χ_0

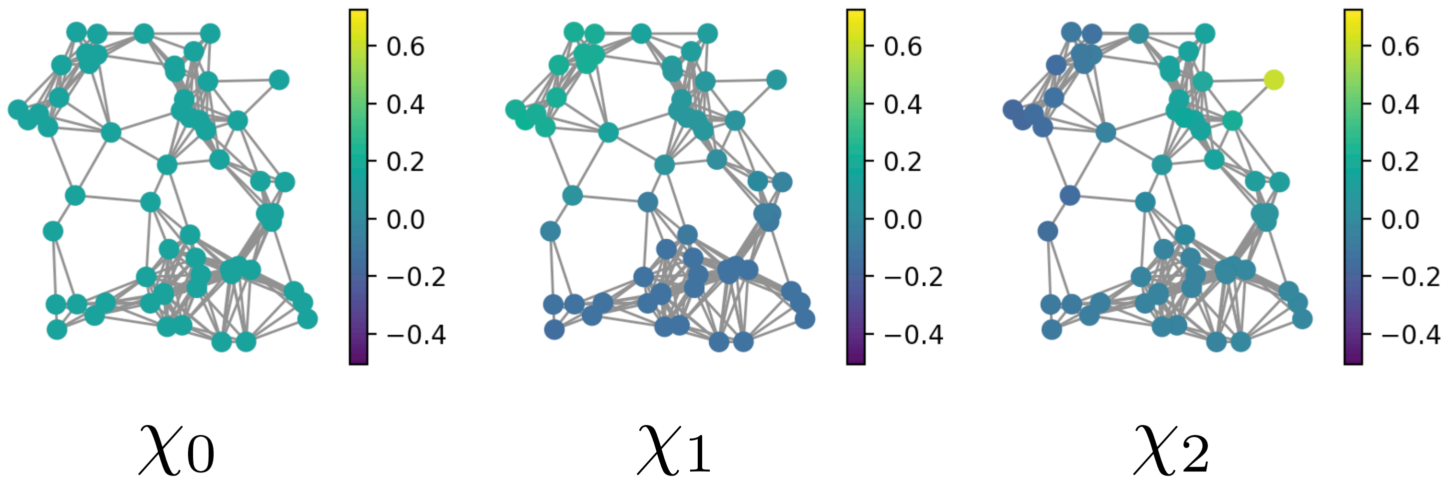


χ_1

GFT atoms (corresponding to discrete frequencies)

GFT provides a first graph dictionary

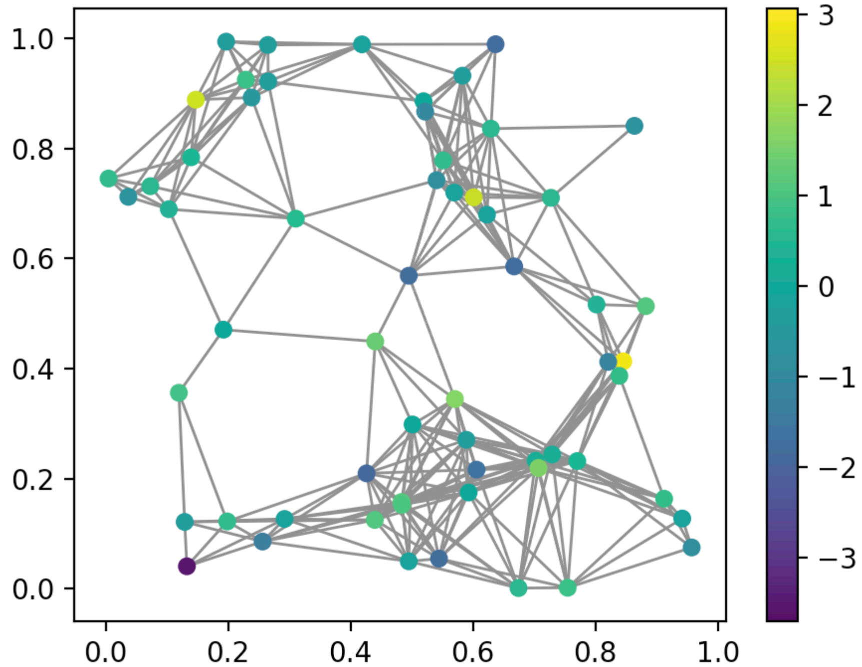




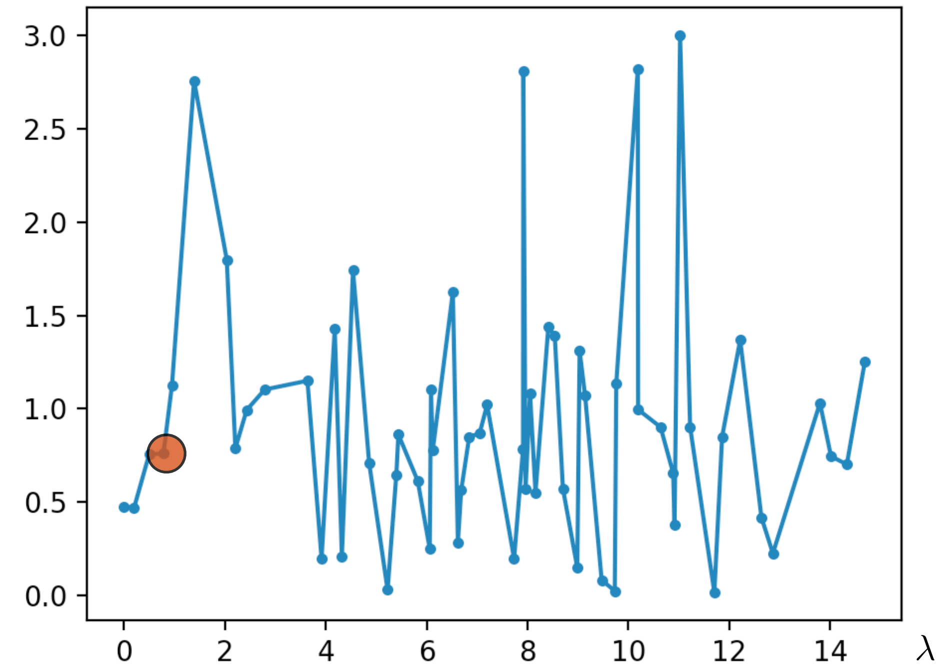
GFT atoms (corresponding to discrete frequencies)

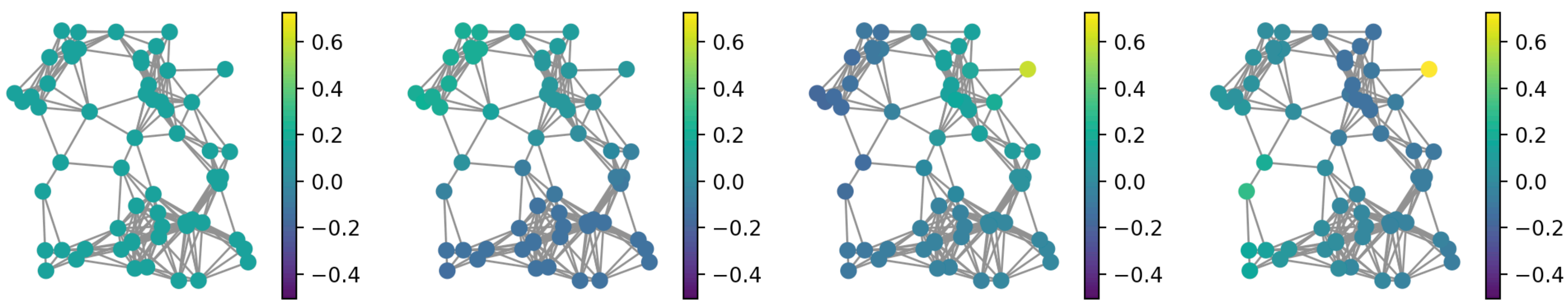
GFT provides a first graph dictionary

f



$\hat{f}_\ell$

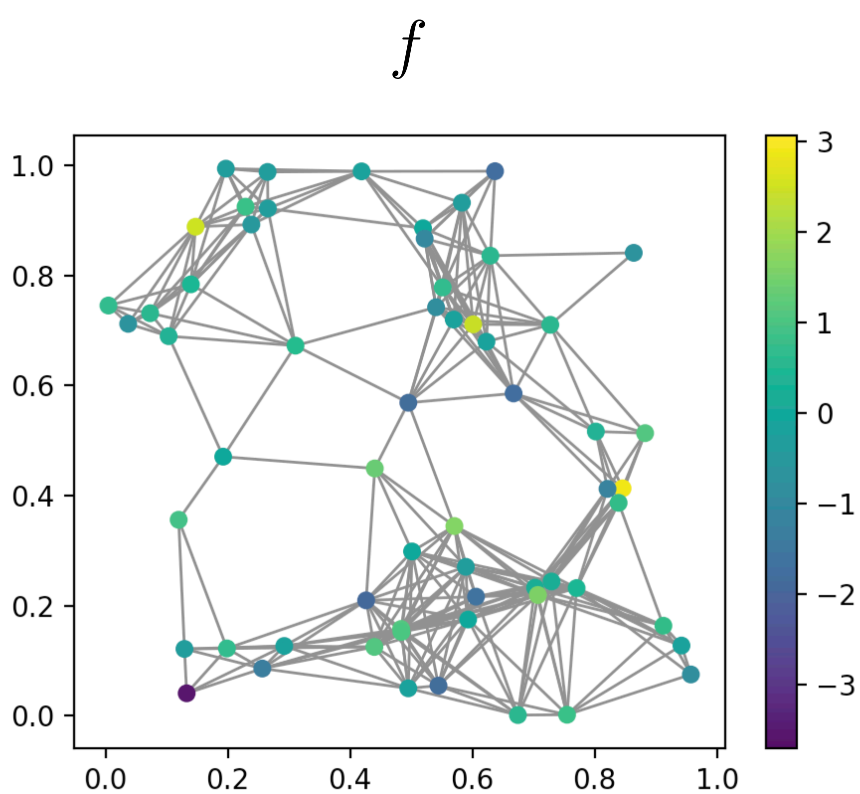


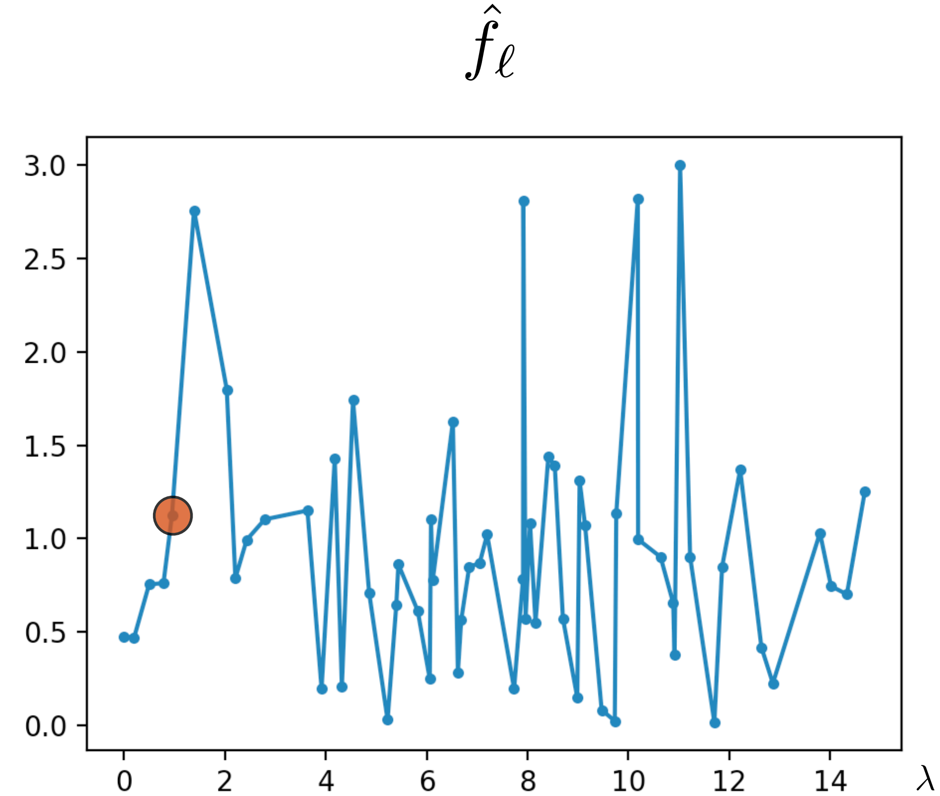


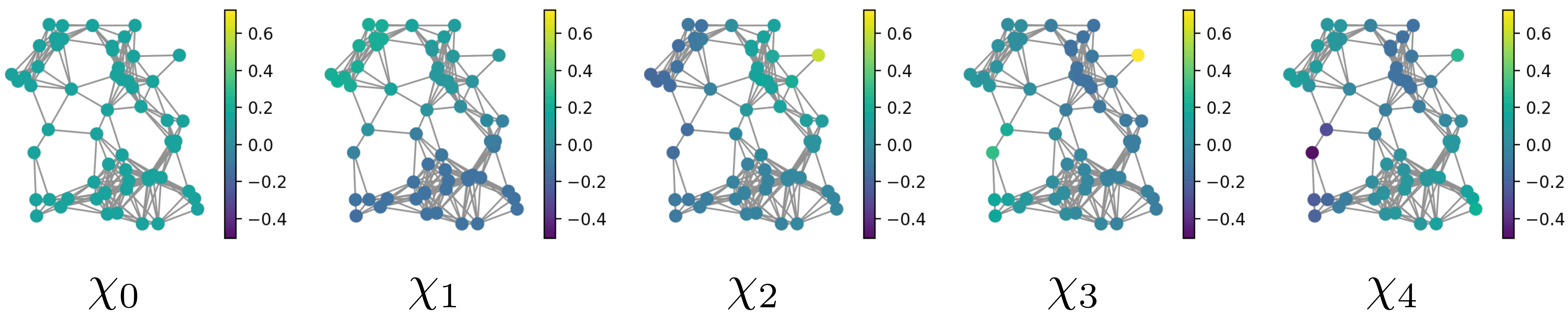
χ_0 χ_1 χ_2 χ_3

GFT atoms (corresponding to discrete frequencies)

GFT provides a first graph dictionary

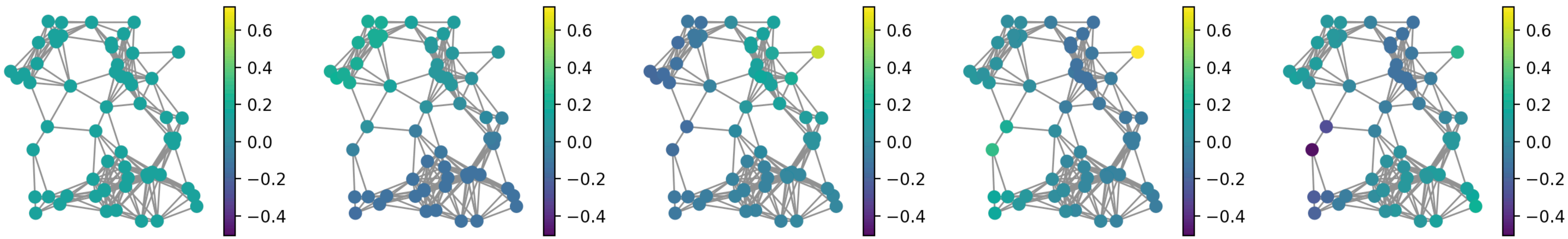






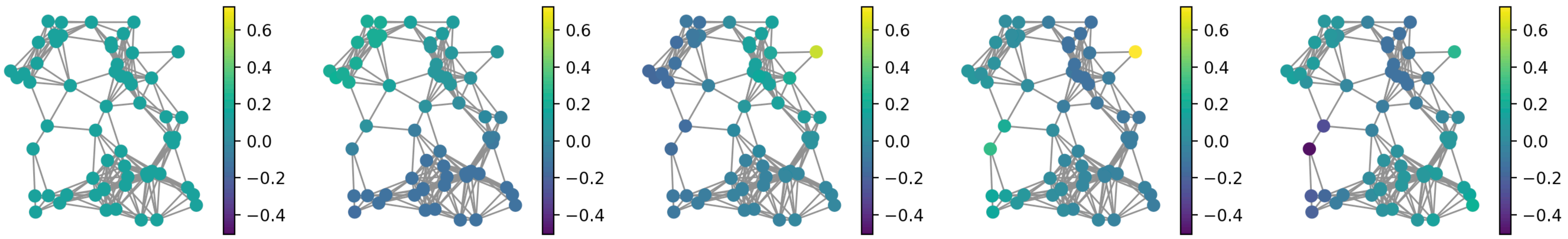
GFT atoms (corresponding to discrete frequencies)

GFT provides a first graph dictionary

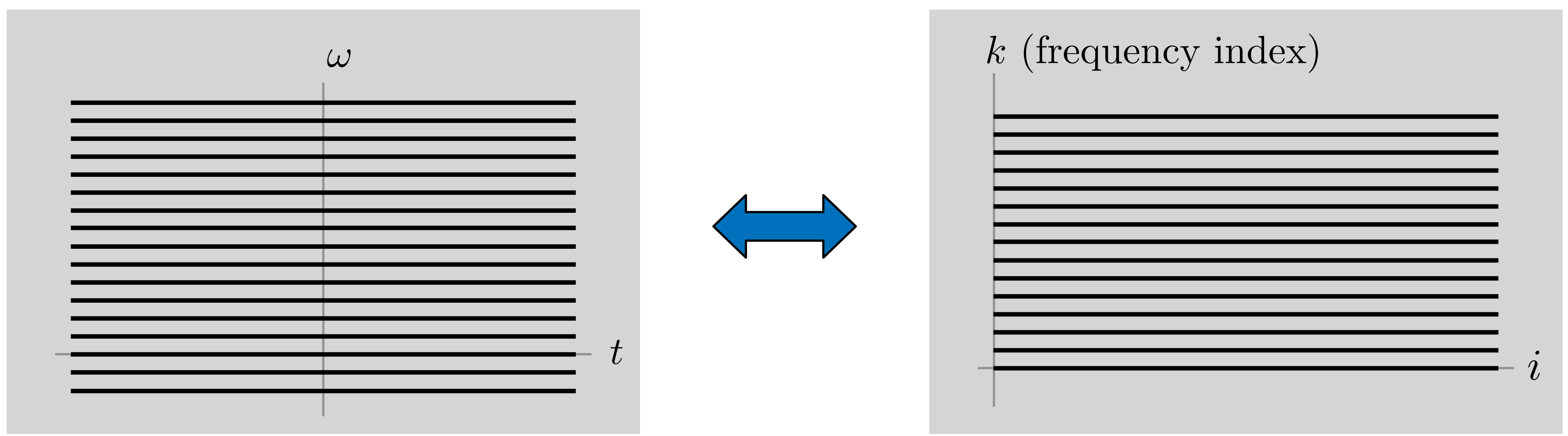


χ_0 χ_1 χ_2 χ_3 χ_4

GFT provides a first graph dictionary

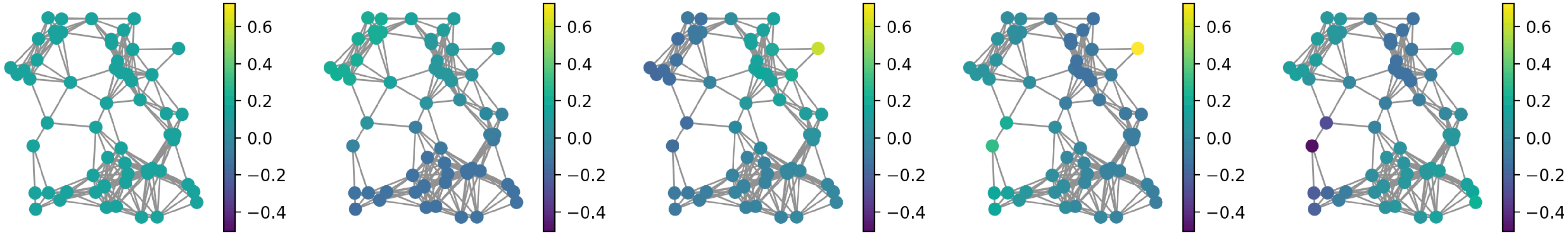


χ_0 χ_1 χ_2 χ_3 χ_4

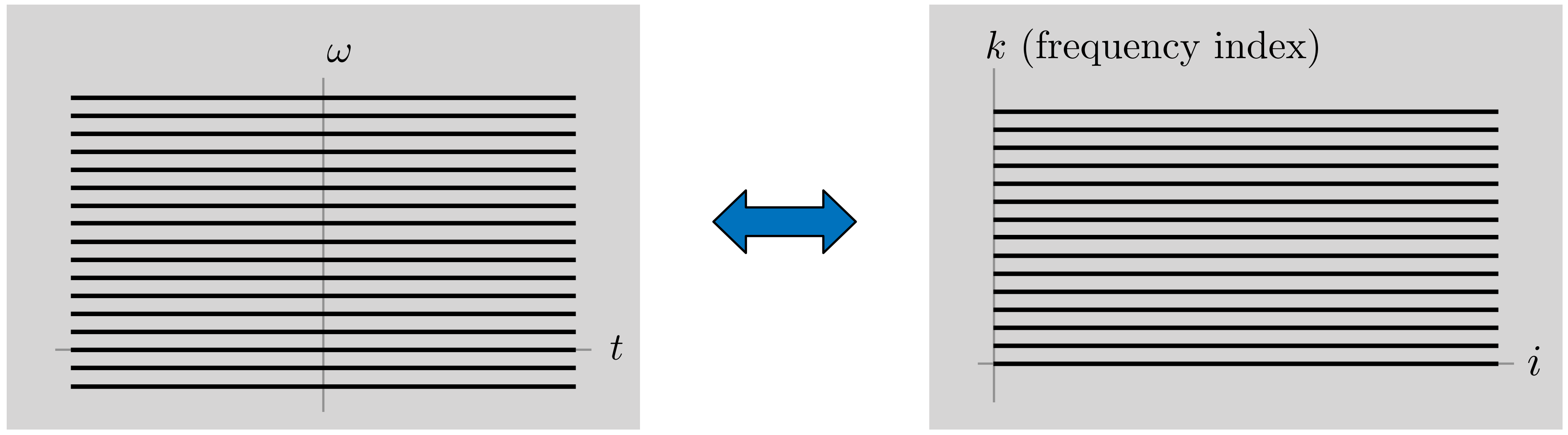


- like complex exponentials in classical FT, eigenvectors in GFT have **global** support

GFT provides a first graph dictionary



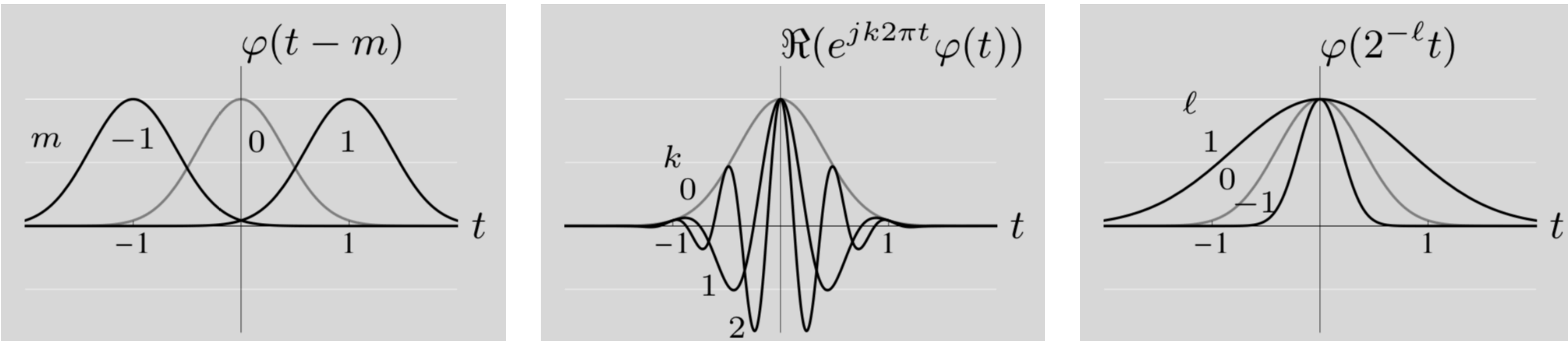
χ_0 χ_1 χ_2 χ_3 χ_4



- like complex exponentials in classical FT, eigenvectors in GFT have **global** support
- can we design **localised** atoms on graphs?

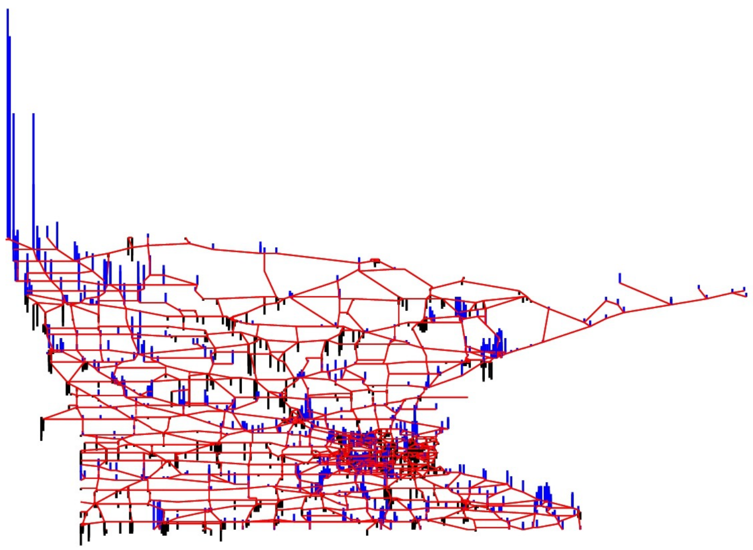
Basic operations for graph signals

basic operations in Euclidean domain



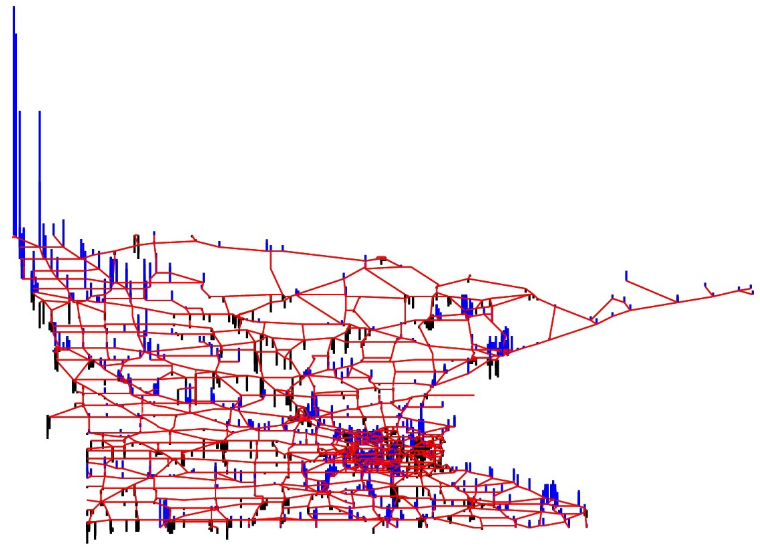
- recall that we used a set of structured functions (e.g., shifted and modulated) to produce localised items

Basic operations for graph signals



- recall that we used a set of structured functions (e.g., shifted and modulated) to produce localised items
- we need to define for graph signals the basic operations: **convolution, shift and modulation**

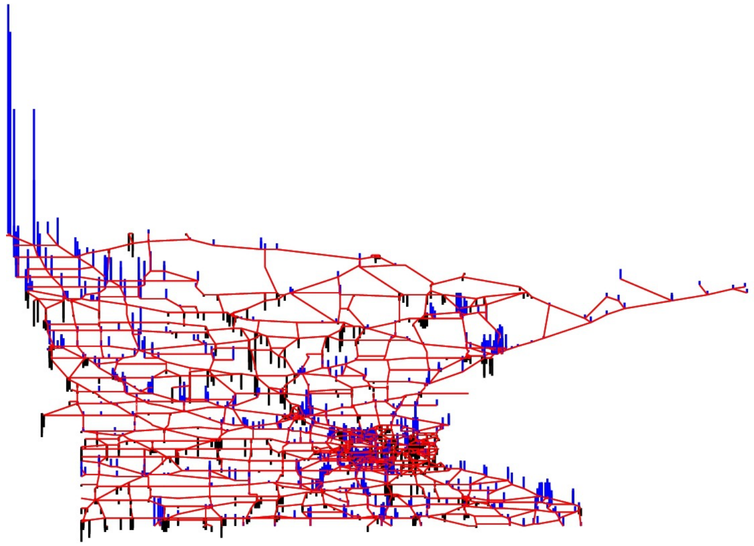
Convolution



classical convolution

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

Convolution

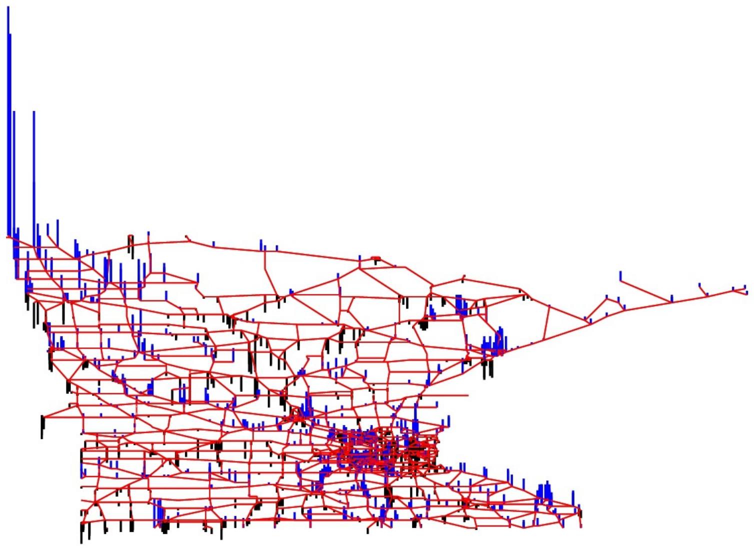


classical convolution

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

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

Convolution



classical convolution

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

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

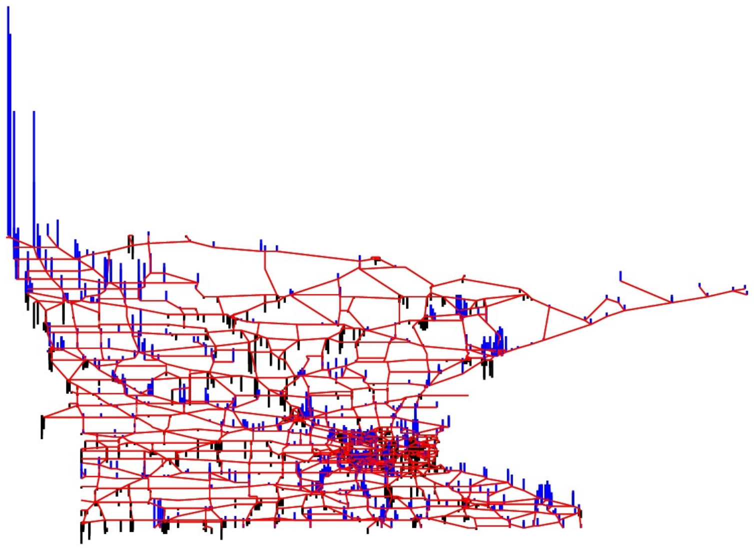
graph convolution

multiplication in graph spectral domain

$$\widehat{(f * g)}(\lambda) = (\hat{f} \circ \hat{g})(\lambda)$$

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

Convolution



classical convolution

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

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

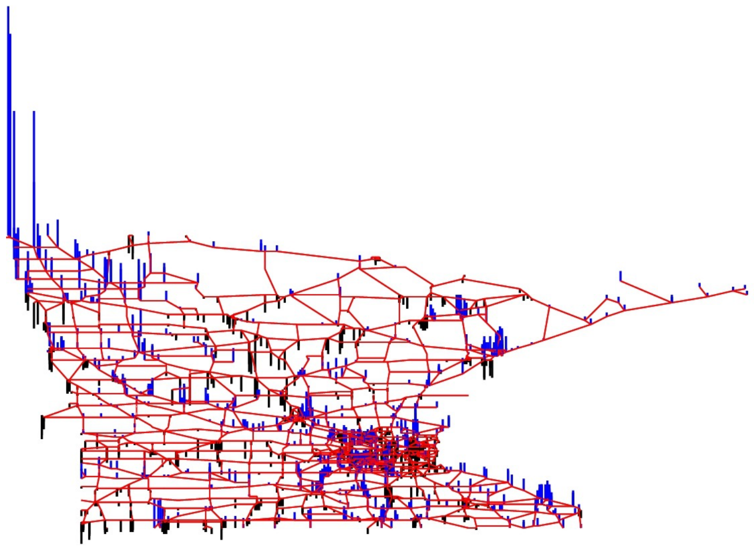
graph convolution **multiplication in graph spectral domain**

$$\widehat{(f * g)}(\lambda) = (\hat{f} \circ \hat{g})(\lambda)$$

$$(f * g)(n) := \sum_{\ell=0}^{N-1} \hat{f}(\ell) \hat{g}(\ell) \chi_\ell(n)$$

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

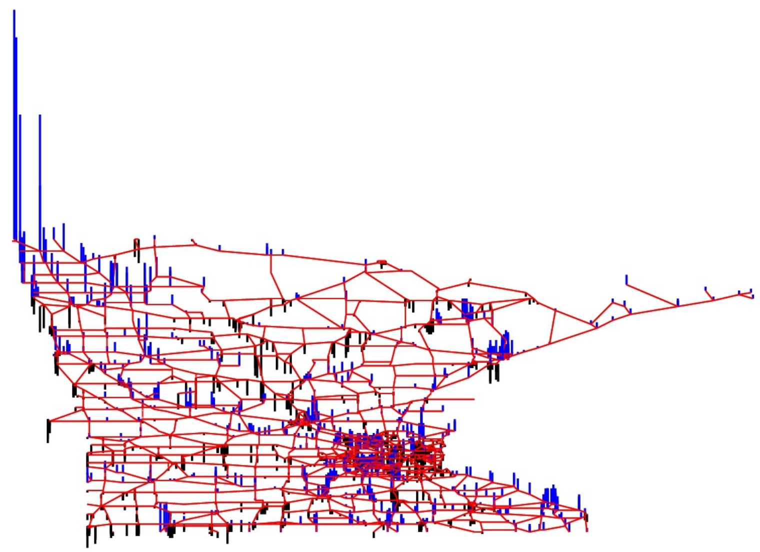
Vertex-domain shift



original signal

classical shift $(T_u f)(t) := f(t-u) = (f * \delta_u)(t)$

Vertex-domain shift



original signal

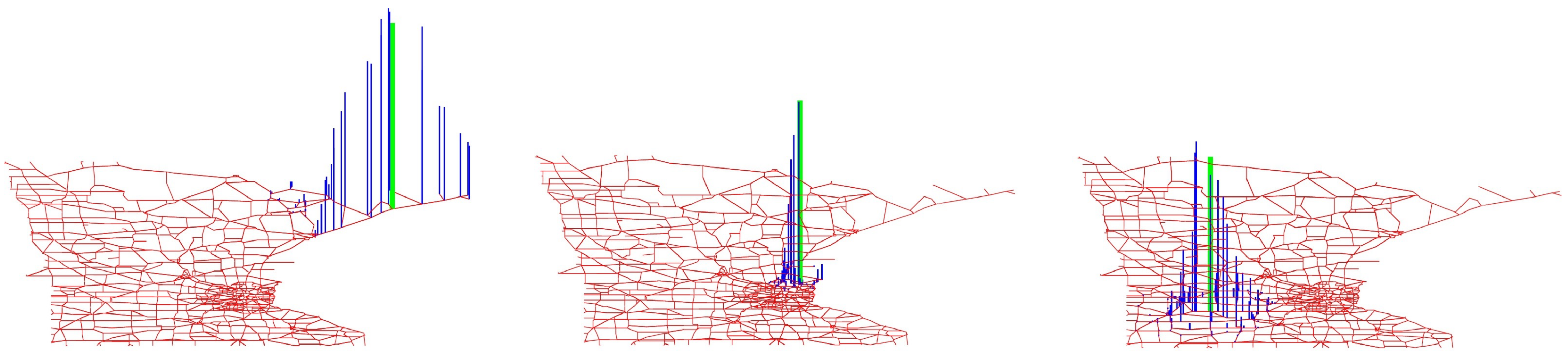
classical shift $(T_u f)(t) := f(t-u) = (f * \delta_u)(t)$

graph shift **convolution with a "delta" on graph**

$$(T_i f)(n) := \sqrt{N}(f * \delta_i)(n)$$
$$= \sqrt{N} \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell^*(i)\chi_\ell(n)$$

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

Vertex-domain shift



shifted version of the signal to different centring vertex (in green)

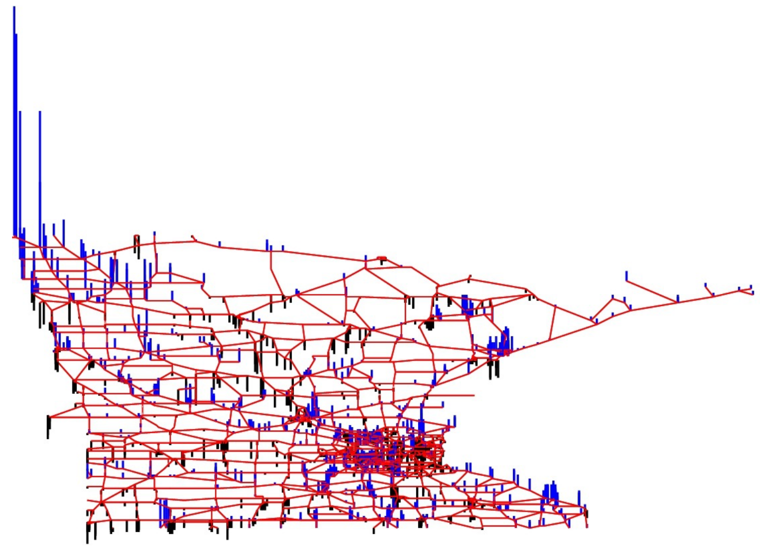
classical shift $(T_u f)(t) := f(t-u) = (f * \delta_u)(t)$

graph shift **convolution with a "delta" on graph**

$$(T_i f)(n) := \sqrt{N}(f * \delta_i)(n)$$
$$= \sqrt{N} \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell^*(i)\chi_\ell(n)$$

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

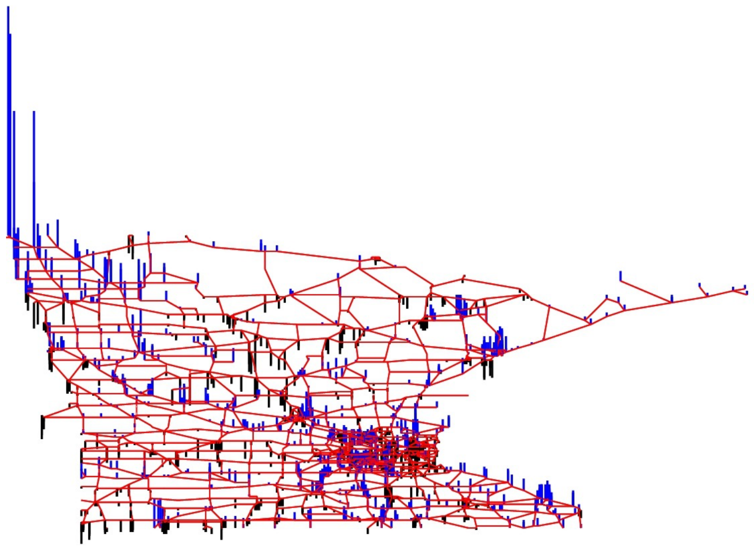
Frequency-domain shift (Modulation)



classical modulation $(M_\xi f)(t) := e^{j2\pi\xi t} f(t)$

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

Frequency-domain shift (Modulation)



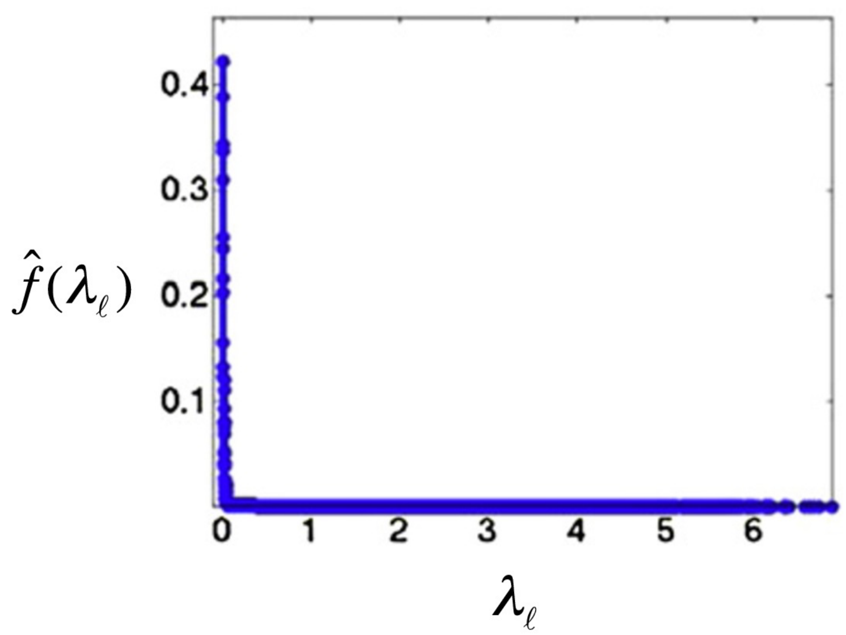
classical modulation $(M_\xi f)(t) := e^{j2\pi\xi t} f(t)$

graph modulation **multiply by a graph Laplacian eigenvector**

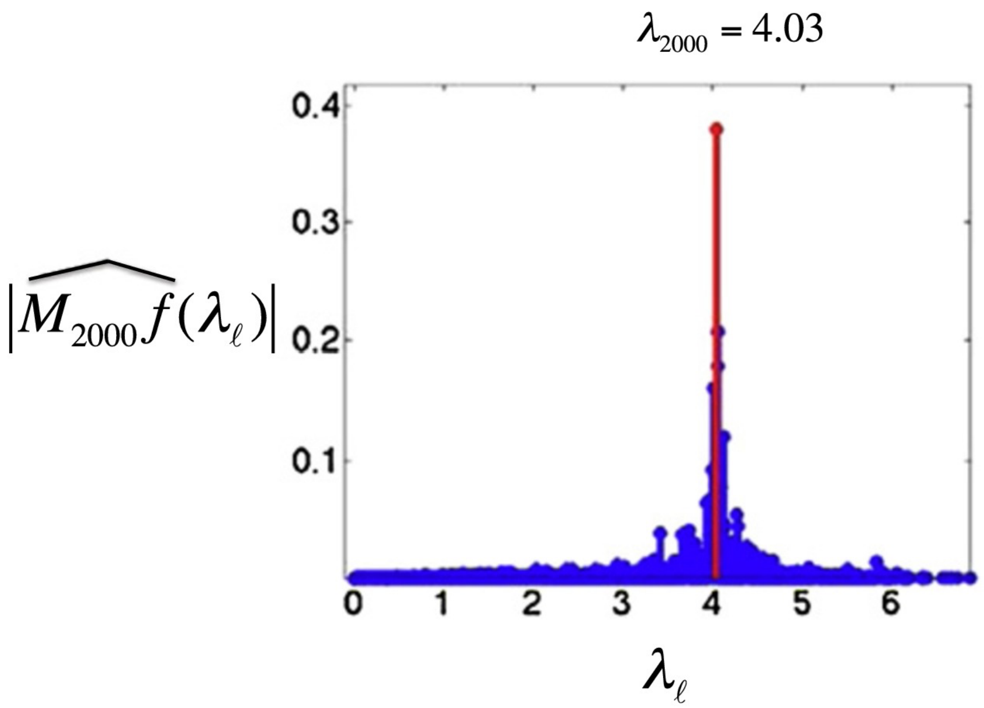
$$(M_k f)(n) := \sqrt{N} f(n) \chi_k(n)$$

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

Frequency-domain shift (Modulation)



original signal



modulated signal

classical modulation $(M_\xi f)(t) := e^{j2\pi\xi t} f(t)$

graph modulation **multiply by a graph Laplacian eigenvector**

$$(M_k f)(n) := \sqrt{N} f(n) \chi_k(n)$$

Windowed graph Fourier transform

- With the shift and modulation operators for graph signals we can define a **windowed graph Fourier transform (WGFT)**

classical windowed Fourier atom

$$g_{u,\xi}(t) := (M_\xi T_u g)(t) = e^{j2\pi\xi t} g(t-u)$$

Windowed graph Fourier transform

- With the shift and modulation operators for graph signals we can define a **windowed graph Fourier transform (WGFT)**

classical windowed Fourier atom

$$g_{u,\xi}(t) := (M_\xi T_u g)(t) = e^{j2\pi\xi t} g(t-u)$$

windowed graph Fourier atom

$$g_{i,k}(n) := (M_k T_i g)(n) = N\chi_k(n) \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$

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

Windowed graph Fourier transform

- With the shift and modulation operators for graph signals we can define a **windowed graph Fourier transform (WGFT)**

classical windowed Fourier atom

$$g_{u,\xi}(t) := (M_\xi T_u g)(t) = e^{j2\pi\xi t} g(t-u)$$

windowed graph Fourier atom

$$g_{i,k}(n) := (M_k T_i g)(n) = N\chi_k(n) \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$

Windowed graph Fourier transform

- With the shift and modulation operators for graph signals we can define a **windowed graph Fourier transform (WGFT)**

classical windowed Fourier atom

$$g_{u,\xi}(t) := (M_\xi T_u g)(t) = e^{j2\pi\xi t} g(t-u)$$

windowed graph Fourier atom

$$g_{i,k}(n) := (M_k T_i g)(n)$$

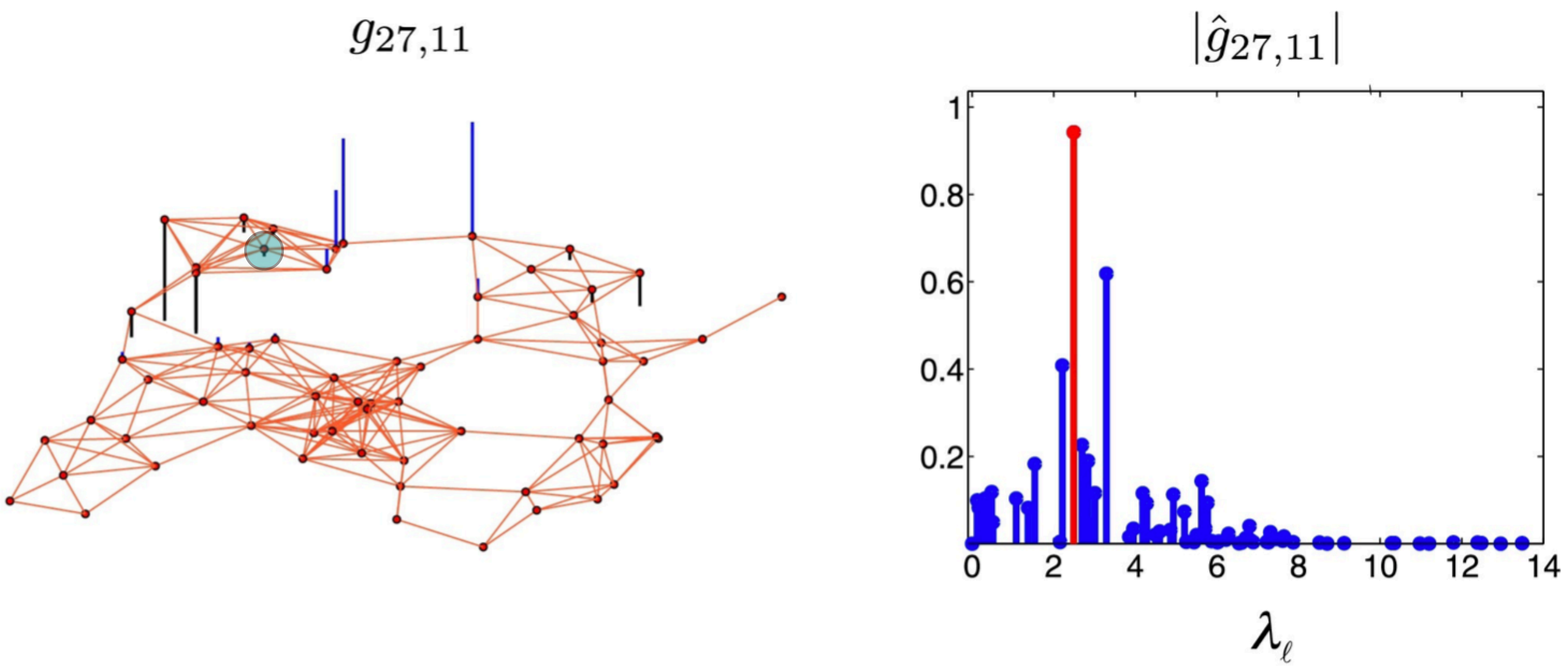
$$= N\chi_k(n) \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$

windowed graph Fourier transform

$$Sf(i,k) := \langle f, g_{i,k} \rangle$$

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

Windowed graph Fourier transform



$$\hat{g}(\lambda_\ell) = 1.45e^{-3\lambda_\ell}$$

Wavelets on graphs

- With the shift and scaling operators for graph signals we can define a **spectral graph wavelet transform (SGWT)**

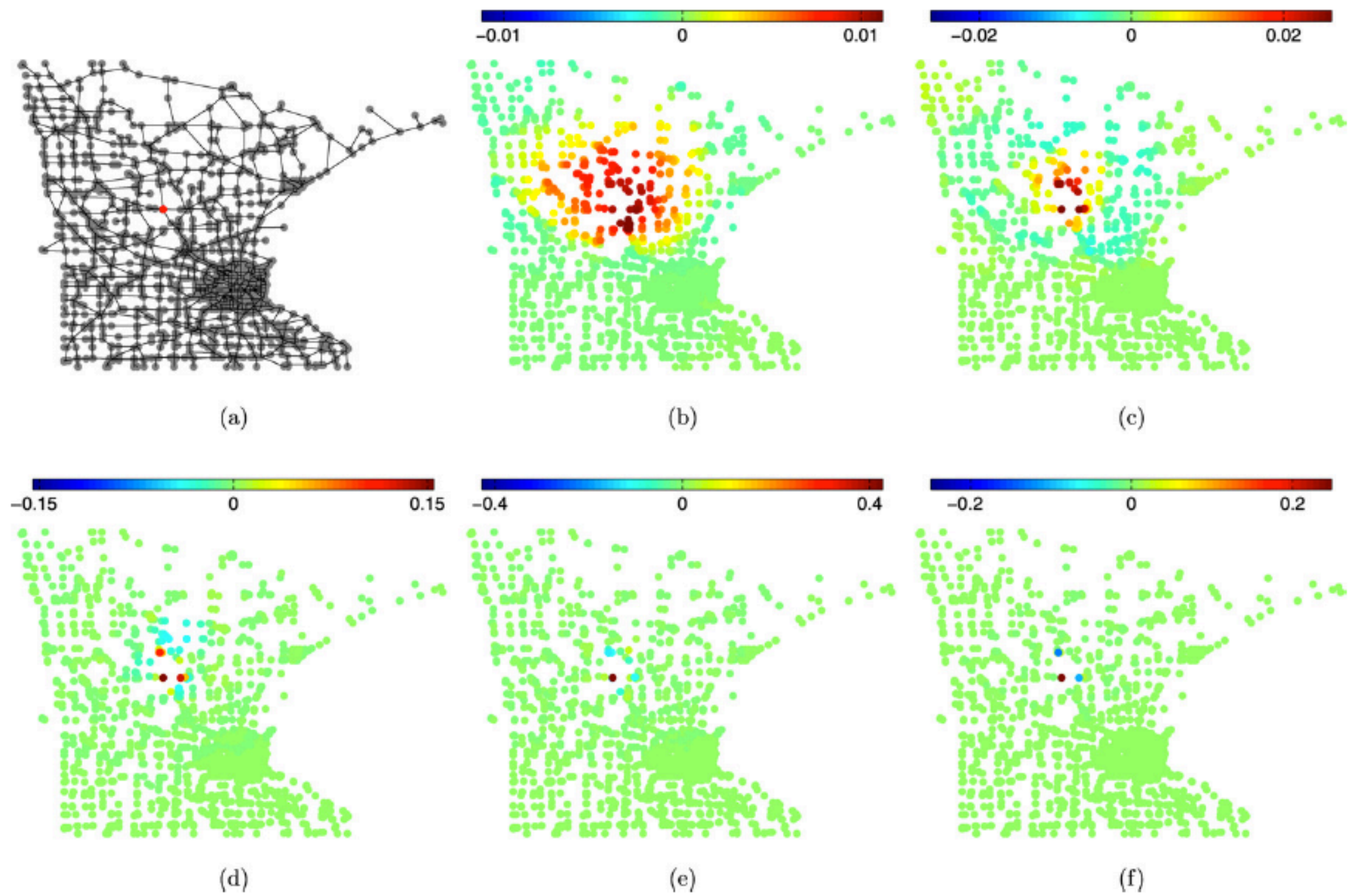


Fig. 4. Spectral graph wavelets on Minnesota road graph, with $K = 100$, $J = 4$ scales. (a) Vertex at which wavelets are centered, (b) scaling function, (c)–(f) wavelets, scales 1–4.

Wavelets on graphs

- With the shift and scaling operators for graph signals we can define a **spectral graph wavelet transform (SGWT)**

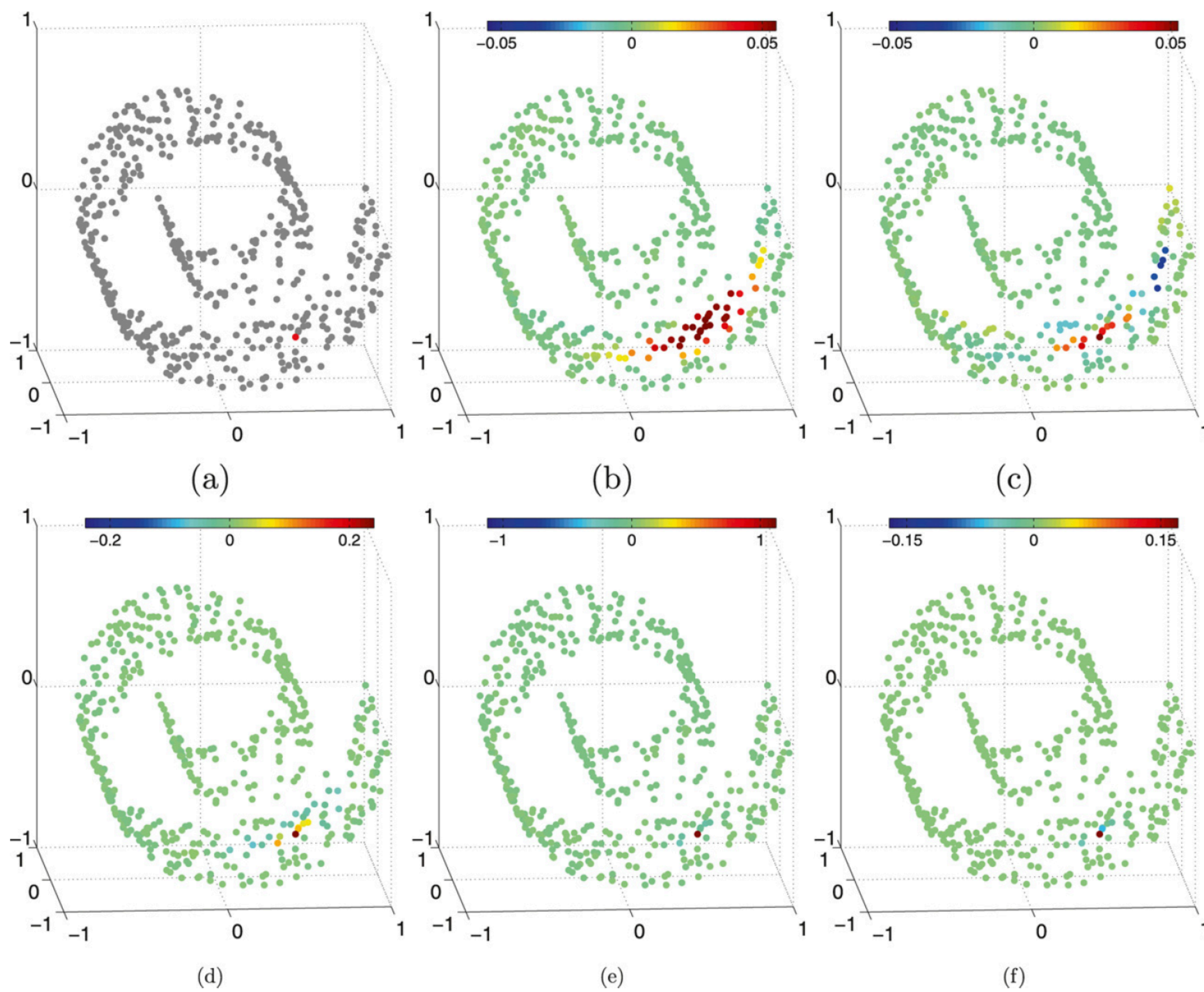


Fig. 3. Spectral graph wavelets on Swiss roll data cloud, with $J = 4$ wavelet scales. (a) Vertex at which wavelets are centered, (b) scaling function, (c)–(f) wavelets, scales 1–4.

Hammond et al., "Wavelets on graphs via spectral graph theory," ACHA, 2011.

WGFT vs SGWT atoms

WGFT atom

$$g_{i,k}(n) := (M_k T_i g)(n)$$

$$= N\chi_k(n) \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$

SGWT atom

$$\psi_{i,s}(n) := (T_i D_s g)(n)$$

$$= \sum_{\ell=0}^{N-1} \hat{g}(s\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$

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

WGFT vs SGWT atoms

WGFT atom

$g_{i,k}(n) := (M_k T_i g)(n)$

$$= N\chi_k(n) \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$

SGWT atom

$\psi_{i,s}(n) := (T_i D_s g)(n)$

$$= \sum_{\ell=0}^{N-1} \hat{g}(s\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$

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

WGFT vs SGWT atoms

WGFT atom

$g_{i,k}(n) := (M_k T_i g)(n)$

$$= N \chi_k(n) \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) \chi_\ell^*(i) \chi_\ell(n)$$

SGWT atom

$\psi_{i,s}(n) := (T_i D_s g)(n)$

$$= \sum_{\ell=0}^{N-1} \hat{g}(s\lambda_\ell) \chi_\ell^*(i) \chi_\ell(n)$$

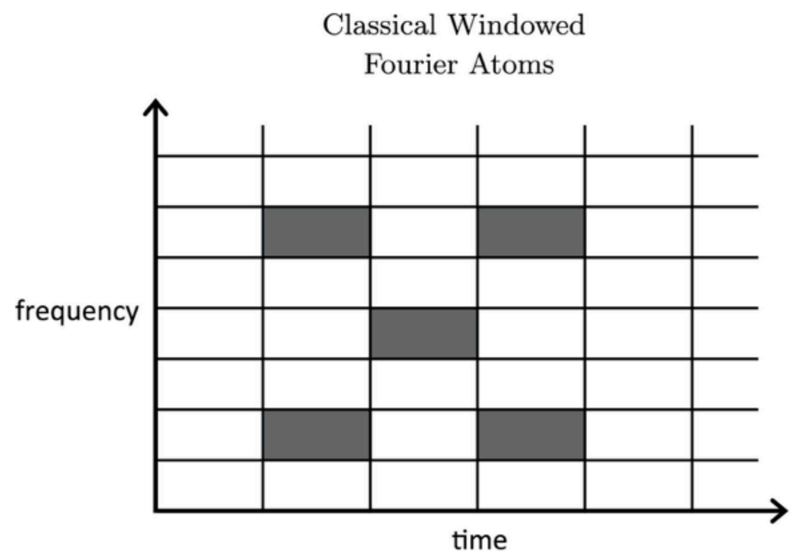
Shuman et al., "Vertex-frequency analysis on graphs," ACHA, 2016.

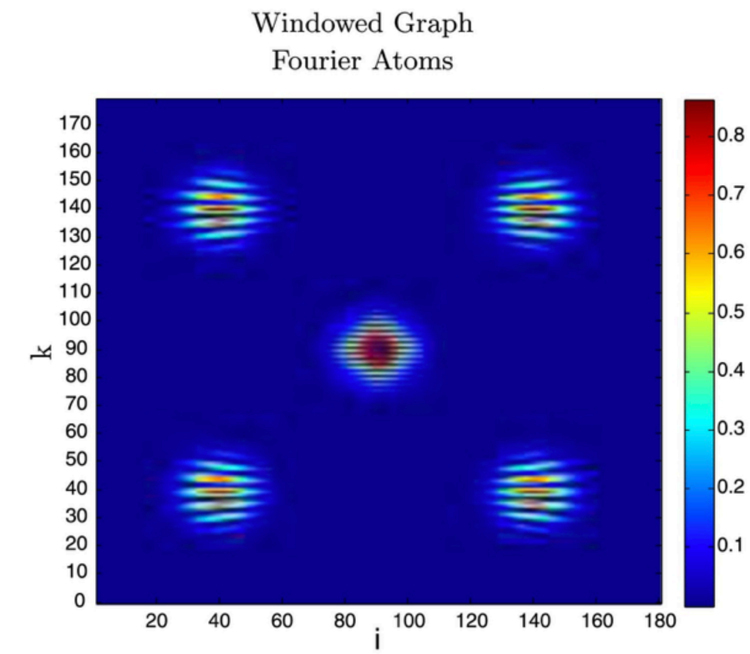
WGFT vs SGWT atoms

WGFT atom

$$g_{i,k}(n) := (M_k T_i g)(n)$$

$$= N \chi_k(n) \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$

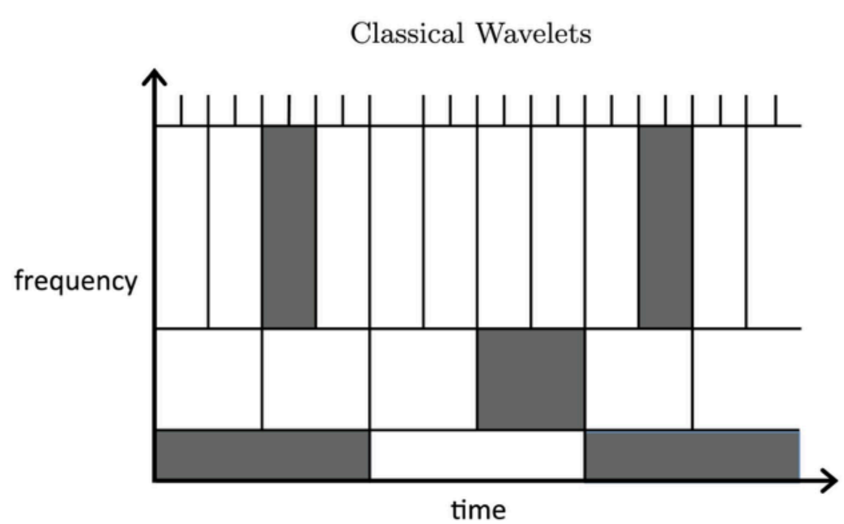


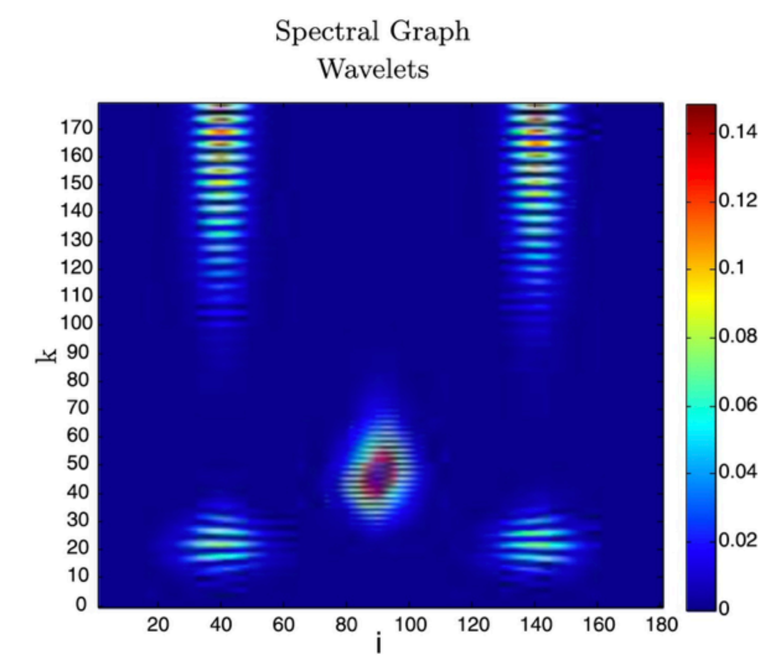


SGWT atom

$$\psi_{i,s}(n) := (T_i D_s g)(n)$$

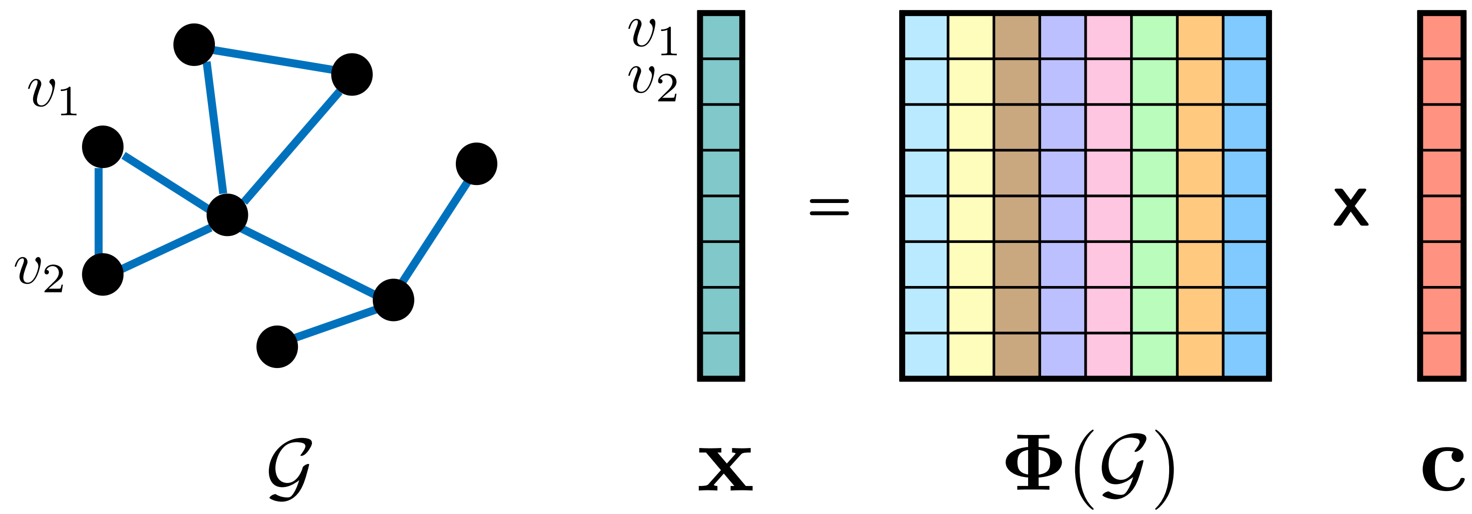
$$= \sum_{\ell=0}^{N-1} \hat{g}(s\lambda_\ell)\chi_\ell^*(i)\chi_\ell(n)$$





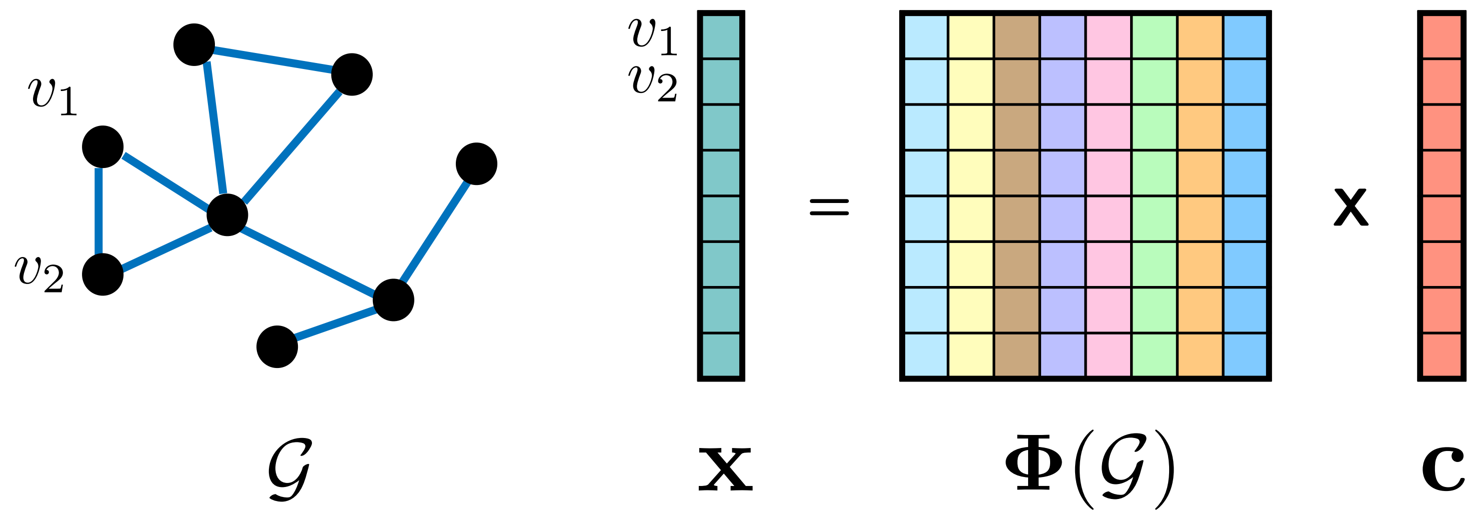
Shuman et al., "Vertex-frequency analysis on graphs," ACHA, 2016.

From analytical to trained graph dictionaries



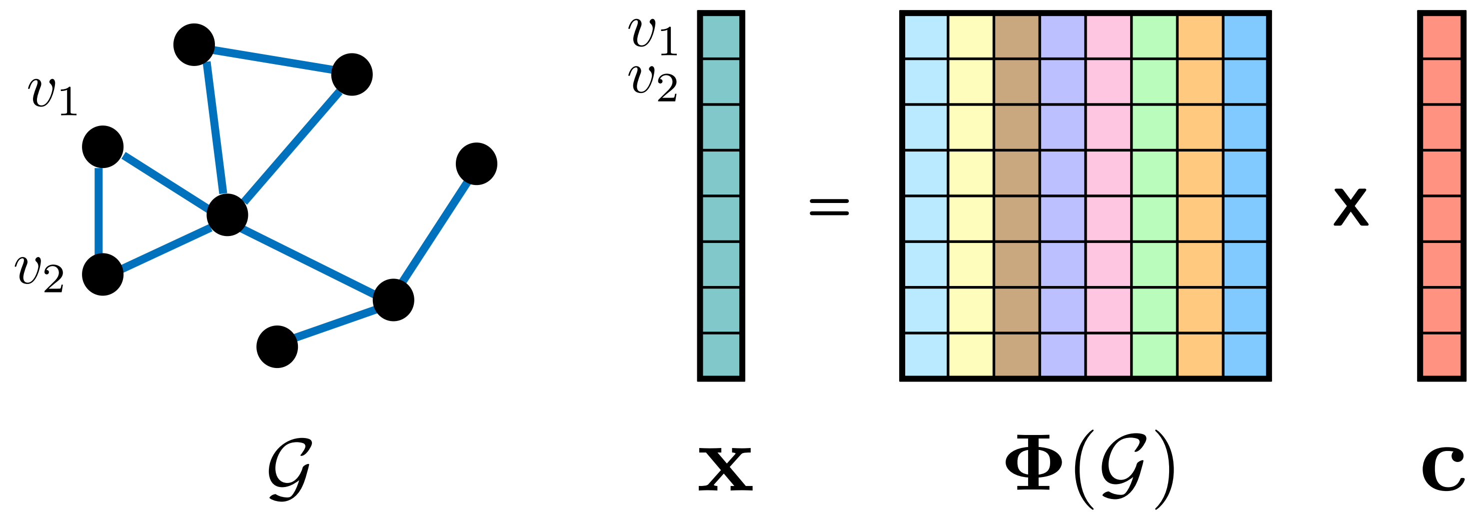
analytical graph dictionaries	$\mathbf{\Phi}(\mathcal{G})$	examples: GFT, WGFT, SGWT

From analytical to trained graph dictionaries



analytical graph dictionaries	$\mathbf{\Phi}(\mathcal{G})$	examples: GFT, WGFT, SGWT
trained graph dictionaries	$\mathbf{\Phi}(\mathcal{G}, \mathbf{x})$	dictionary learning on graphs?

From analytical to trained graph dictionaries

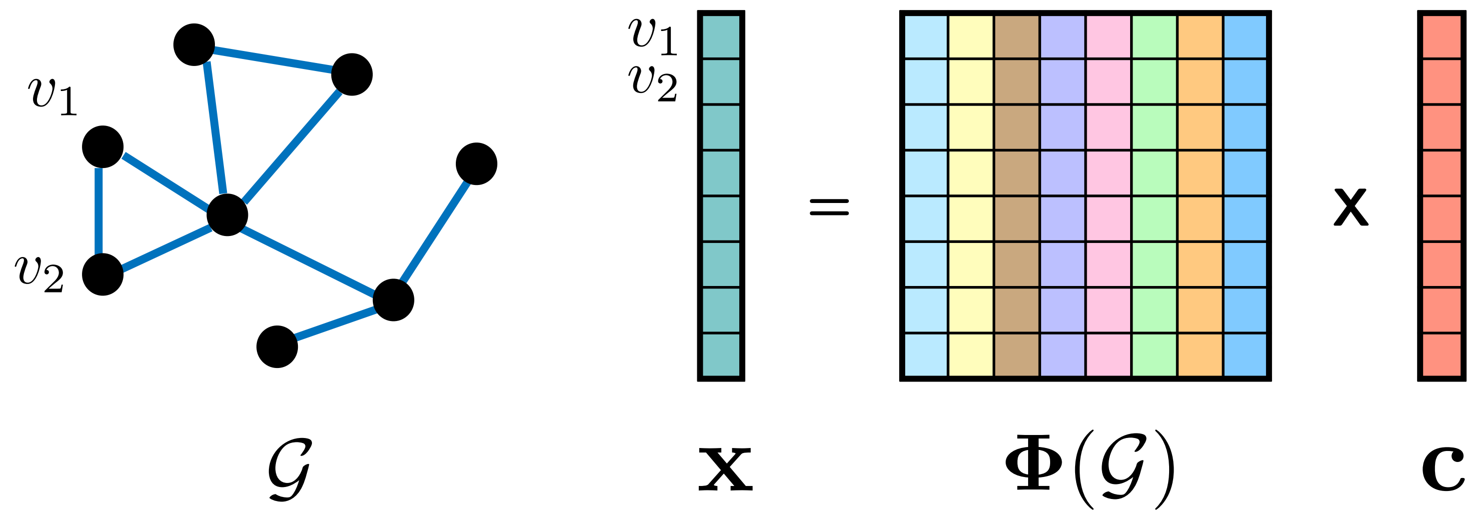


$$\boxed{f} \xrightarrow{\text{GFT}} \boxed{\chi^T f} \xrightarrow{\hat{g}(\Lambda)} \boxed{\hat{g}(\Lambda)\chi^T f} \xrightarrow{\text{IGFT}} \boxed{\chi\hat{g}(\Lambda)\chi^T f}$$

analytical graph dictionaries $\quad \mathbf{\Phi}(\mathcal{G}) = \hat{g}(L) = \chi\hat{g}(\Lambda)\chi^T$

trained graph dictionaries $\quad \mathbf{\Phi}(\mathcal{G}, \mathbf{x})$ dictionary learning on graphs?

From analytical to trained graph dictionaries



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

analytical graph dictionaries

$$\mathbf{\Phi}(\mathcal{G}) = \hat{g}(L) = \chi\hat{g}(\Lambda)\chi^T$$

trained graph dictionaries

$$\mathbf{\Phi}(\mathcal{G}, \mathbf{x}) \Rightarrow$$ learning $\hat{g}(\lambda)$ by adapting to $\mathbf{x}$

Dictionary learning on graphs

objective:

regularisation

$$\min_{\{\theta_i\}_{i=1}^{s} \in \mathbb{R}^{K+1}, C \in \mathbb{R}^{NS \times M}} ||X - DC||_F^2 + \mu \sum_{i=1}^{s} ||\theta_i||_F^2$$

adaptation to data

$$\text{subject to} \quad D = [\chi \hat{g}_{\theta_1}(L)\chi^T \;\; \chi \hat{g}_{\theta_2}(L)\chi^T \;\; \cdots \;\; \chi \hat{g}_{\theta_s}(L)\chi^T]$$

structured graph dictionaries

$$||c_m||_0 \leq T_0 \quad (C = [c_1 \;\; c_2 \;\; \cdots c_M])$$

sparsity constraint

Thanou et al., "Learning parametric dictionaries for signals on graphs," IEEE TSP, 2014.

Dictionary learning on graphs

objective:

regularisation

$$\min_{\{\theta_i\}_{i=1}^{s} \in \mathbb{R}^{K+1}, C \in \mathbb{R}^{NS \times M}} ||X - DC||_F^2 + \mu \sum_{i=1}^{s} ||\theta_i||_F^2$$

adaptation to data

$$\text{subject to} \qquad D = [\chi \hat{g}_{\theta_1}(L) \chi^T \; \chi \hat{g}_{\theta_2}(L) \chi^T \; \cdots \; \chi \hat{g}_{\theta_s}(L) \chi^T]$$

structured graph dictionaries

$$||c_m||_0 \leq T_0 \quad (C = [c_1 \; c_2 \; \cdots c_M])$$

sparsity constraint

solved by iterating between two steps:

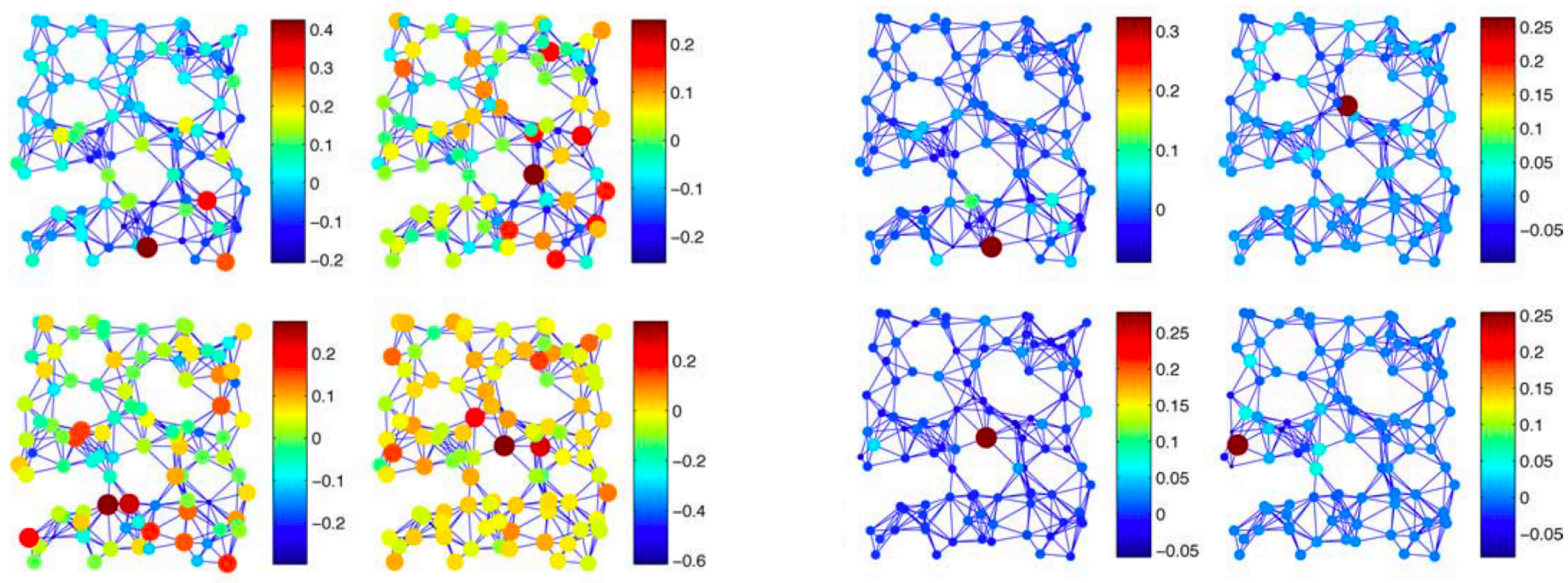
- **sparse approximation:** fixing D and solve for C
- **dictionary update:** fixing C and solve for D

Thanou et al., "Learning parametric dictionaries for signals on graphs," IEEE TSP, 2014.

Comparison with non-graph dictionaries

non-graph dictionary atoms (KSVD)

parametric dictionary atoms

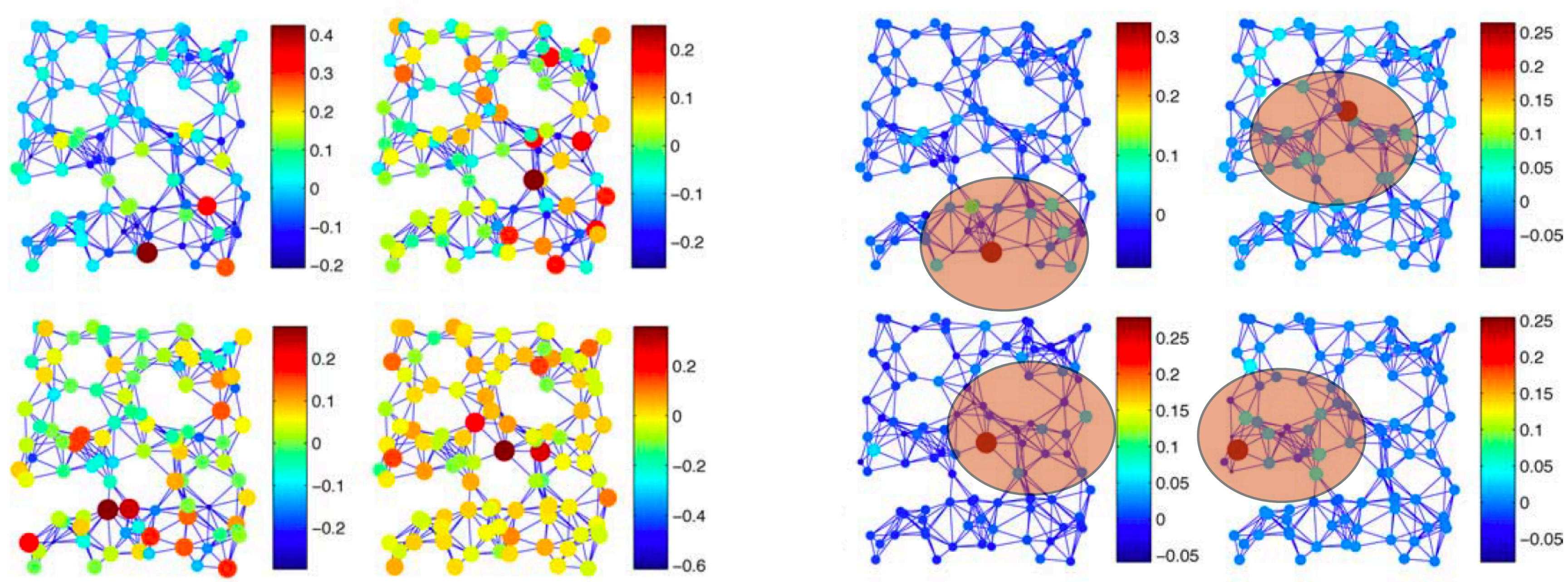


- non-graph dictionary atoms adapt to data but ignore the structure (hence are not localised)

Comparison with non-graph dictionaries

non-graph dictionary atoms (KSVD)

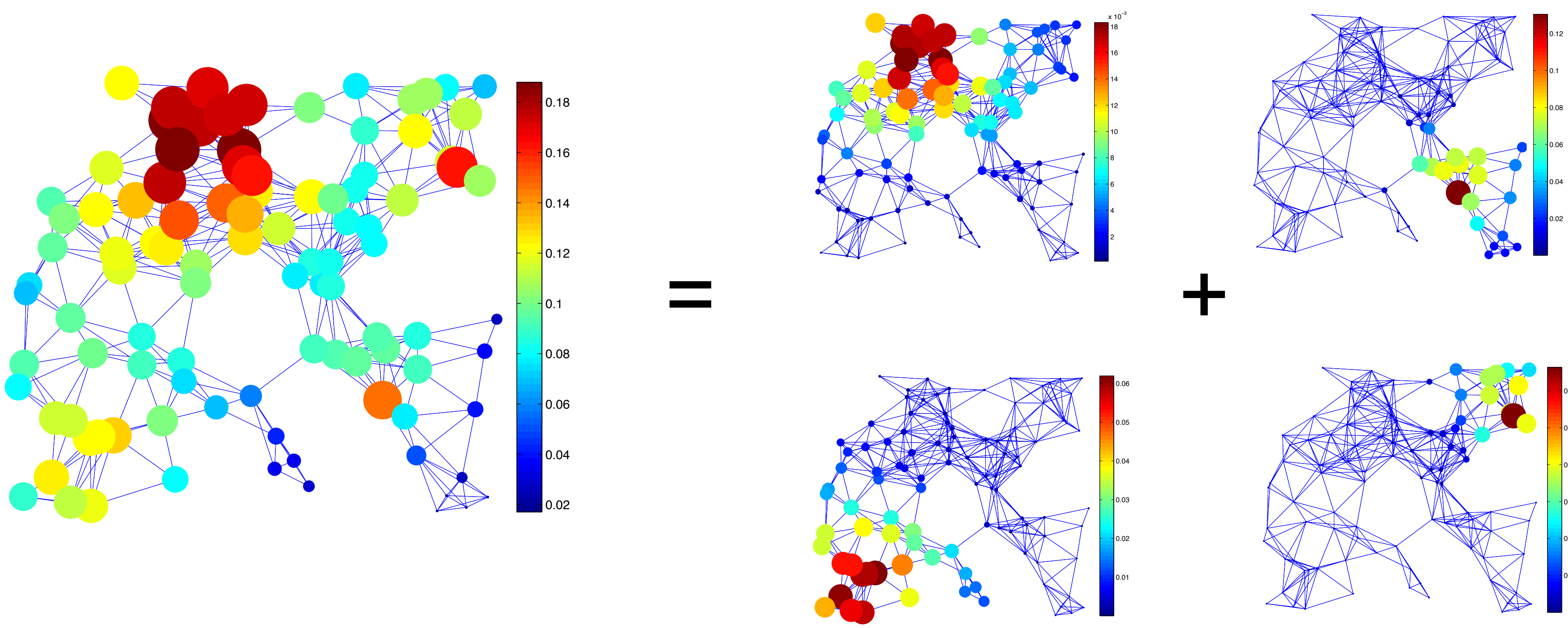
parametric dictionary atoms



- non-graph dictionary atoms adapt to data but ignore the structure (hence are not localised)
- graph dictionary atoms adapt to data and can also be designed to be localised

Thanou et al., "Learning parametric dictionaries for signals on graphs," IEEE TSP, 2014.

Decomposition using parametric dictionary



- the dictionary atoms adapt to localised patterns in different regions of the graph

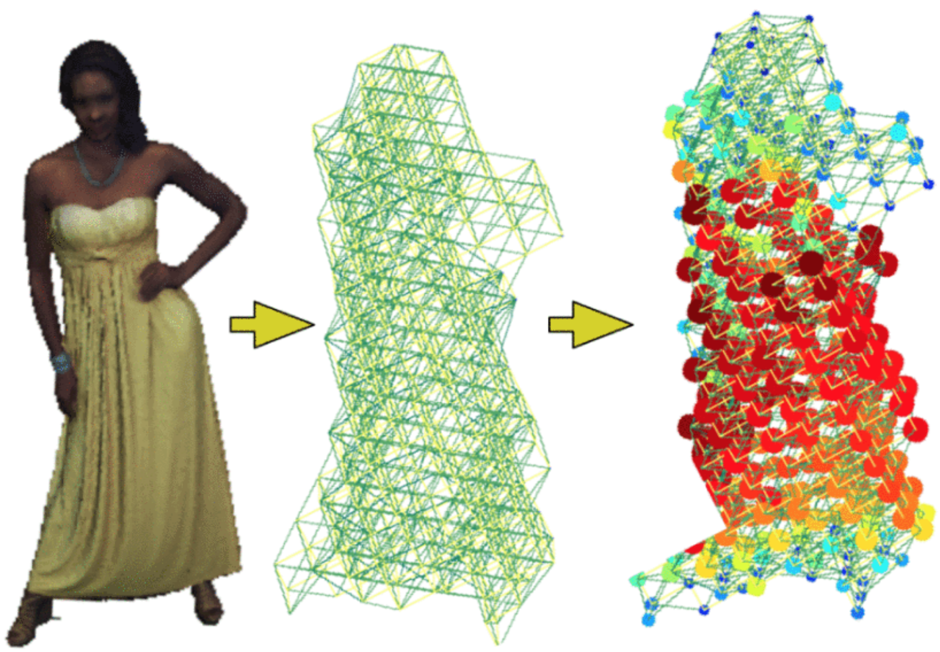
Graph dictionary design - Summary

- Analytical vs trained graph dictionaries
  - mathematical modelling of data on graphs
  - adaptation to data on graphs
- Both approaches focus on design or learning of the kernel function
  - shift, modulation, scaling, learning-based
- This lecture has focused on Laplacian spectrum based designs
  - other possibilities exist (e.g., purely vertex-domain designs)
- Connection with other fields
  - representation learning on graphs (e.g., node embedding)
  - deep learning on graphs

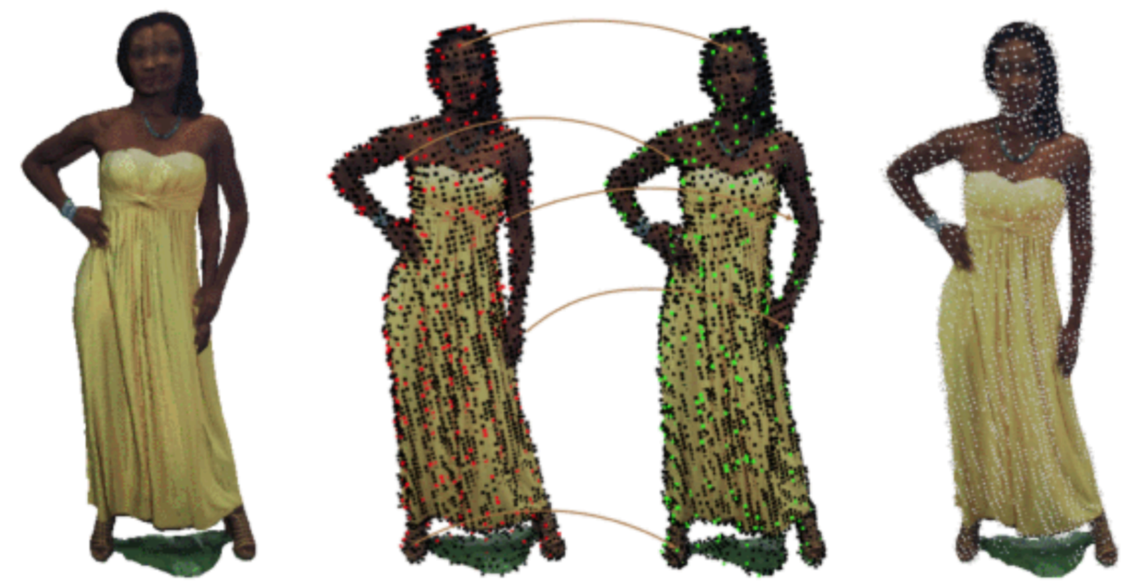
Lecture 2

- Graph signal processing: Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Representation of graph signals
- Applications

Application I: 3D point cloud analysis



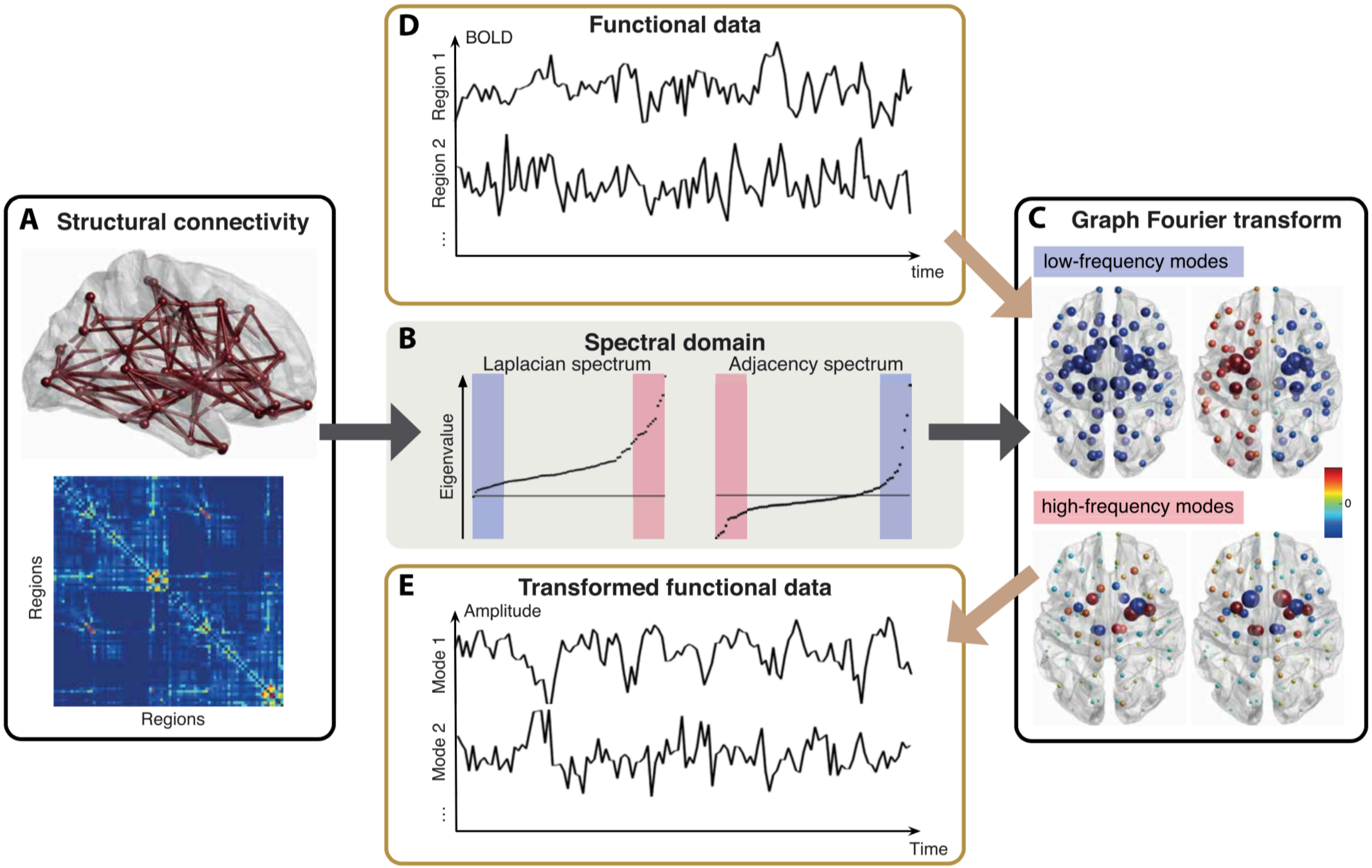
(a) (b) (c)



(a) $\mathcal{I}_t + \mathcal{I}_{t+1}$ (b) Correspondence between $\mathcal{I}_t$ and $\mathcal{I}_{t+1}$ (c) $\mathcal{I}_{t,mc} + \mathcal{I}_{t+1}$

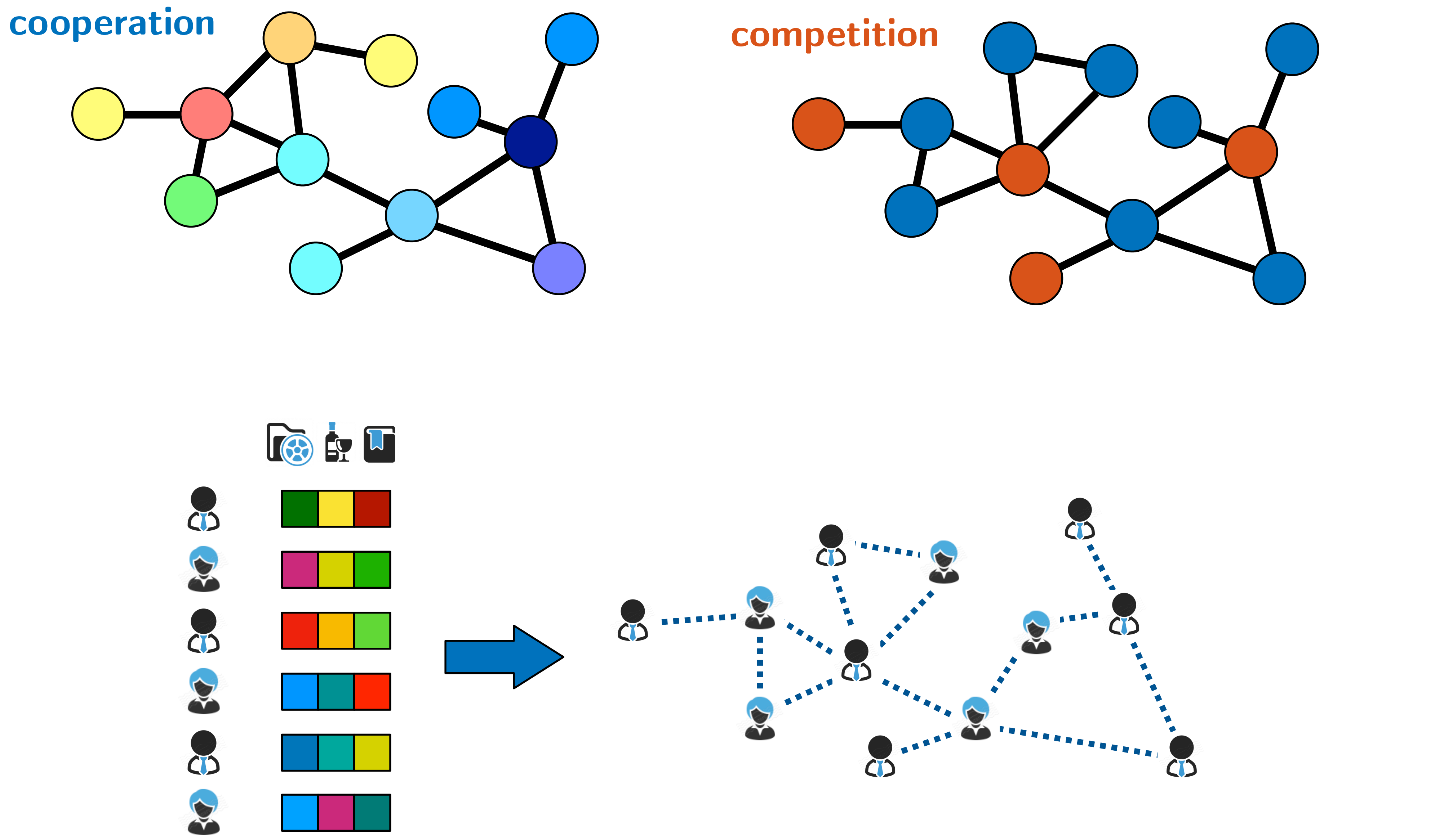
Thanou et al., "Graph-based compression of dynamic 3D point cloud sequences," IEEE TIP, 2016.

Application II: Neuroscience



Huang et al., "A graph signal processing perspective on functional brain imaging," Proceedings of the IEEE, 2018.

Application III: Inferring strategic relations



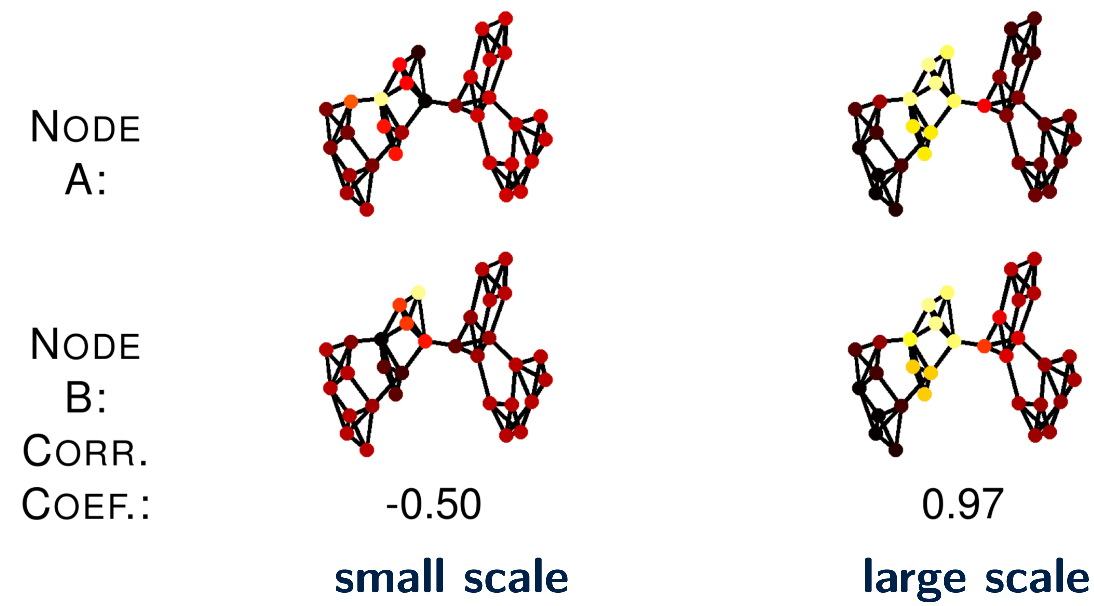
infer strategic relations from equilibrium actions

Leng et al., "Learning quadratic games on networks," ICML, 2020.

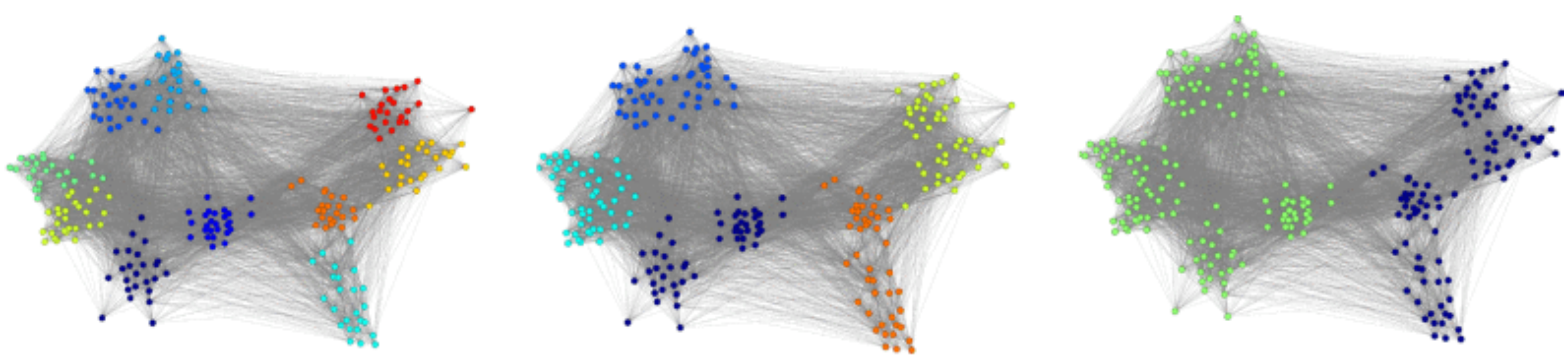
Application IV: Community detection

spectral graph wavelets
at different scales:

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



multi-scale community
detection:

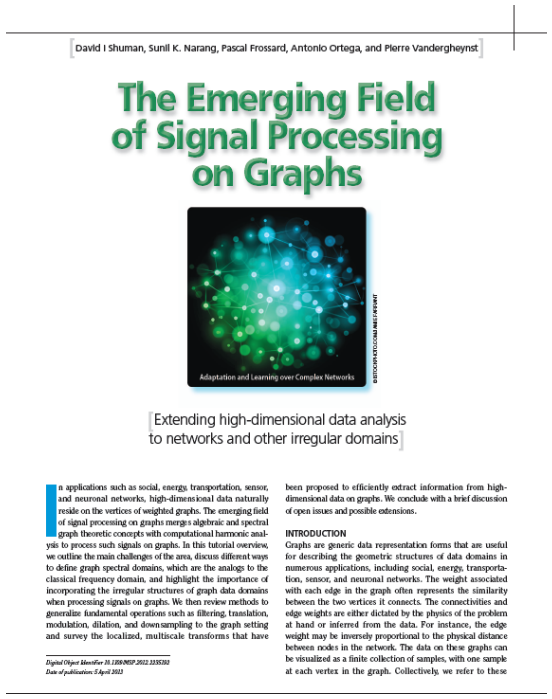


Tremblay et al., "Graph wavelets for multiscale community mining," IEEE TSP, 2014.

Promising directions in GSP

- Mathematical models for graph signals
  - incorporating underlying physical processes
  - probabilistic modelling on graphs
  - how to handle temporal dynamics?
- Graph construction
  - how to infer graph topology given observed data?
- Implementation issues
  - fast graph Fourier transform
  - distributed processing
- Connection with other fields
  - complex networks and systems
  - deep learning on graphs
  - Bayesian modelling

References


David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst

The Emerging Field of Signal Processing on Graphs

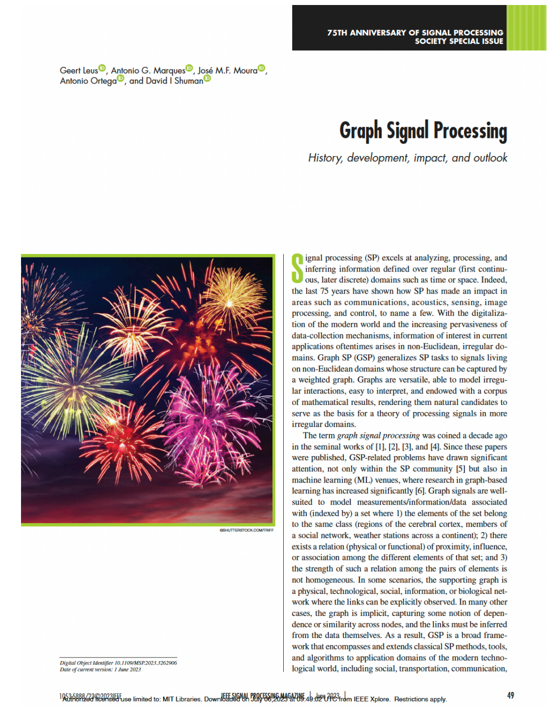
Extending high-dimensional data analysis to networks and other irregular domains


INVITED PAPER

Graph Signal Processing: Overview, Challenges, and Applications

This article presents methods to process data associated to graphs (graph signals) extending techniques (transforms, sampling, and others) that are used for conventional signals.

By Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst


75TH ANNIVERSARY OF SIGNAL PROCESSING SOCIETY SPECIAL ISSUE

Geert Leus, Antonio G. Marques, José M.F. Moura, Antonio Ortega, and David I Shuman

Graph Signal Processing

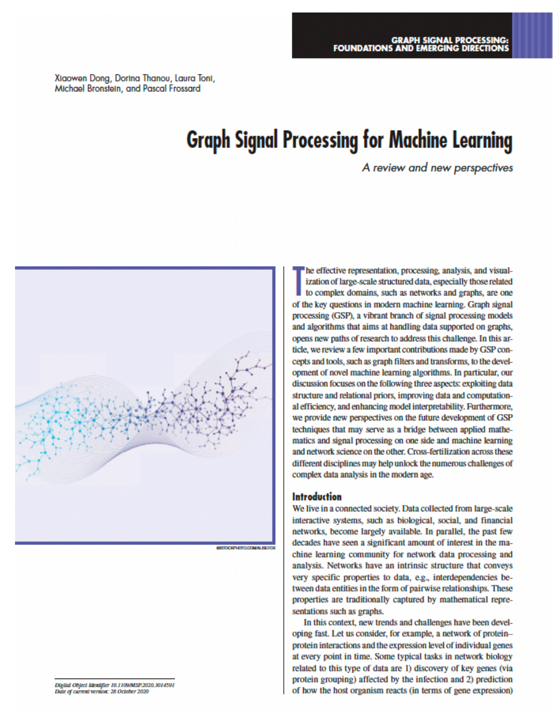
History, development, impact, and outlook


Appl. Comput. Harmon. Anal. 40 (2016) 260–291

Applied and Computational Harmonic Analysis

Vertex-frequency analysis on graphs

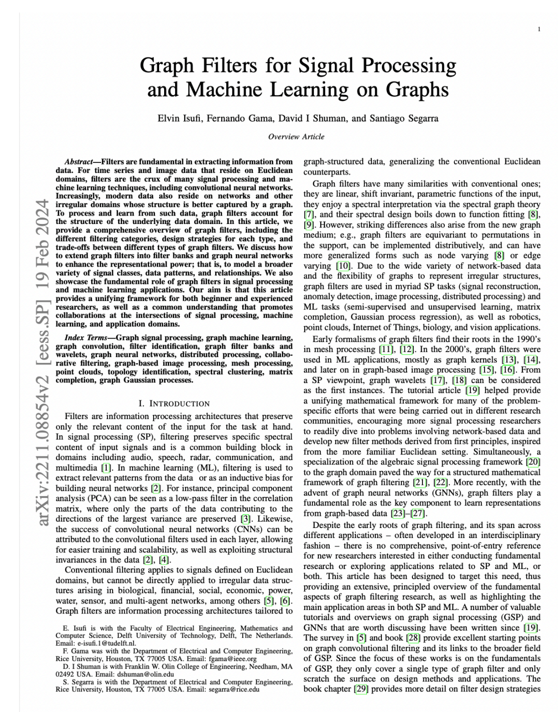
David I Shuman, Benjamin Ricaud, Pierre Vandergheynst


GRAPH SIGNAL PROCESSING: FOUNDATIONS AND EMERGING DIRECTIONS

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

Graph Signal Processing for Machine Learning

A review and new perspectives


Graph Filters for Signal Processing and Machine Learning on Graphs

Elvin Isufi, Fernando Gama, David I Shuman, and Santiago Segarra

Overview Article

arXiv:2211.08854v2 [eess.SP] 19 Feb 2024