AIMS CDT - Signal Processing

Michaelmas Term 2022

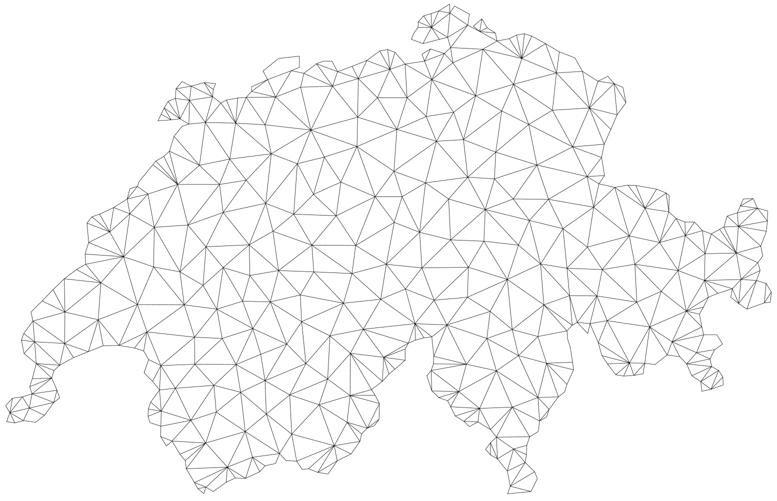
Xiaowen Dong

Department of Engineering Science

UNIVERSITY OF OXFORD

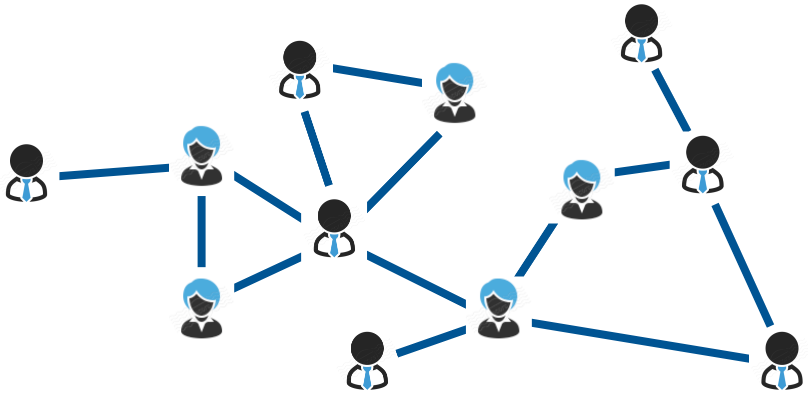
Introduction to Graphs Signal Processing

Graphs are everywhere

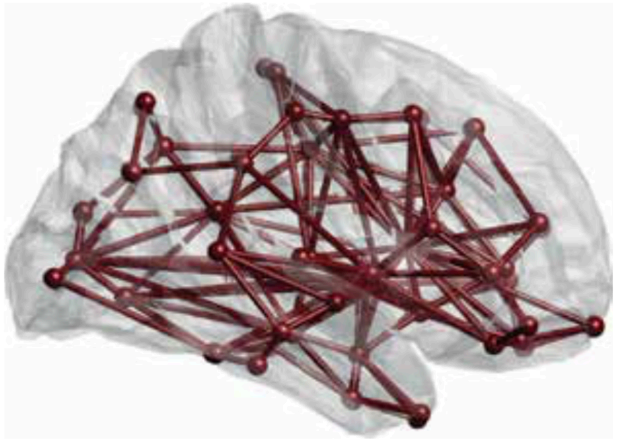


geographical network

traffic network

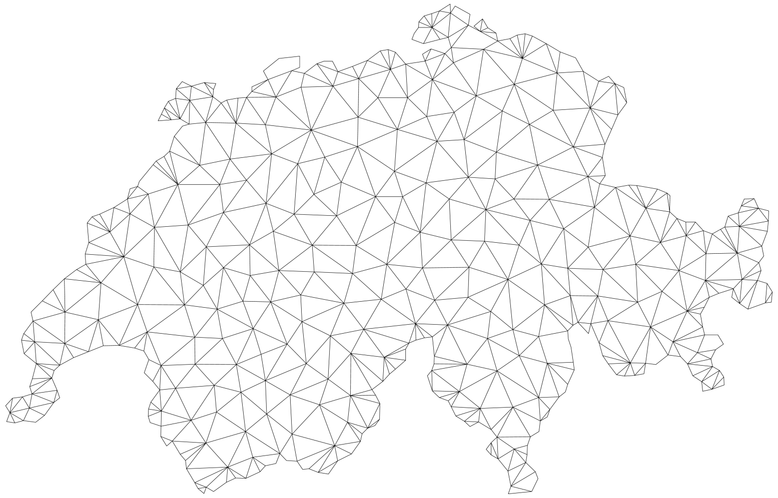


social network



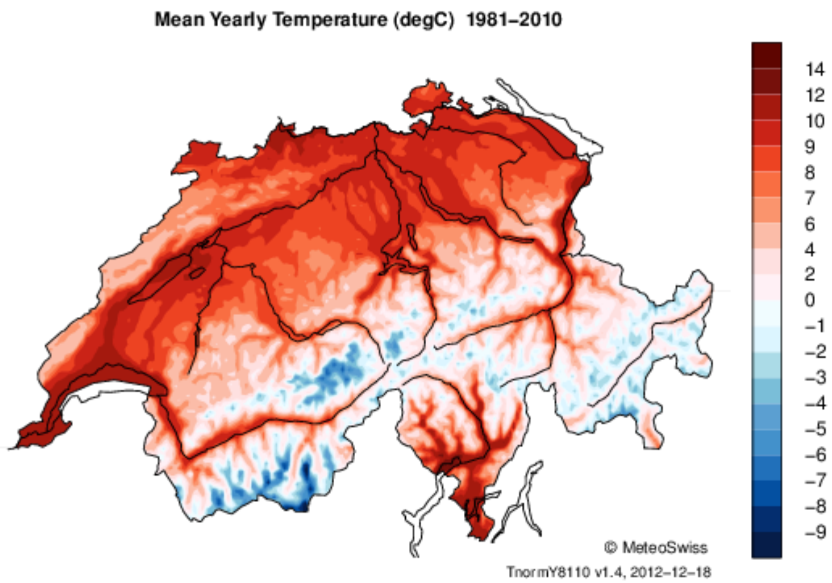
brain network

Graph-structured data are everywhere



- vertices
  - geographical regions
- edges
  - geographical proximity between regions

Graph-structured data are everywhere

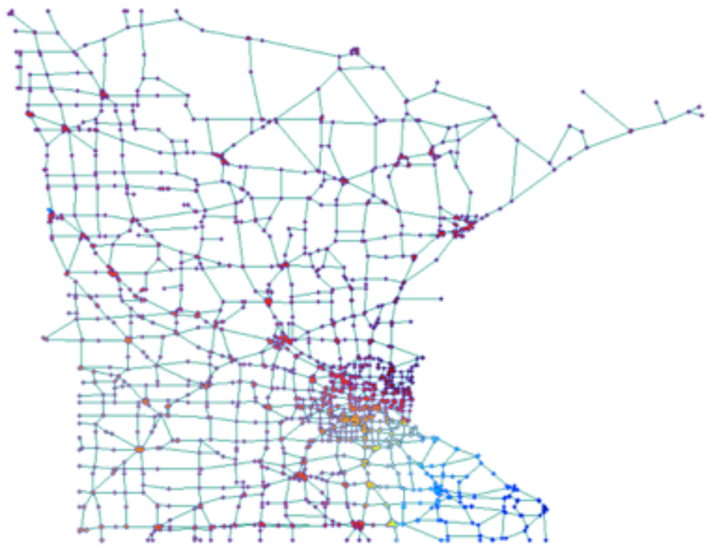


- vertices
  - geographical regions
- edges
  - geographical proximity between regions
- signal
  - temperature records in these regions

Graph-structured data are everywhere

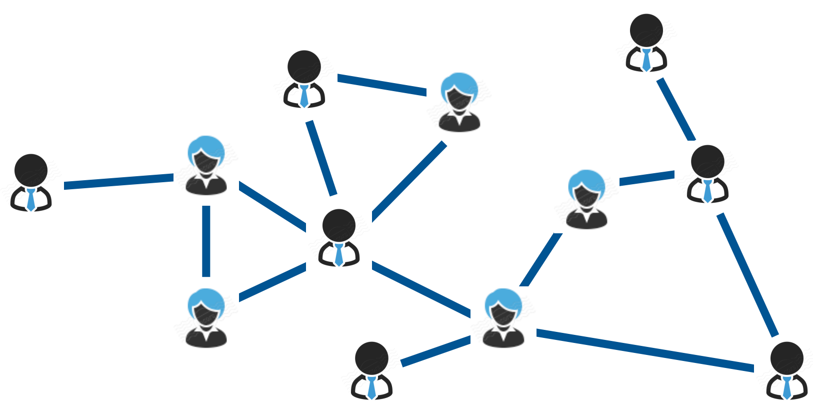
- vertices
  - road junctions
- edges
  - road connections

Graph-structured data are everywhere



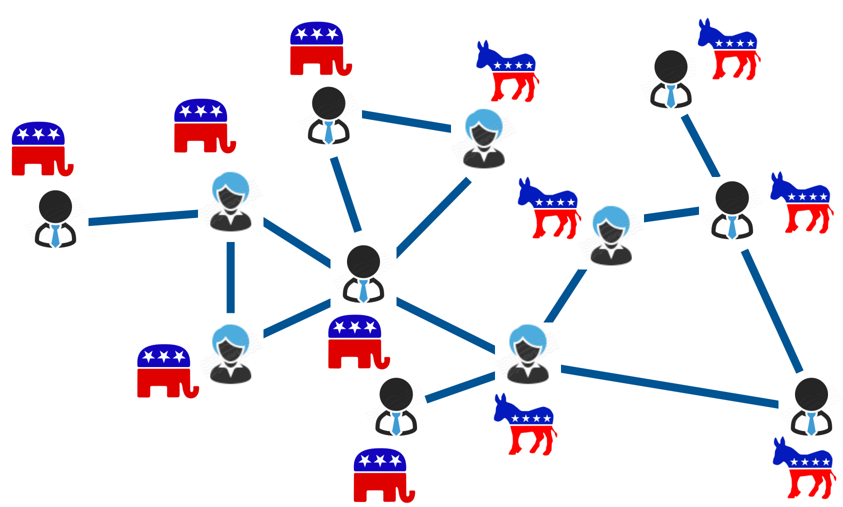
- vertices
  - road junctions
- edges
  - road connections
- signal
  - traffic congestion at junctions

Graph-structured data are everywhere



- vertices
  - individuals
- edges
  - friendship between individuals

Graph-structured data are everywhere

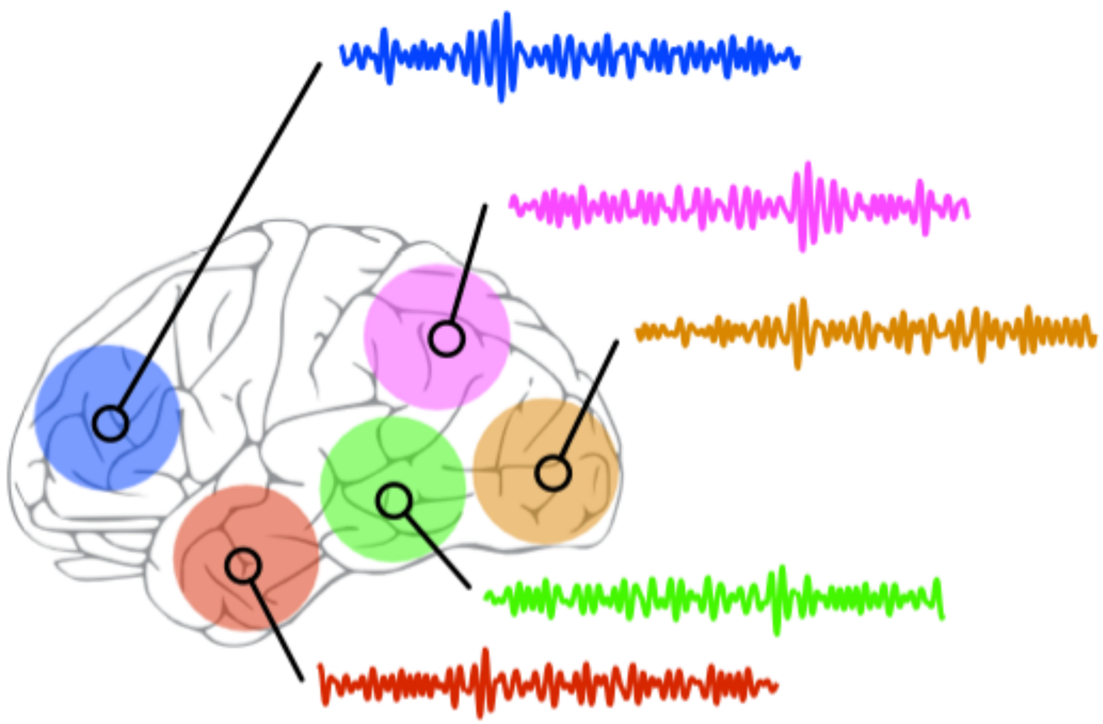


- vertices
  - individuals
- edges
  - friendship between individuals
- signal
  - political view

Graph-structured data are everywhere

- vertices
  - brain regions
- edges
  - structural connectivity between brain regions

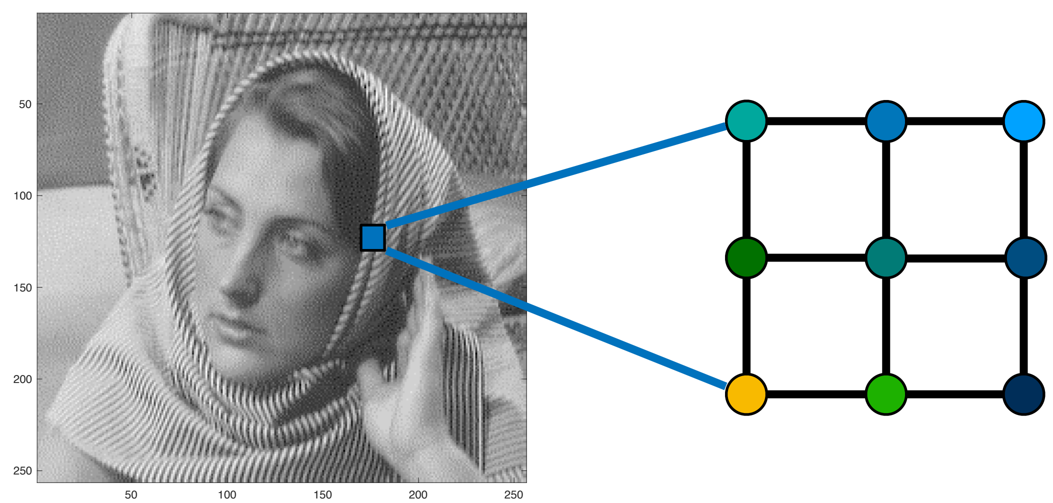
Graph-structured data are everywhere



- vertices
  - 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.

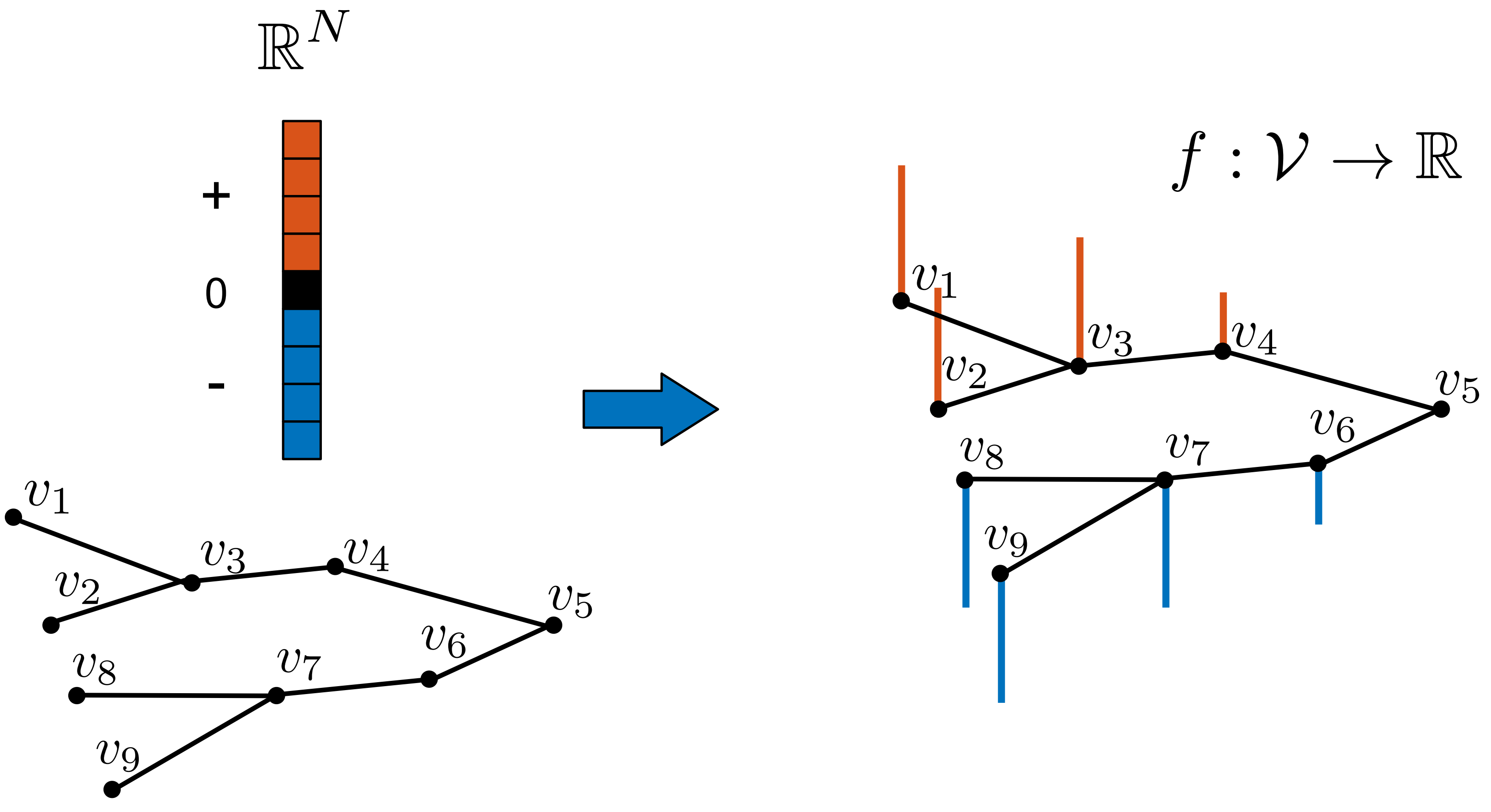
Graph-structured data are everywhere



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

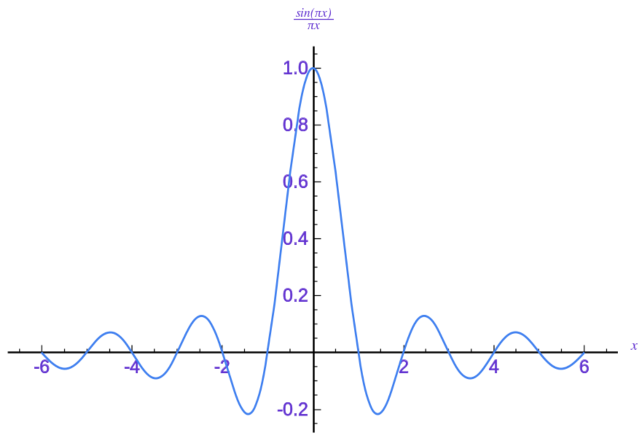
Graph signal processing

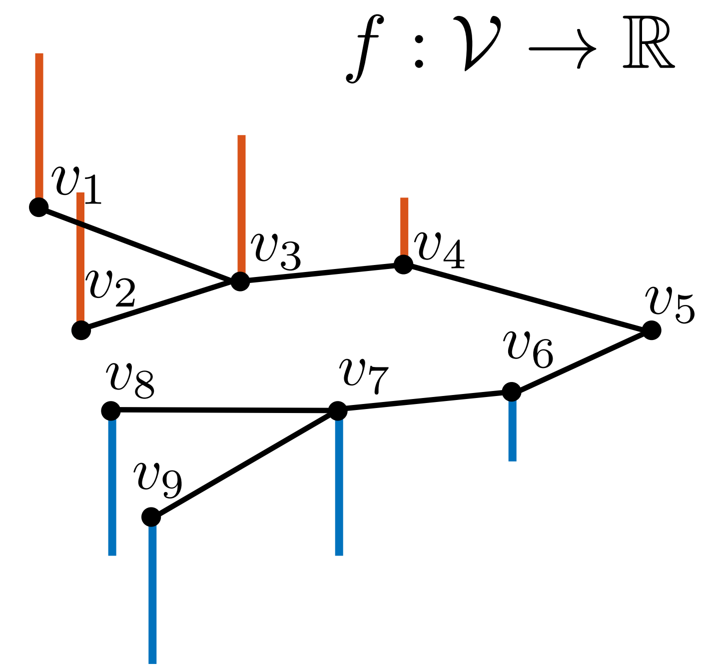
- Graph-structured data can be represented by signals defined on graphs or **graph signals**



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

Graph signal processing

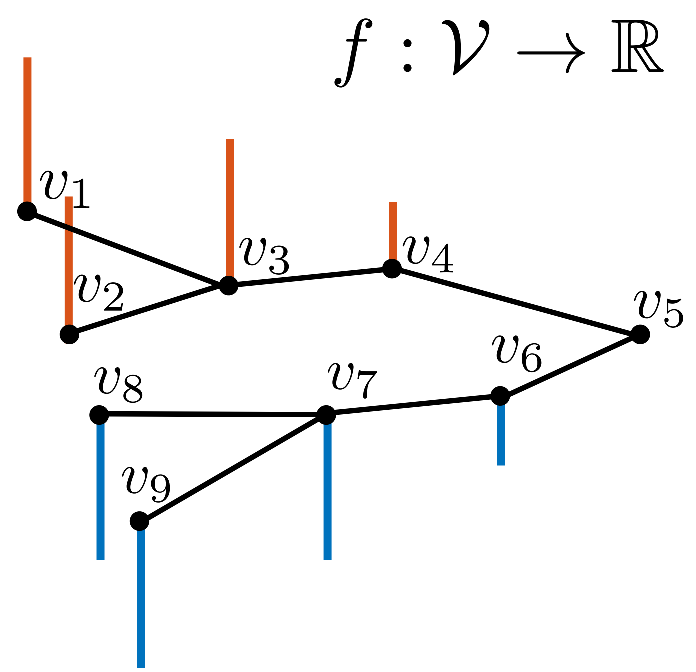




how to generalise classical signal processing tools on irregular domains such as graphs?

Graph signal processing

- Graph signals provide a nice compact format to encode structure within data
- Generalisation of classical signal processing tools can greatly benefit analysis of such data
- Numerous applications: Transportation, biomedical, social, economic network analysis
- An increasingly rich literature
  - classical signal processing
  - algebraic and spectral graph theory
  - computational harmonic analysis
  - machine learning



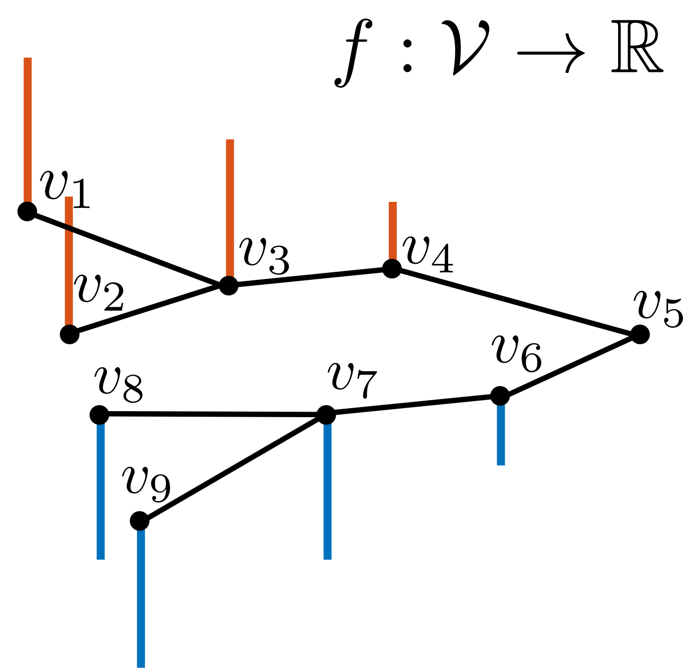
Outline

- Graph signal processing (GSP): Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Connection with literature
- Representation of graph signals
- Applications

Two paradigms

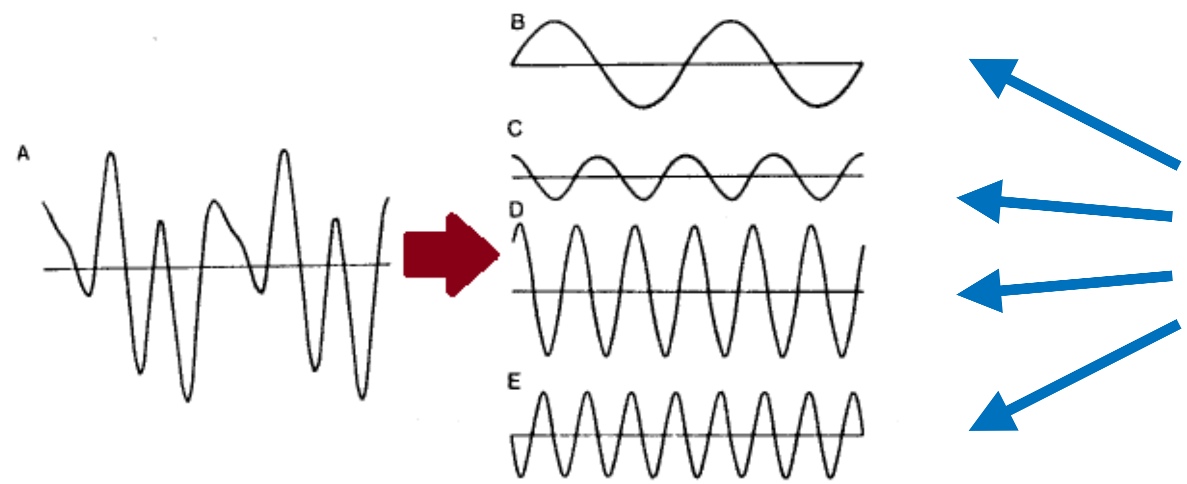
- Main GSP approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs

important for analysis of signal properties



Need for notion of frequency

- Classical Fourier transform provides frequency domain representation of signals

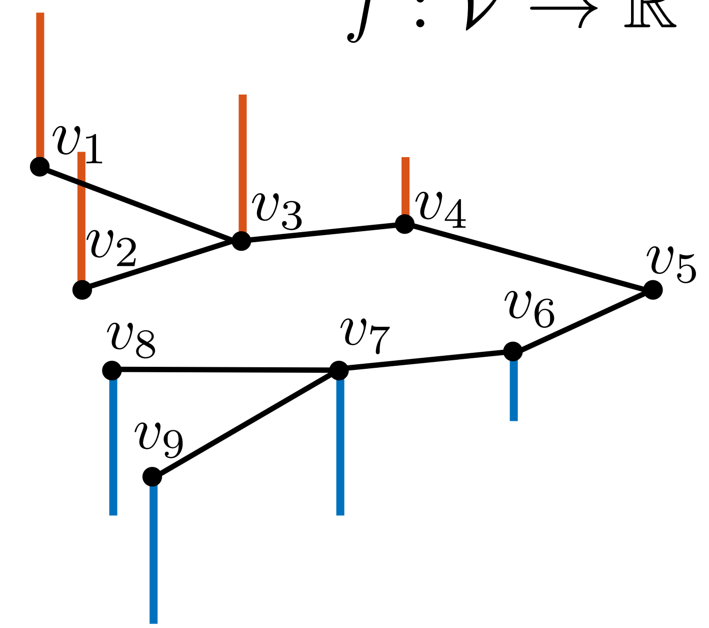


- "building blocks" of signal
- different frequency (oscillation)

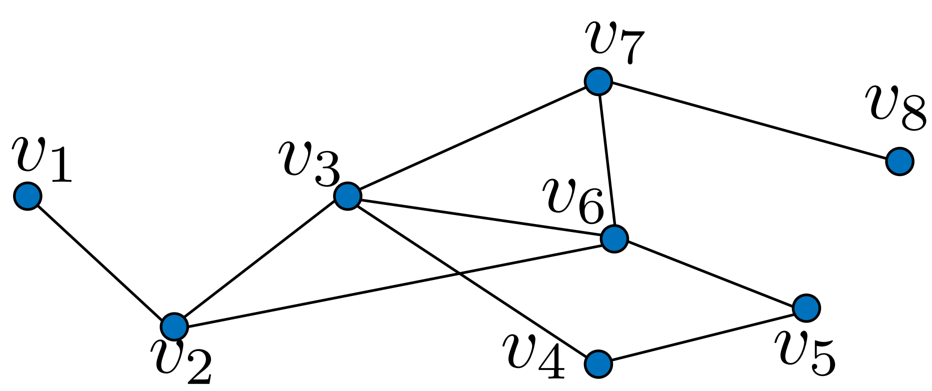
- What about a notion of frequency for graph signals?

  we need the graph Laplacian matrix

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



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!**

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

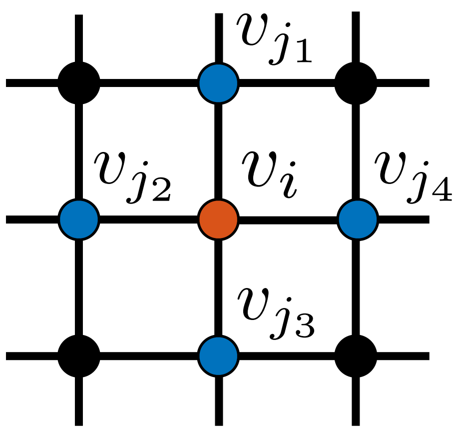
$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{D}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

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

Graph Laplacian

Why graph Laplacian?

- approximation of the Laplace operator

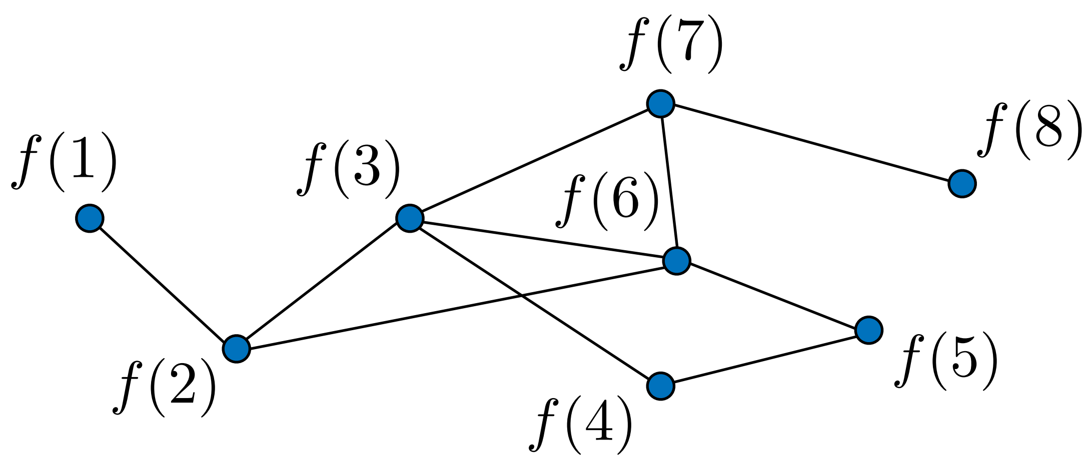


$$(Lf)(i) = \big(4f(i) - f(j_1) - f(j_2) - f(j_3) - f(j_4)\big)/(\delta x)^2$$

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

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

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

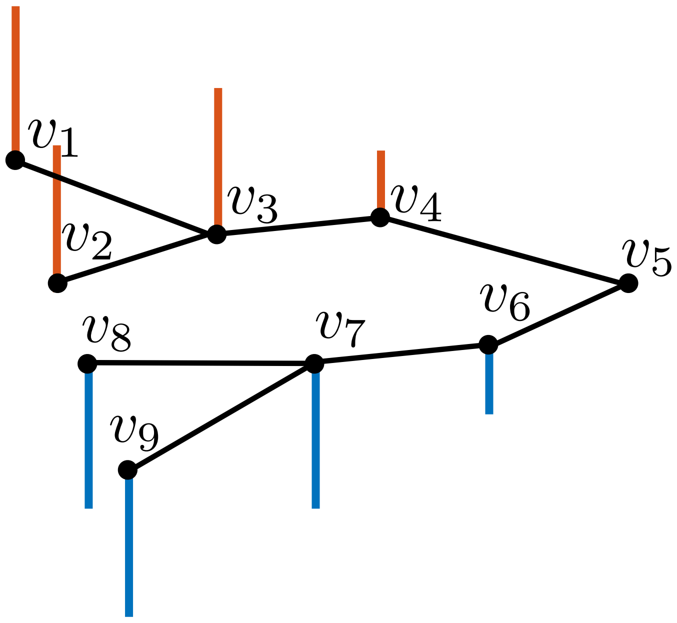
$$\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}^T \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 &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

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

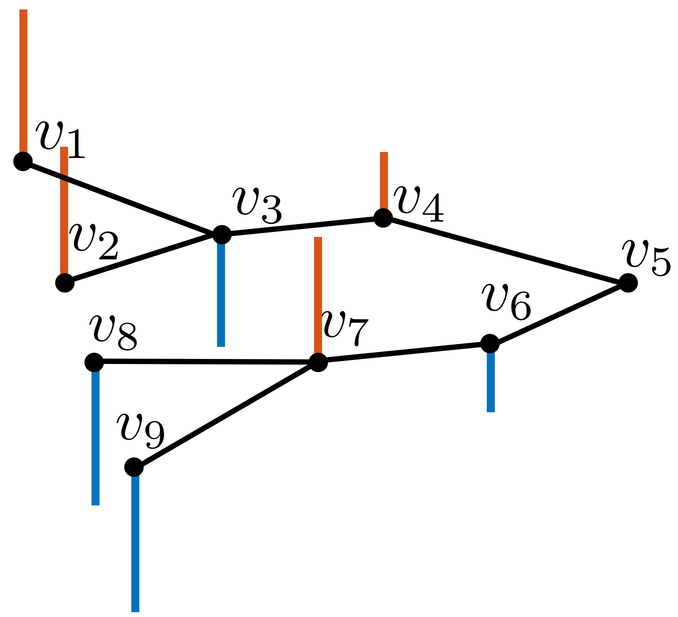
a measure of "smoothness"

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

Graph Laplacian



$f^T L f = 1$



$f^T L f = 21$

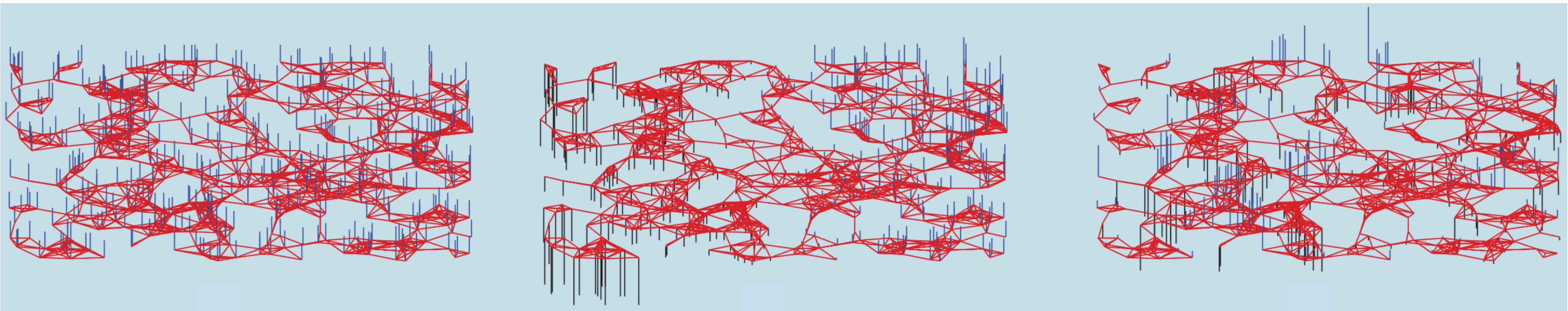
Graph Laplacian

- 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}

low frequency → high frequency

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

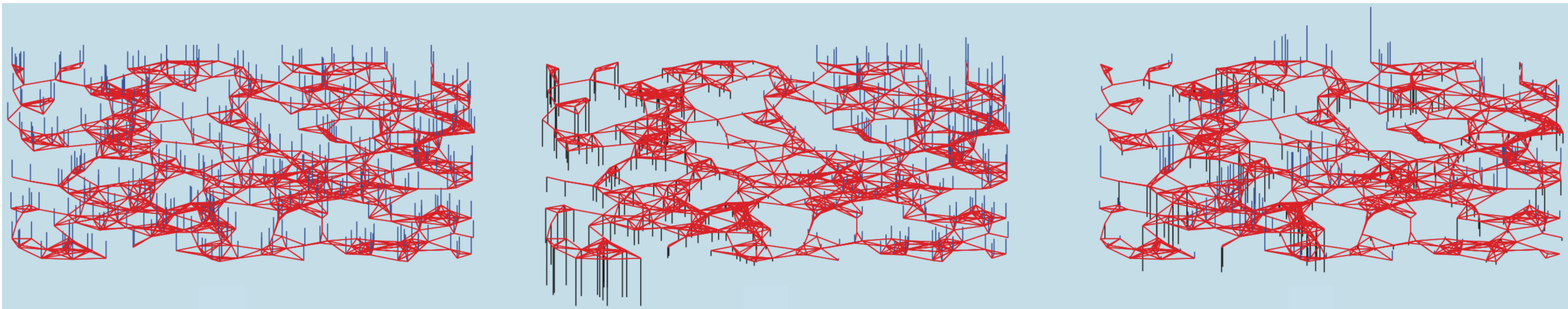
$\chi_0^T L \chi_0 = \lambda_0 = 0$

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

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

Graph Fourier transform



χ_0 χ_1 χ_{50}

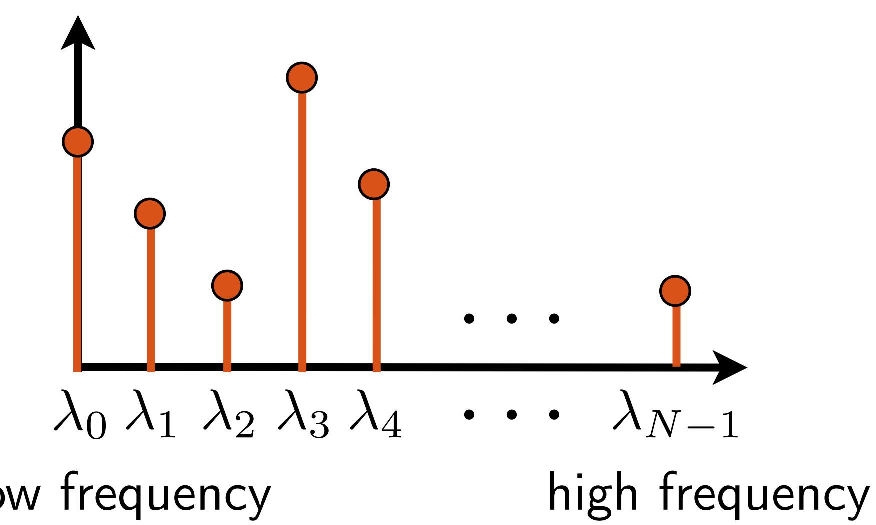
low frequency → high frequency

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

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

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

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

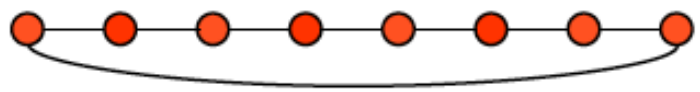
eigenvectors: χ_ℓ

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

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

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

Two special cases



- (Unordered) Laplacian eigenvalues: $\lambda_\ell = 2 - 2\cos\left(\frac{2\ell\pi}{N}\right)$

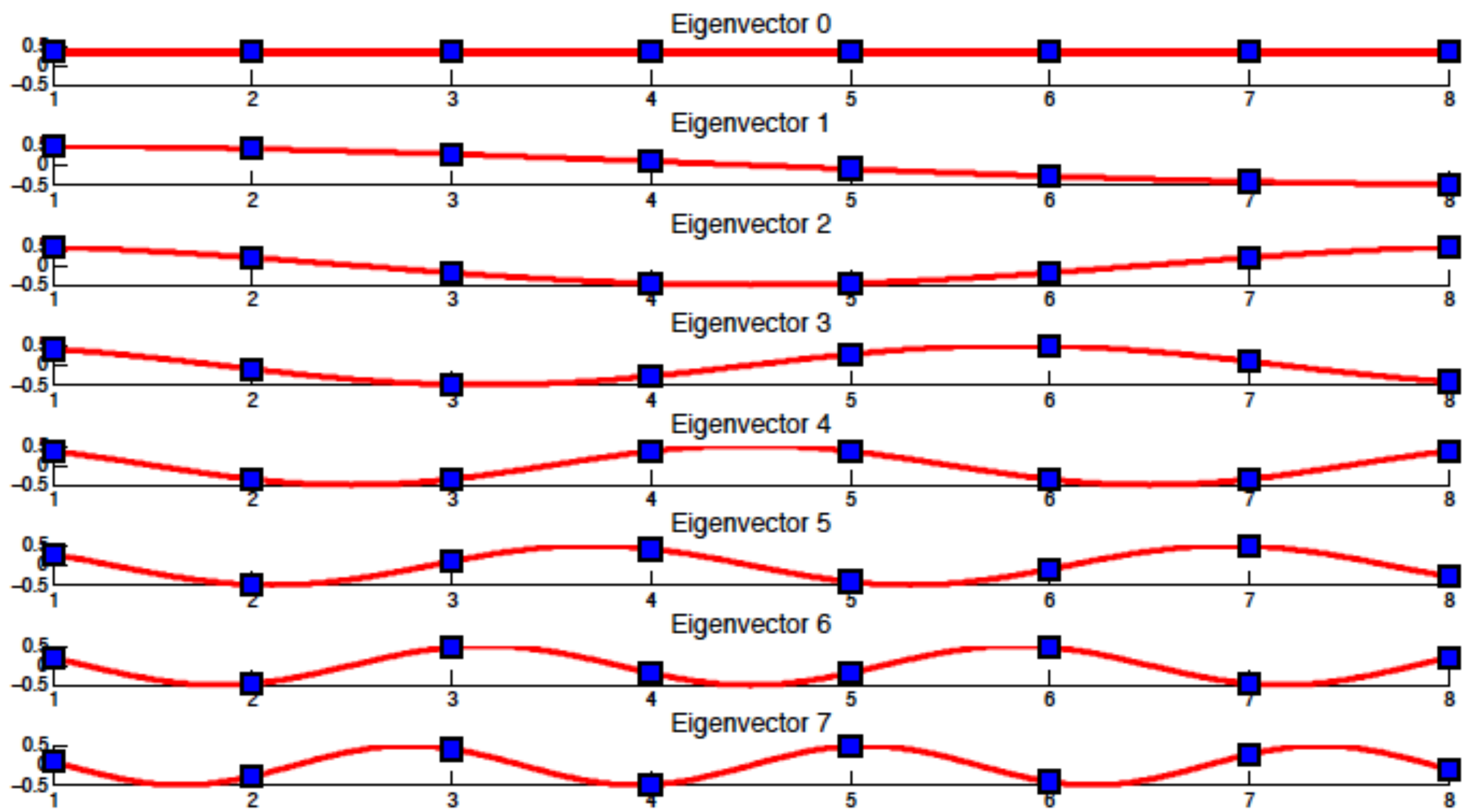
- One possible choice of orthogonal Laplacian eigenvectors:

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

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

Two special cases

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

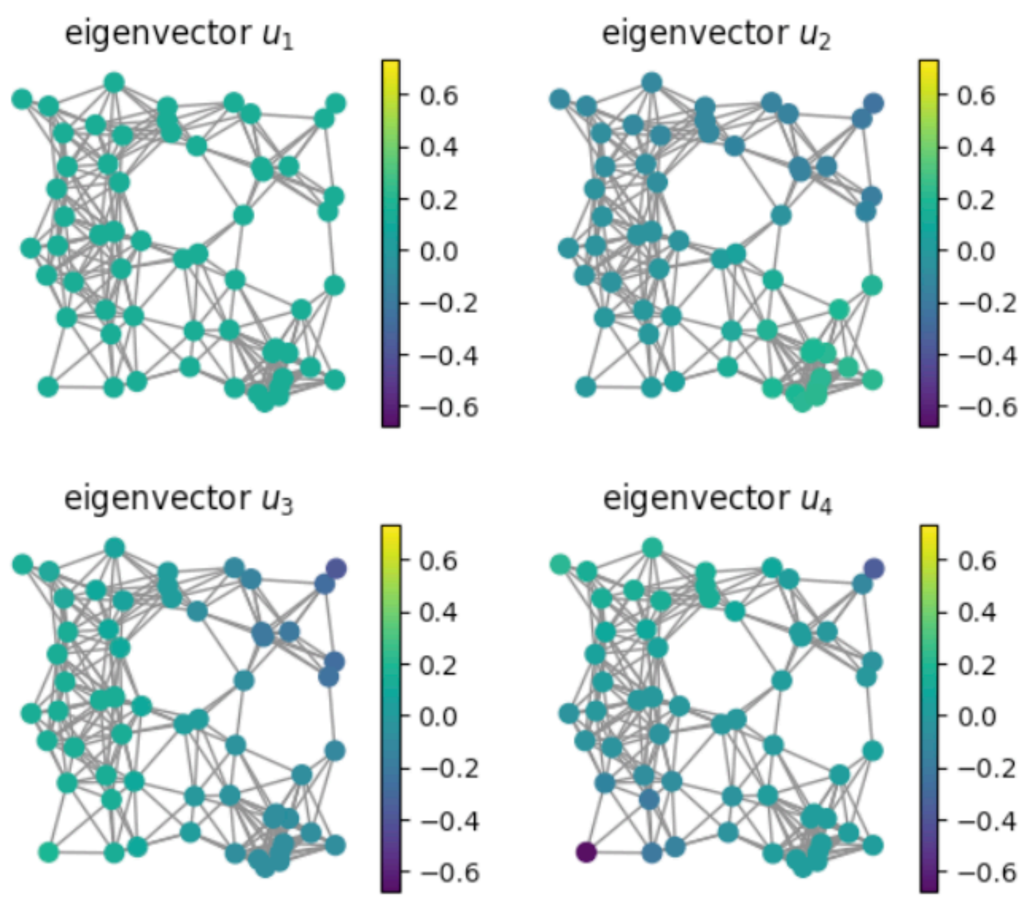


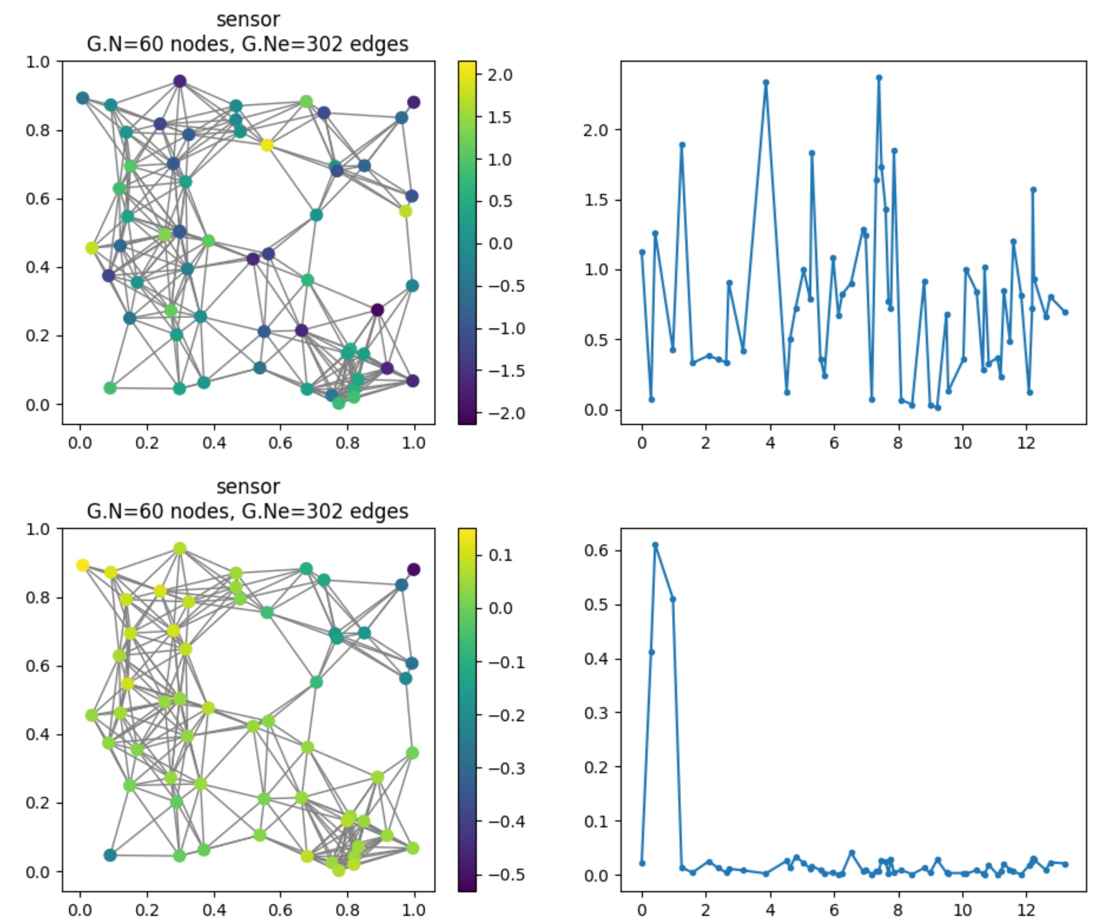
$\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N-1} \\ | & & | \end{bmatrix}$ is the Discrete Cosine Transform matrix (DCT-II, Strang, 1999), which is used in JPEG image compression

Vandergheynst and Shuman, "Wavelets on graphs, an introduction," Université de Provence, 2011.

Graph Fourier transform

- Example on a simple graph



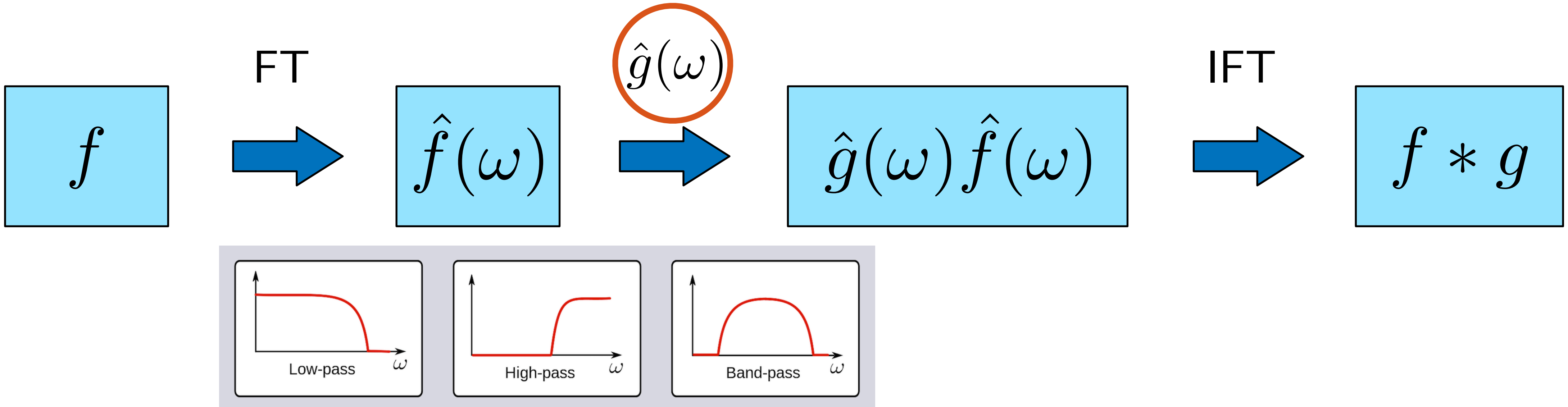


Outline

- Graph signal processing (GSP): Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Connection with literature
- Representation of graph signals
- Applications

Classical frequency filtering

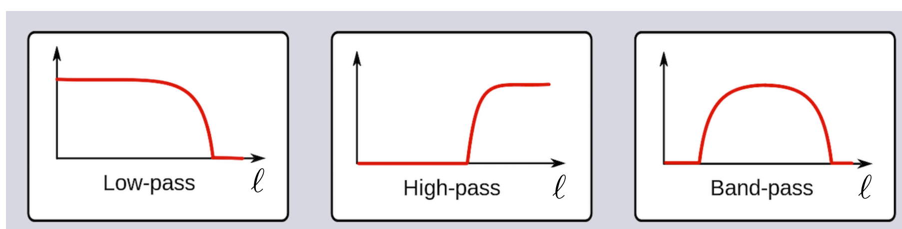
Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$

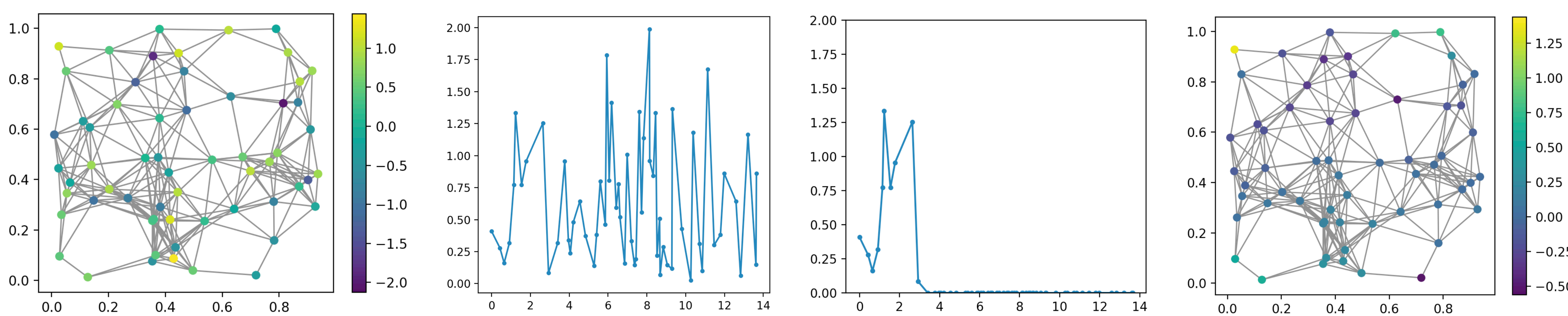


Graph spectral filtering

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

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

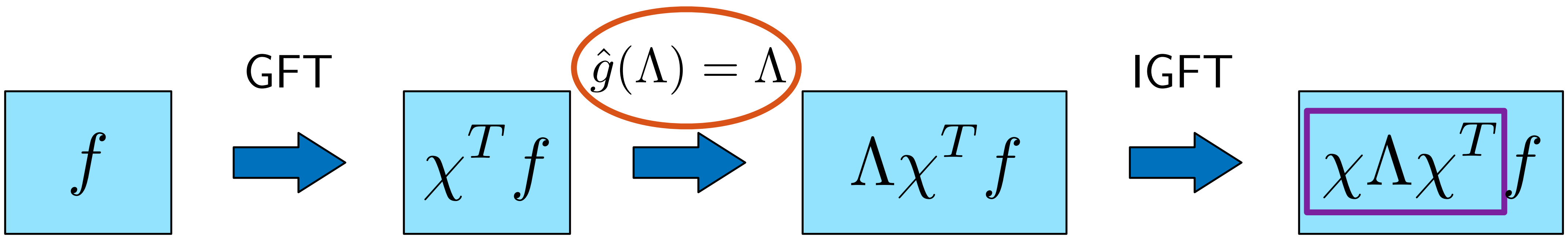




Graph Laplacian revisited

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: $Lf = \chi \Lambda \chi^T f$



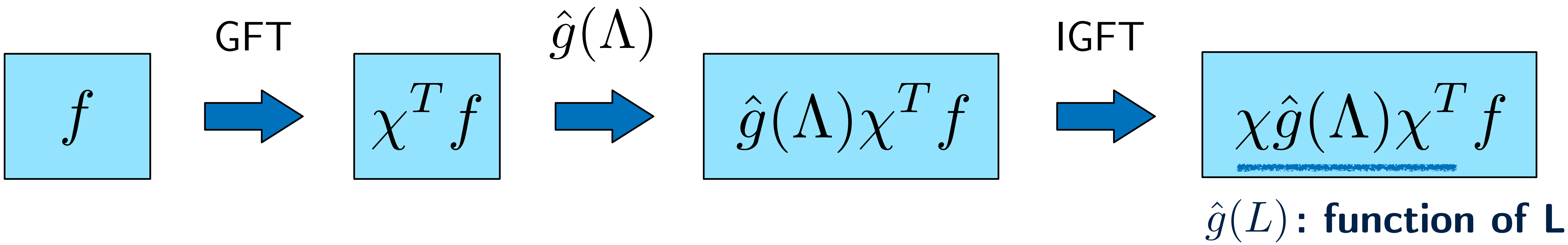
$$\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}$$

The Laplacian operator filters the signal in the spectral domain by its eigenvalues!

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

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

- Closely related to kernels and regularisation on graphs

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

A practical example

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: $||y - f||_2^2$

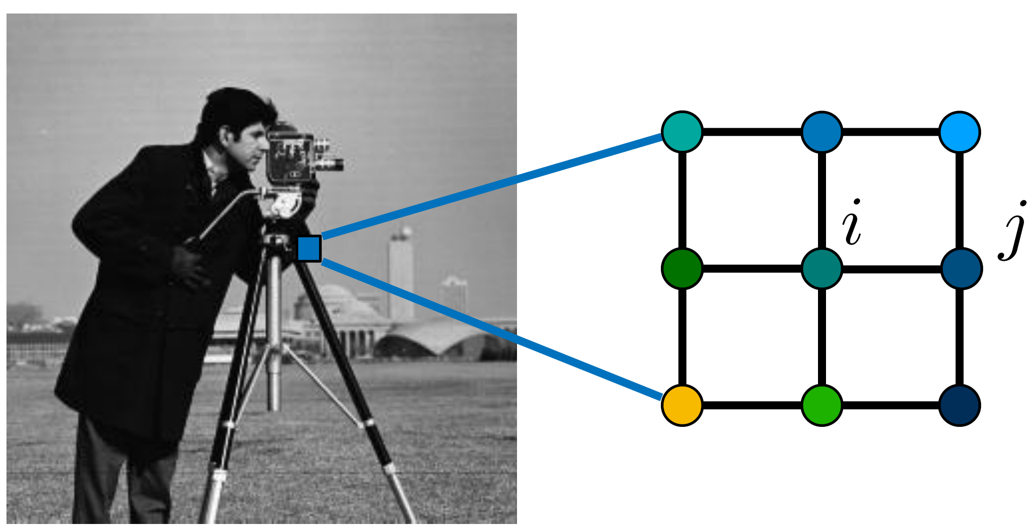
"smoothness" assumption: $\gamma y^T L y$

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

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

A practical example

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

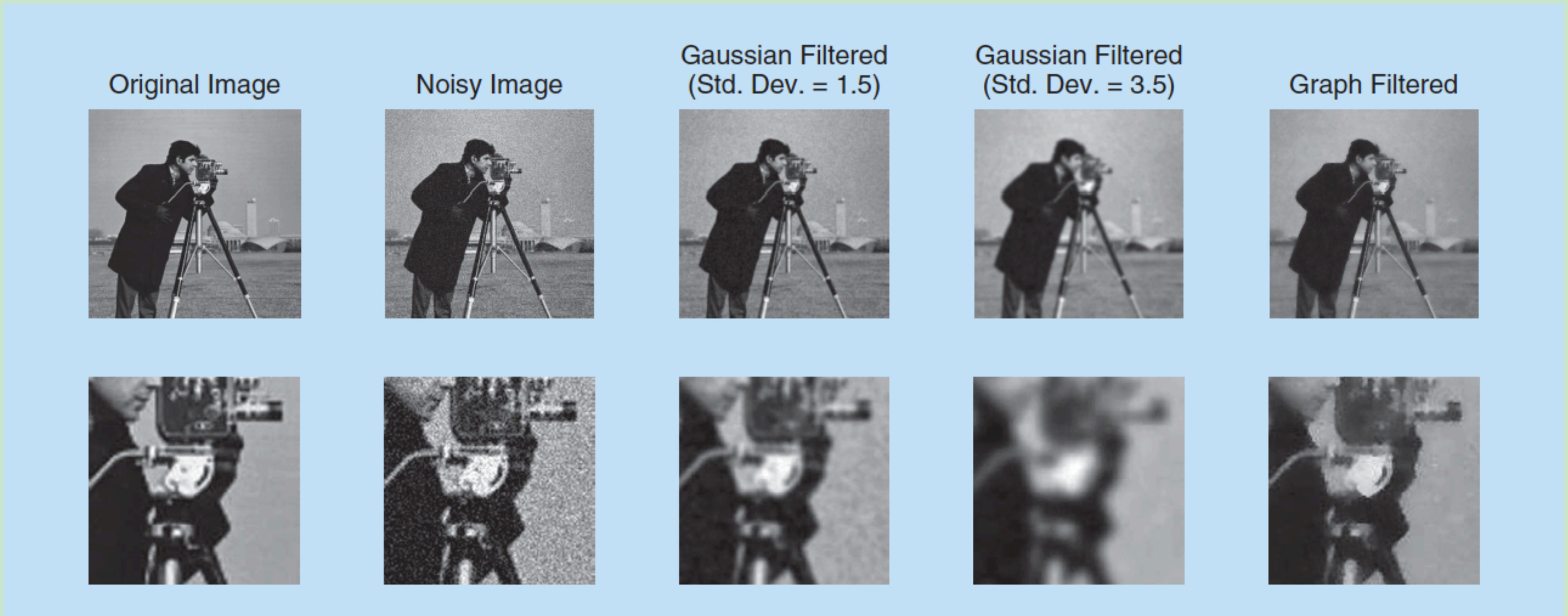


$$w_{ij} = \frac{1}{|f(i) - f(i)|}$$

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

A practical example

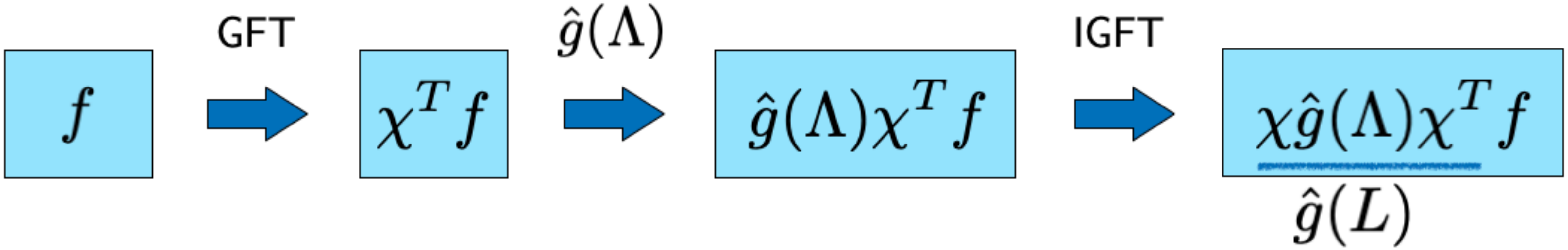
- noisy image as observed noisy graph signal
- regular grid graph (weights inversely proportional to pixel value difference)



[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 transform/dictionary design



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

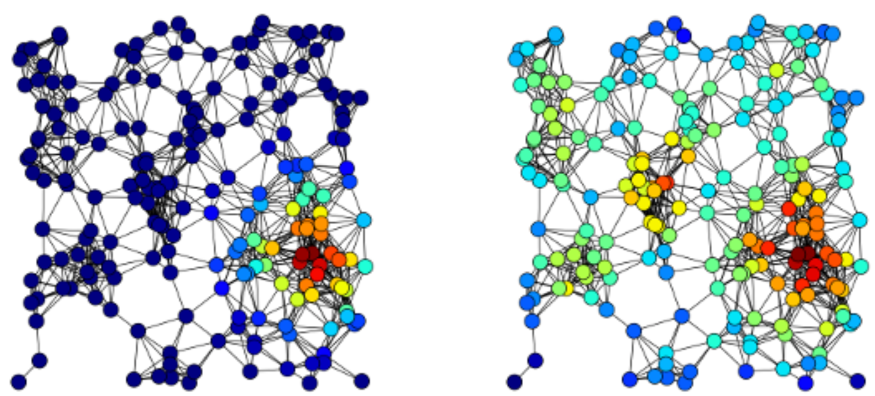
Shuman et al., "Vertex-frequency analysis on graphs," ACHA, 2016.

Outline

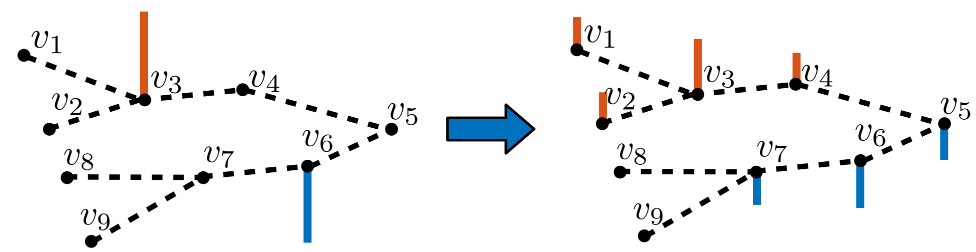
- Graph signal processing (GSP): Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Connection with literature
- Representation of graph signals
- Applications

GSP and the literature

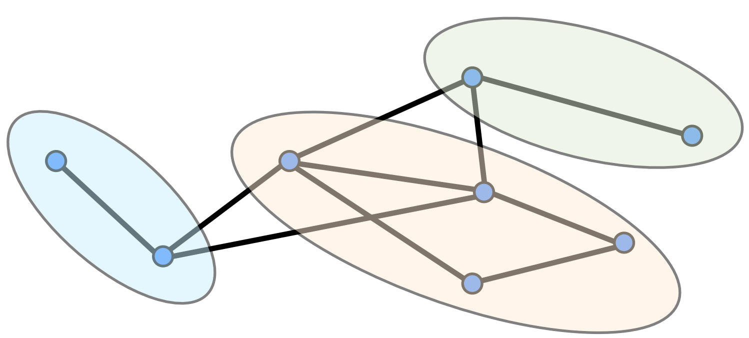
there is a rich literature about data analysis and learning on graphs



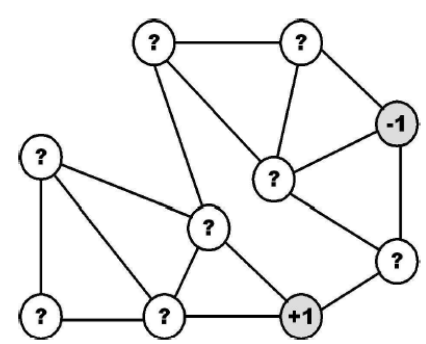
network science (node centrality)



diffusion on graphs

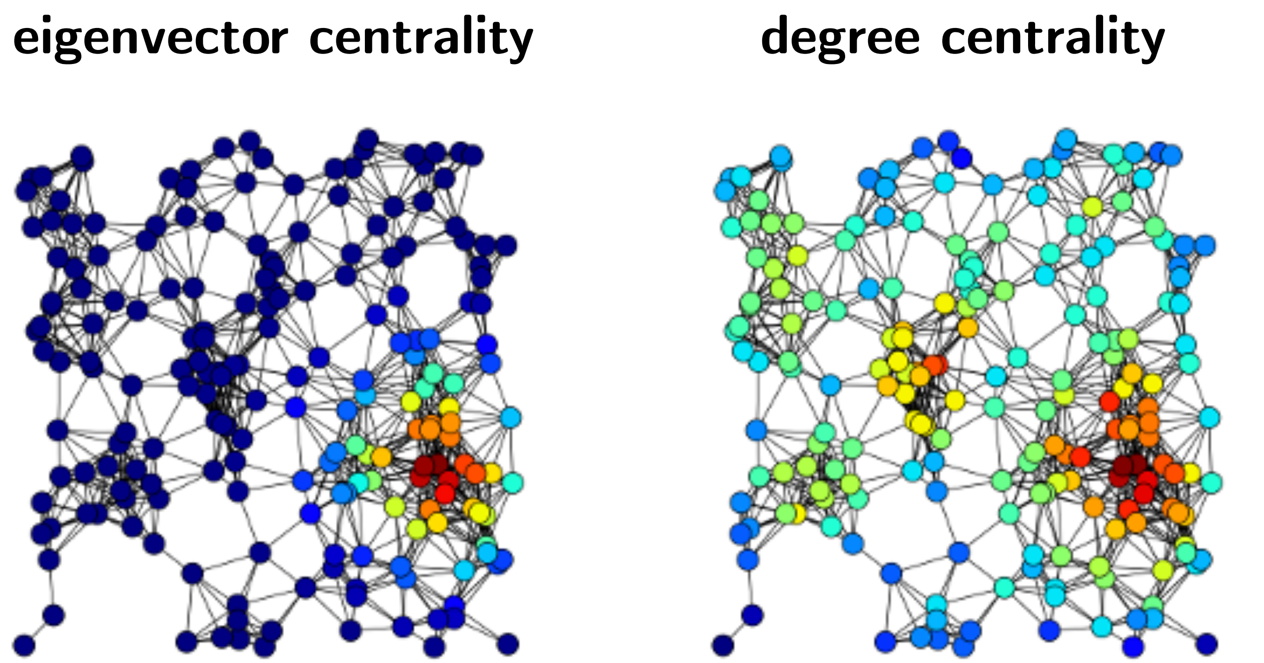


unsupervised learning (dimensionality reduction, clustering)



semi-supervised learning

Network centrality

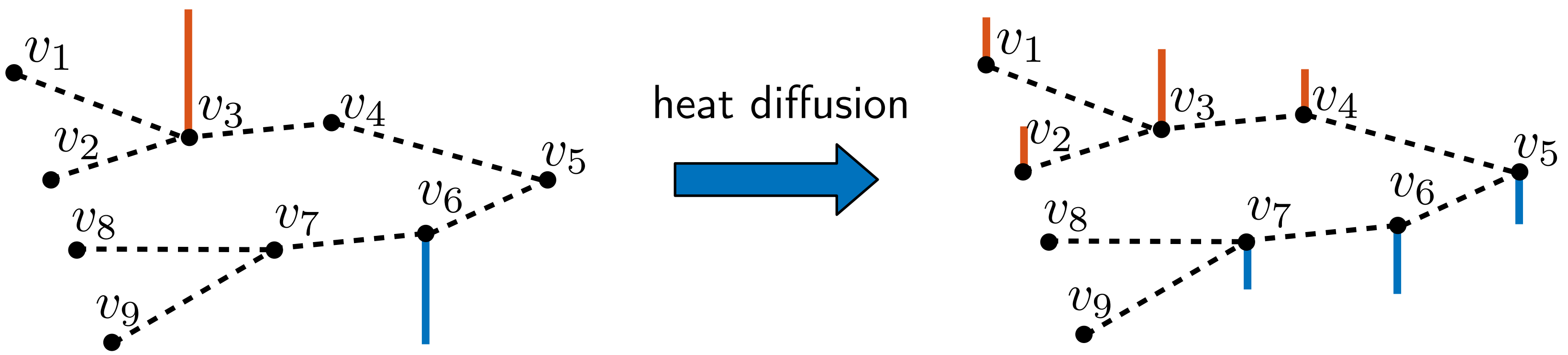


$$Wx = \lambda_{\max} x \qquad d = [d(v_1), \cdots, d(v_N)]$$

- Google's PageRank is a variant of eigenvector centrality
- eigenvectors of W can also be used to provide a frequency interpretation for graph signals

PageRank: http://www.ams.org/publicoutreach/feature-column/fcarc-pagerank
Sandryhaila and Moura, "Discrete signal processing on graphs," IEEE TSP, 2013.

Diffusion on graphs



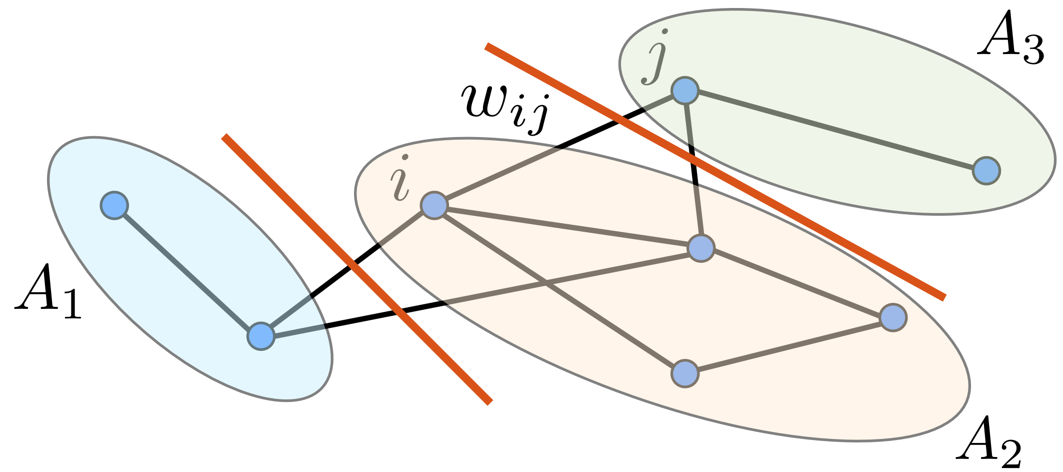
$$\frac{\partial x}{\partial \tau} + Lx = 0$$
$$x(v, 0) = x_0(v)$$

$$\Longrightarrow \quad x(v, \tau) = e^{-\tau L} x_0(v)$$

- heat diffusion on graphs is a typical physical process on graphs
- other possibilities exist (e.g., random walk on graphs)
- many have an interpretation of filtering on graphs

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

Graph clustering (community detection)

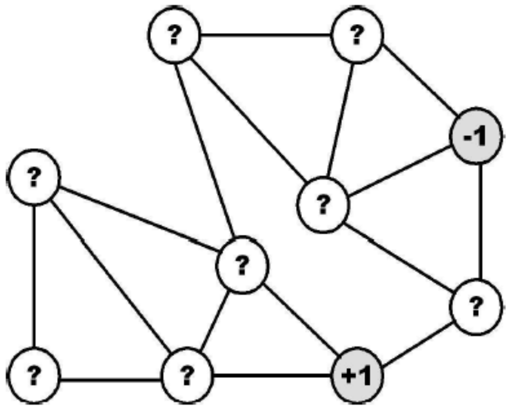


$$NCut(A_1, ..., A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{vol(A_i)}$$

- first k eigenvectors of graph Laplacian provide solution to graph cut minimisation
- eigenvectors of graph Laplacian enable a Fourier-like analysis for graph signals

von Luxburg, "A tutorial on spectral clustering," Statistics and Computing, 2007.

Semi-supervised learning



$$y : \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ +1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

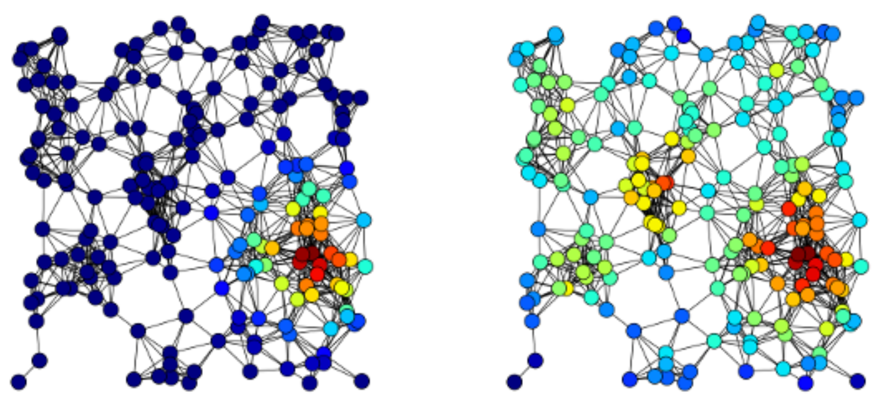
$$\min_{x \in \mathbb{R}^N} ||y - x||_2^2 + \alpha \, x^T L x,$$

- learning by assuming smoothness of predicted labels (label propagation)
- this is equivalent to a denoising problem for graph signal y

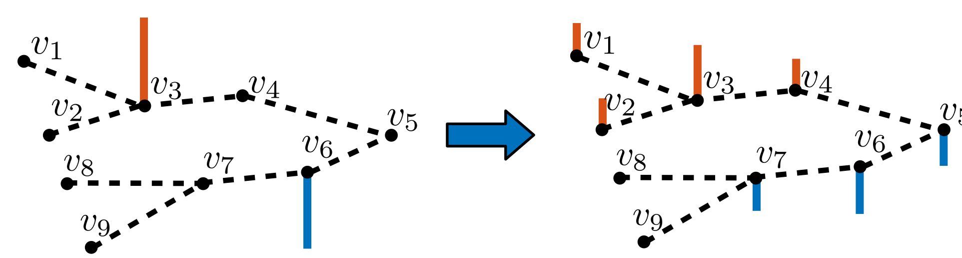
Zhu, "Semi-supervised learning with graphs," Ph.D. dissertation, CMU, 2005.

GSP and the literature

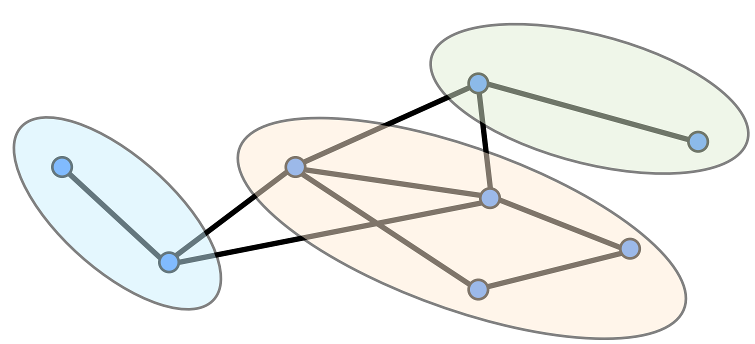
centrality, diffused information, cluster membership, node labels (and node features in general) can ALL be viewed as graph signals



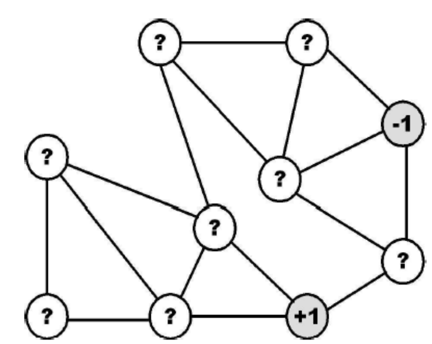
network science



diffusion on graphs



unsupervised learning (dimensionality reduction, clustering)



semi-supervised learning

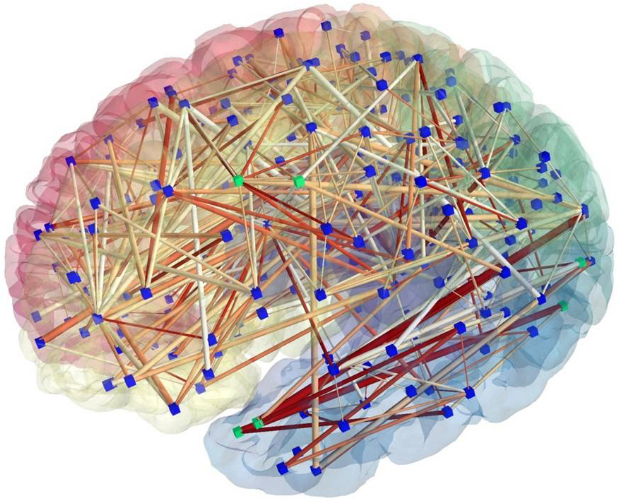
Outline

- Graph signal processing (GSP): Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Connection with literature
- Representation of graph signals
- Applications

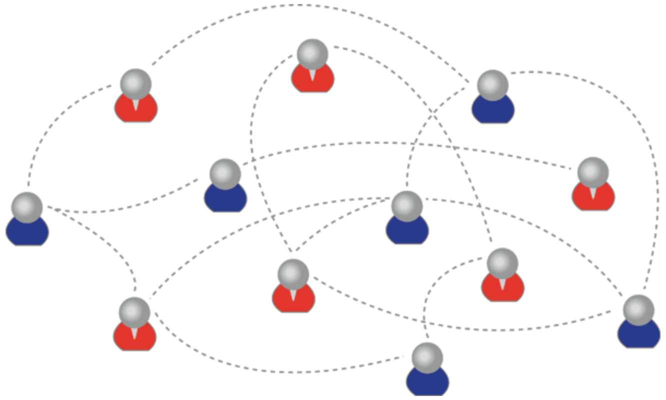
Why representations for graph signals?

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

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



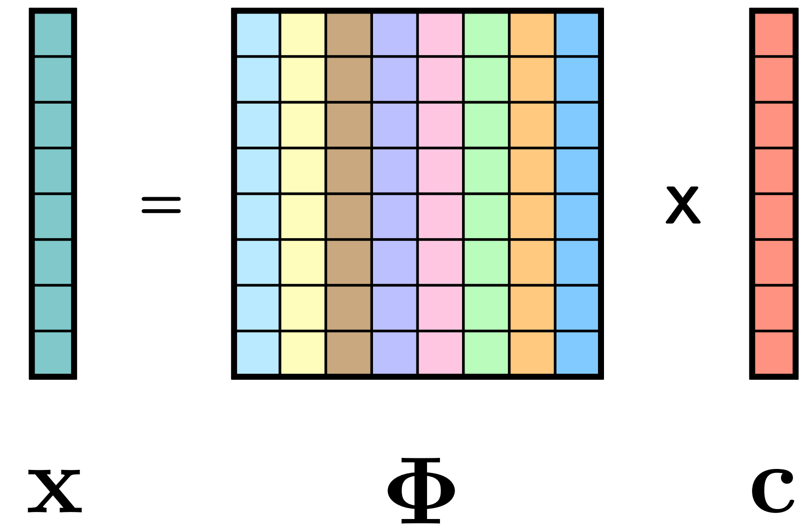
neuroscience (e.g., brain analysis)



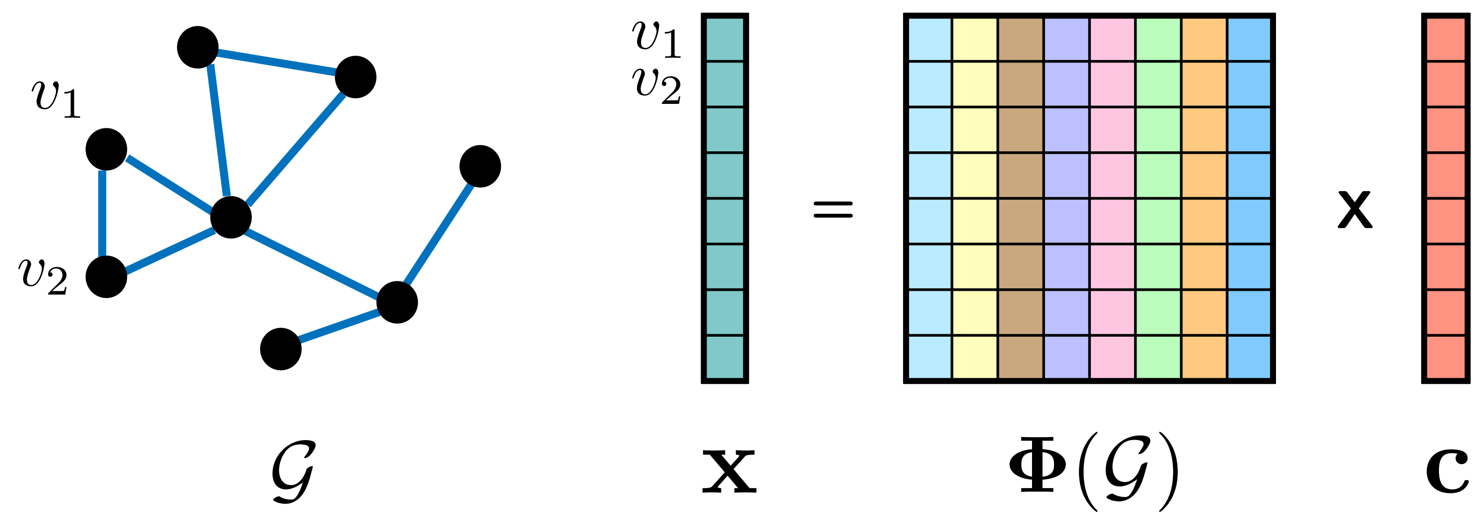
social network analysis (e.g., community, recommendation)

Classical vs. Graph dictionaries

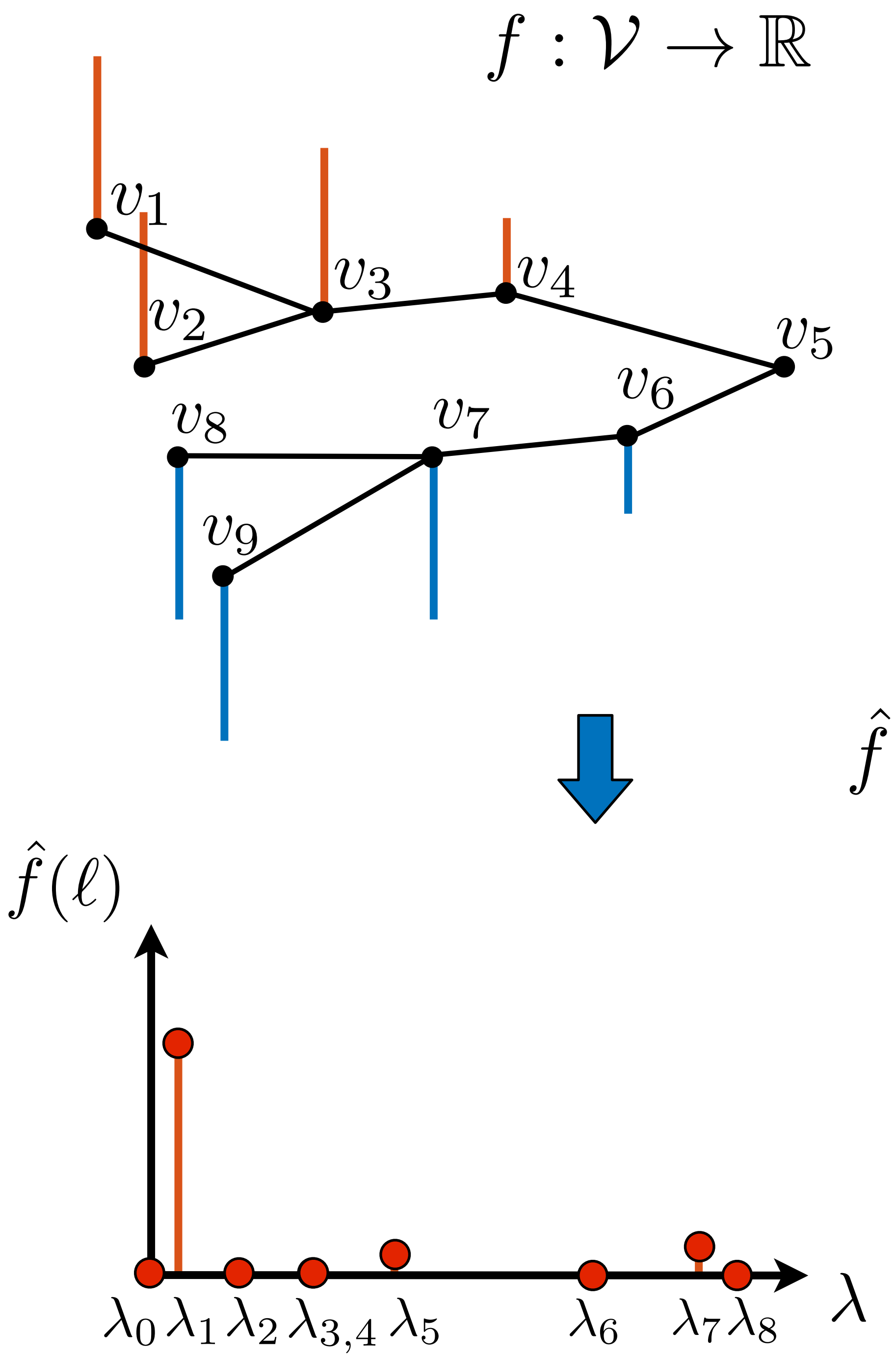
classical signal



graph signal



GFT as a first graph dictionary



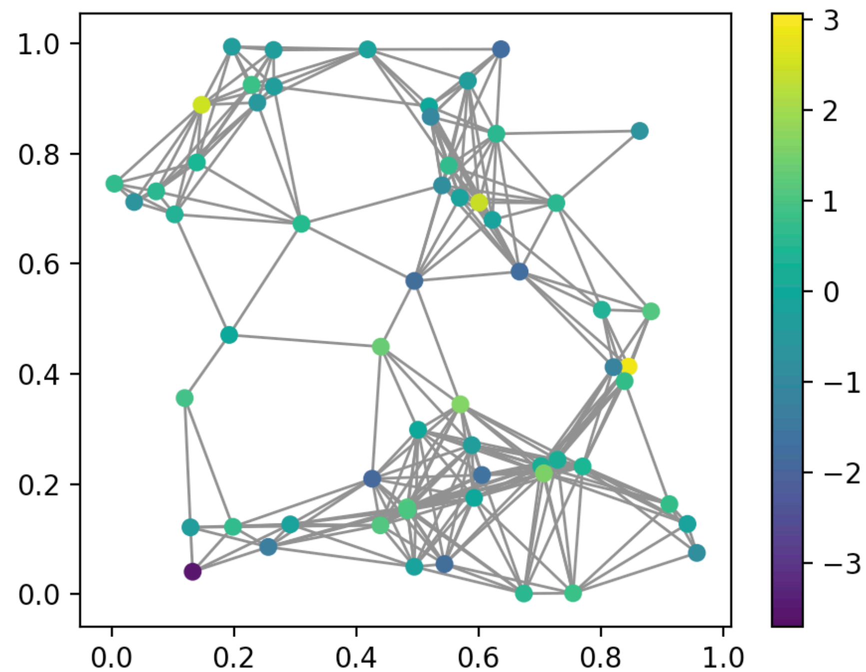
vertex domain

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

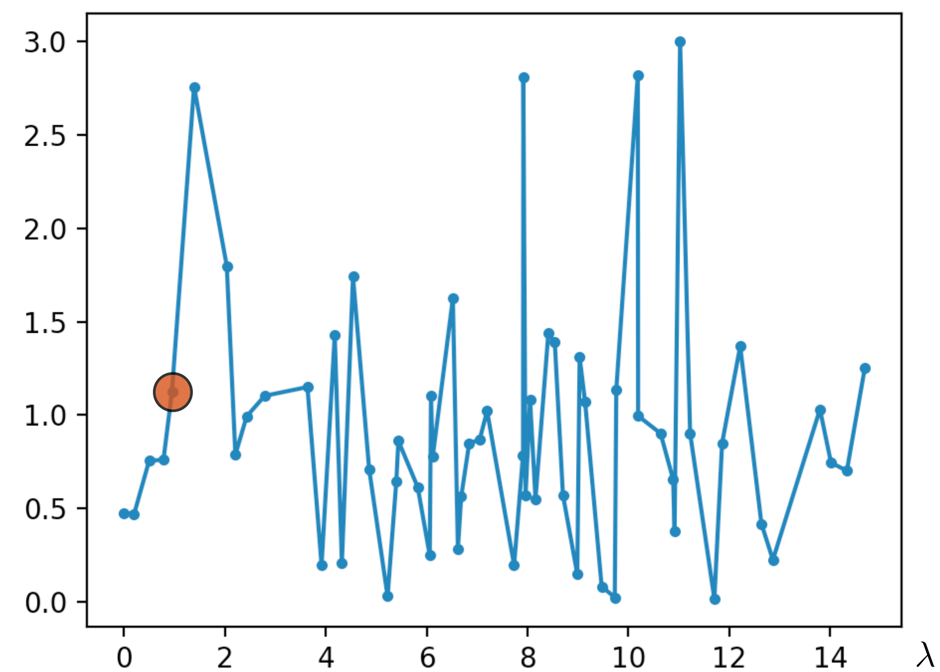
frequency (graph spectral) domain

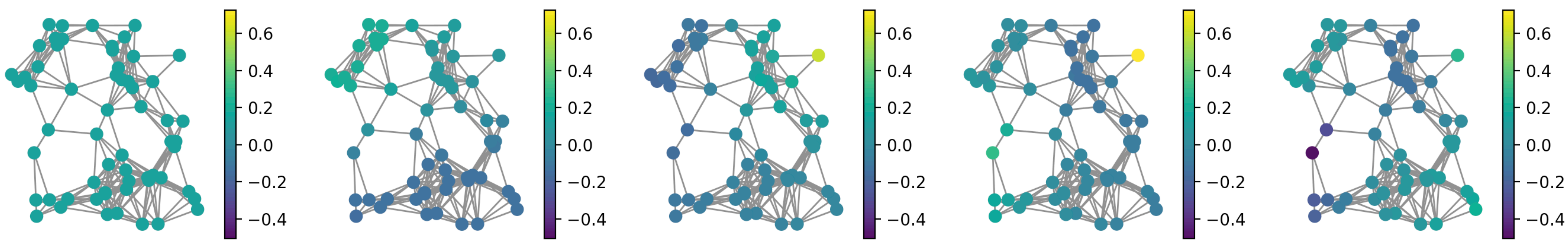
GFT as a first graph dictionary

f



$\hat{f}_\ell$

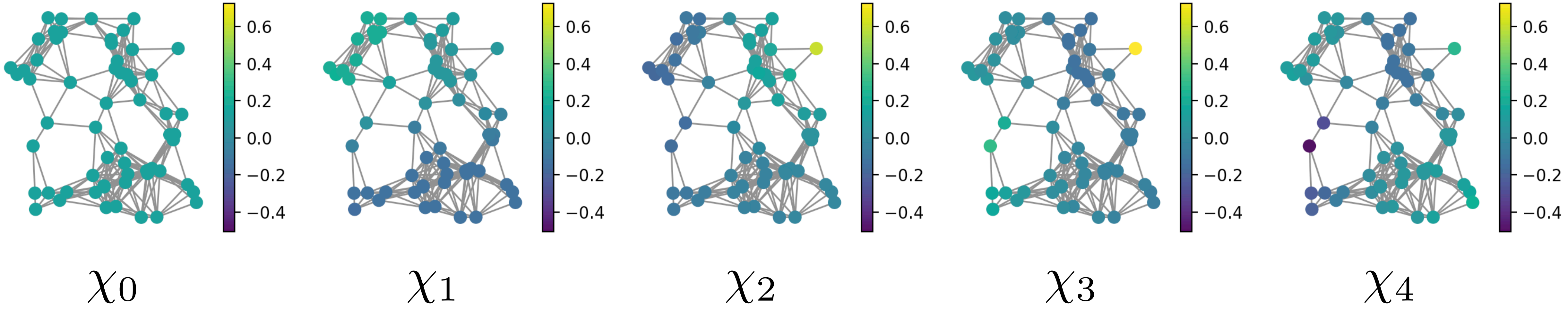


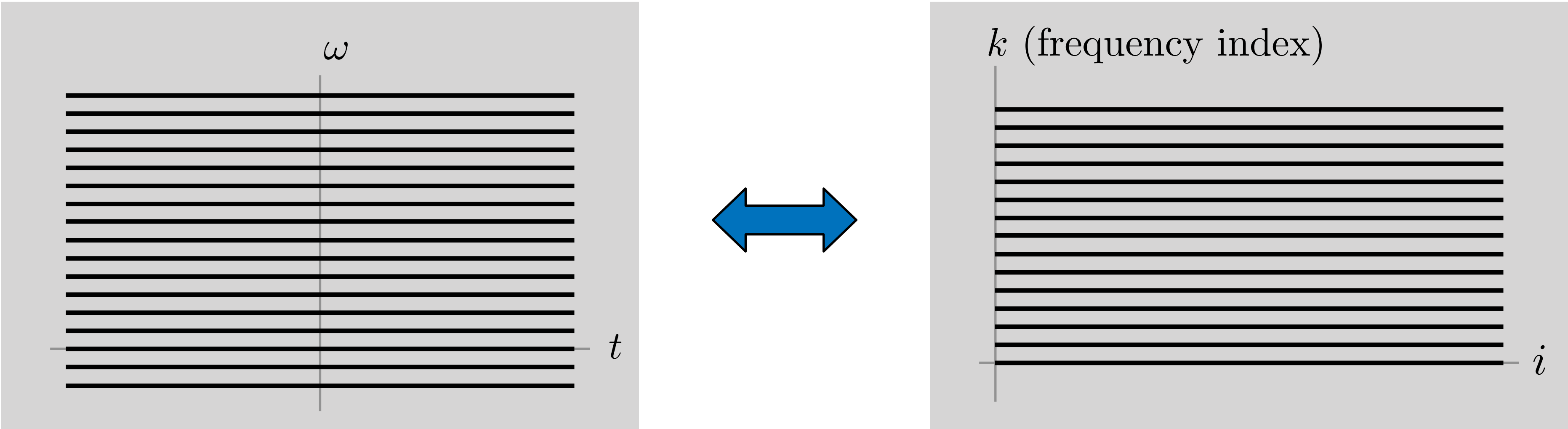


χ_0 χ_1 χ_2 χ_3 χ_4

GFT atoms (corresponding to discrete frequencies)

GFT as a first graph dictionary

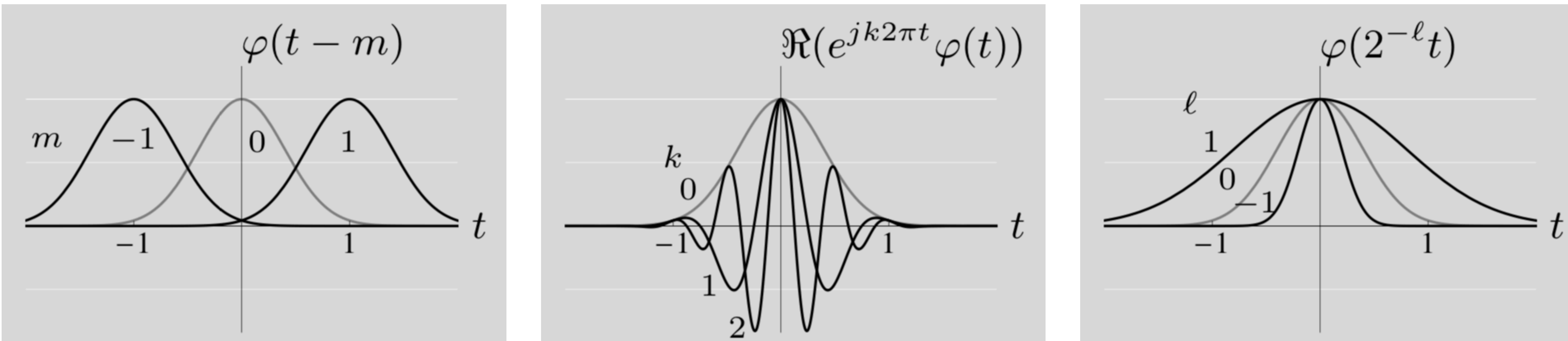




- like complex exponentials in classical FT, eigenvectors in GFT have **global** support
- can we design **localised** atoms on graphs?

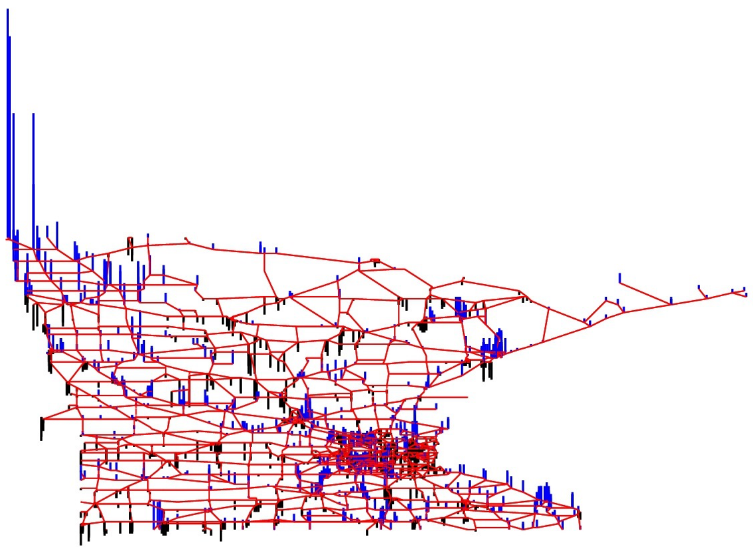
Basic operations for graph signals

basic operations in Euclidean domain



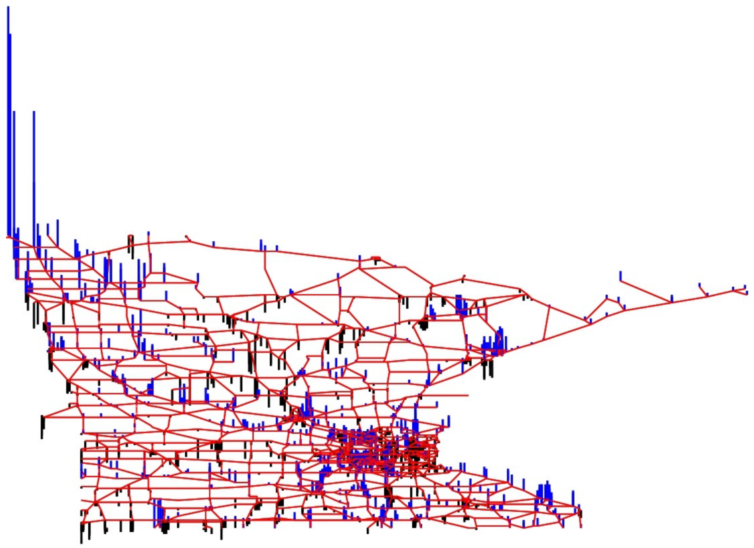
- recall that we used a set of structured functions (e.g., shifted and modulated) to produce localised items

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 of **convolution**, **shift**, **modulation**

Convolution



classical convolution

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

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

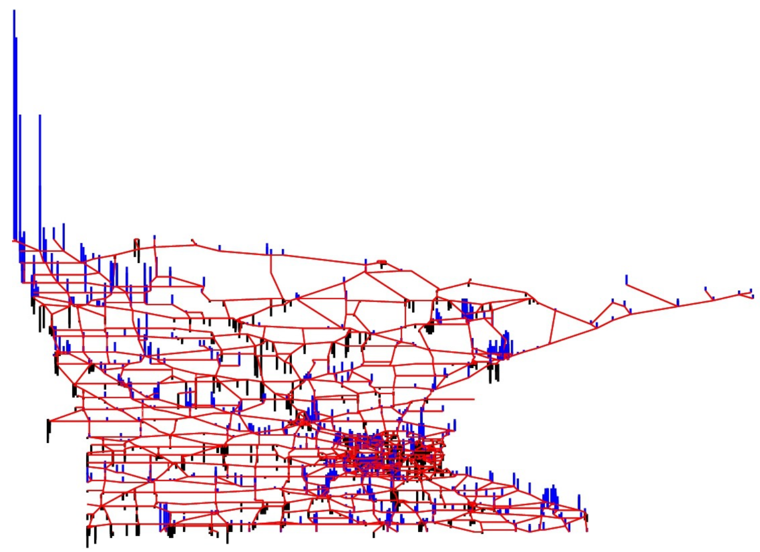
graph convolution **multiplication in graph spectral domain**

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

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

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

Vertex-domain shift



original signal

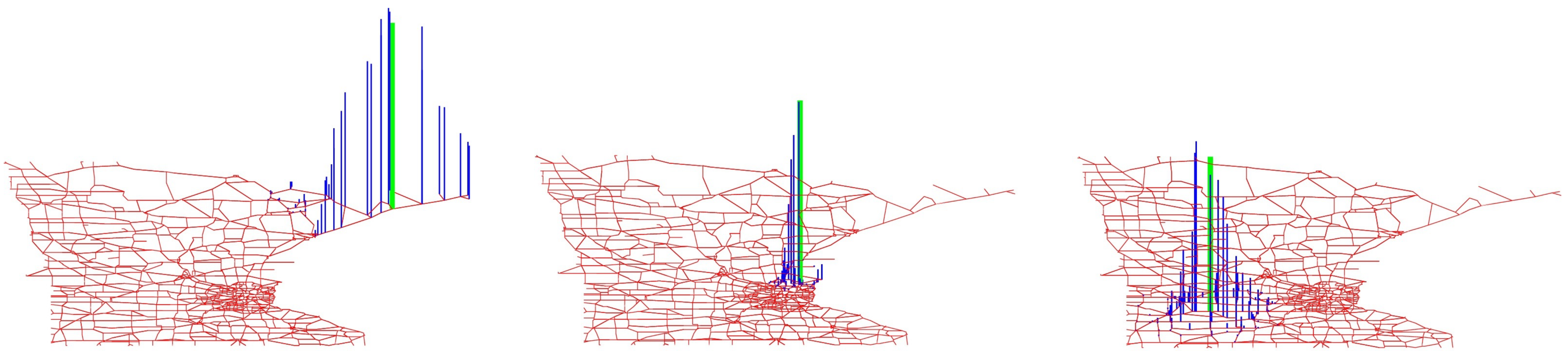
classical shift $(T_u f)(t) := f(t-u) = (f * \delta_u)(t)$

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)

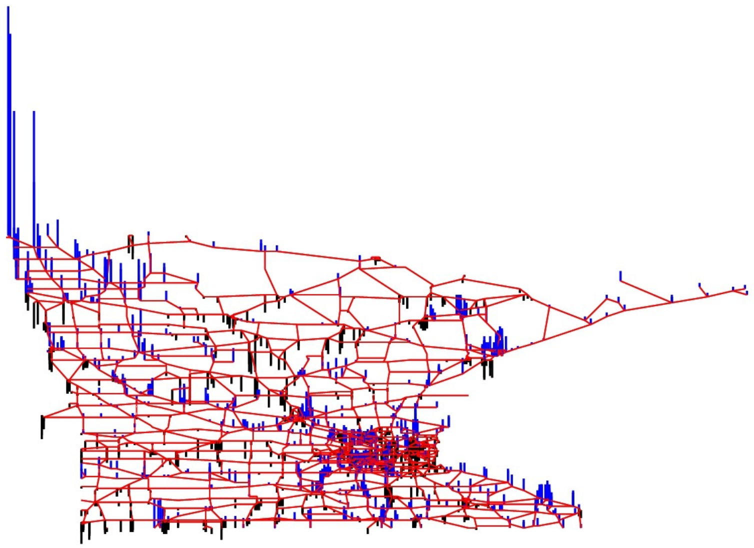
classical shift $(T_u f)(t) := f(t-u) = (f * \delta_u)(t)$

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.

Modulation



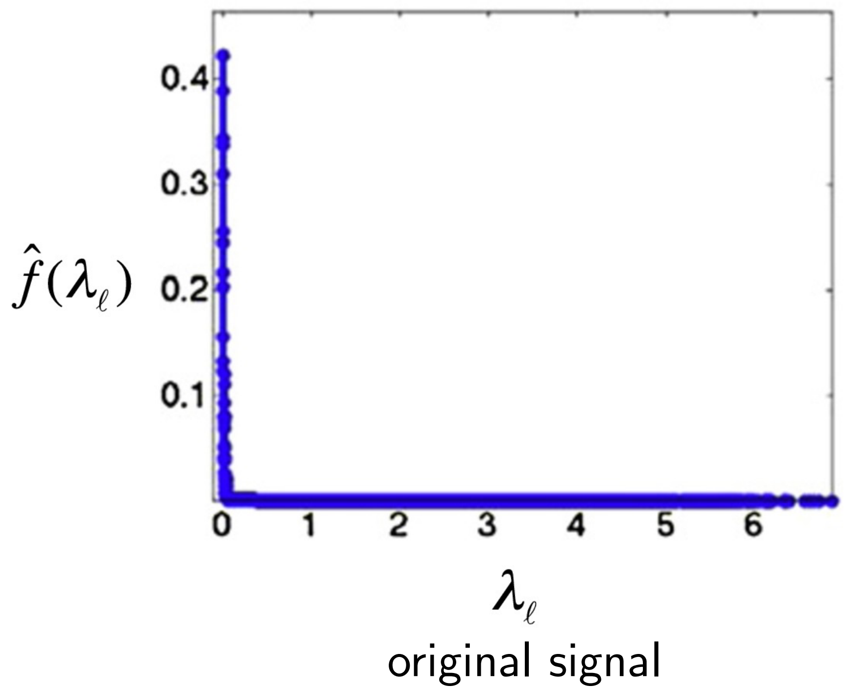
classical modulation $(M_\xi f)(t) := e^{j2\pi\xi t} f(t)$

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

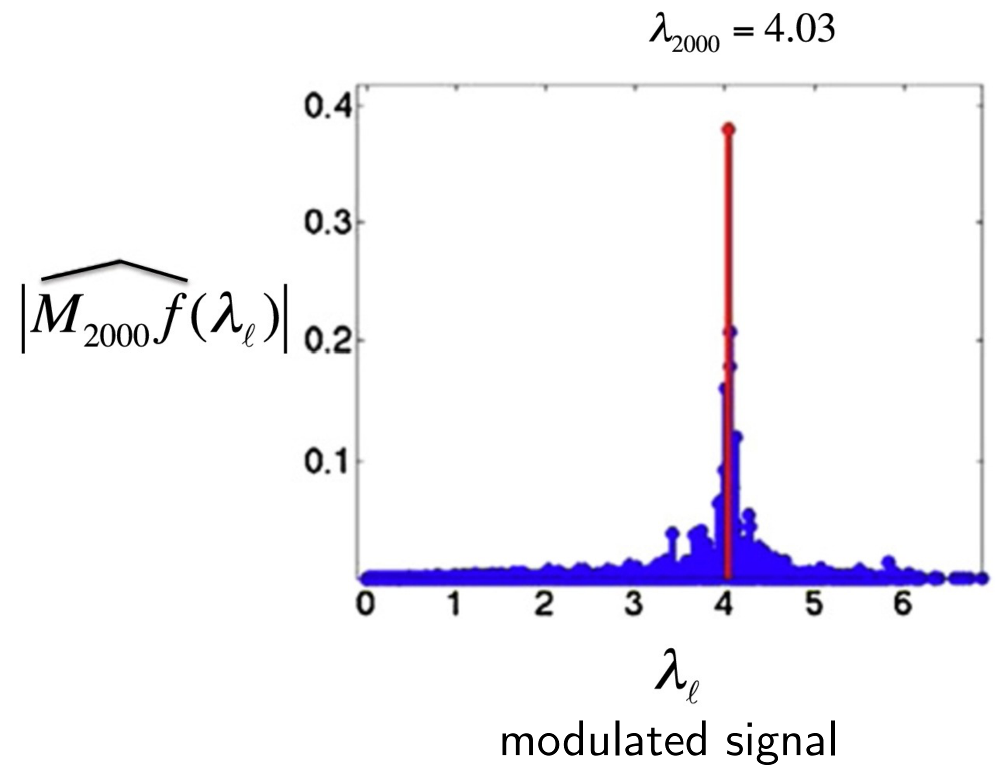
$$(M_k f)(n) := \sqrt{N} f(n) \chi_k(n)$$

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

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**

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

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 now define a windowed graph Fourier transform

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

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

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

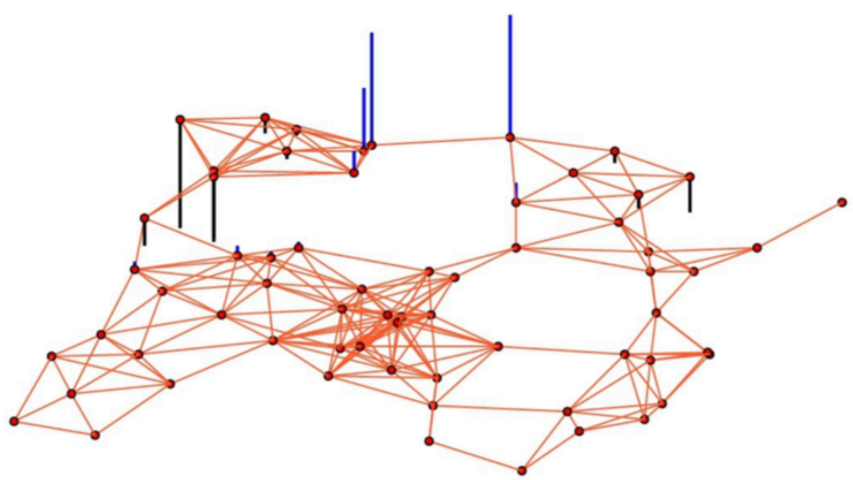
windowed graph Fourier transform (WGFT)

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

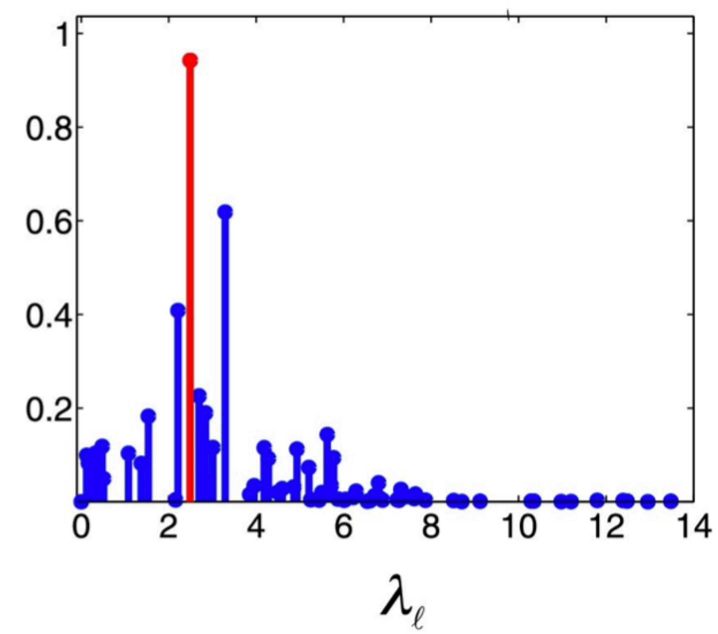
Shuman et al., "Vertex-frequency analysis on graphs," ACHA, 2016.

Windowed graph Fourier transform

$g_{27,11}$



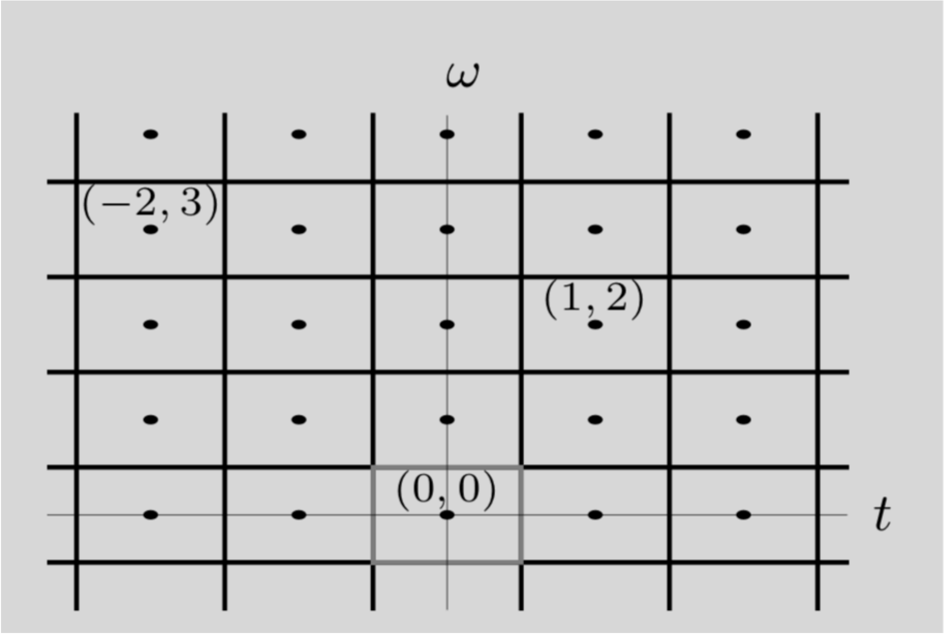
$|\hat{g}_{27,11}|$



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

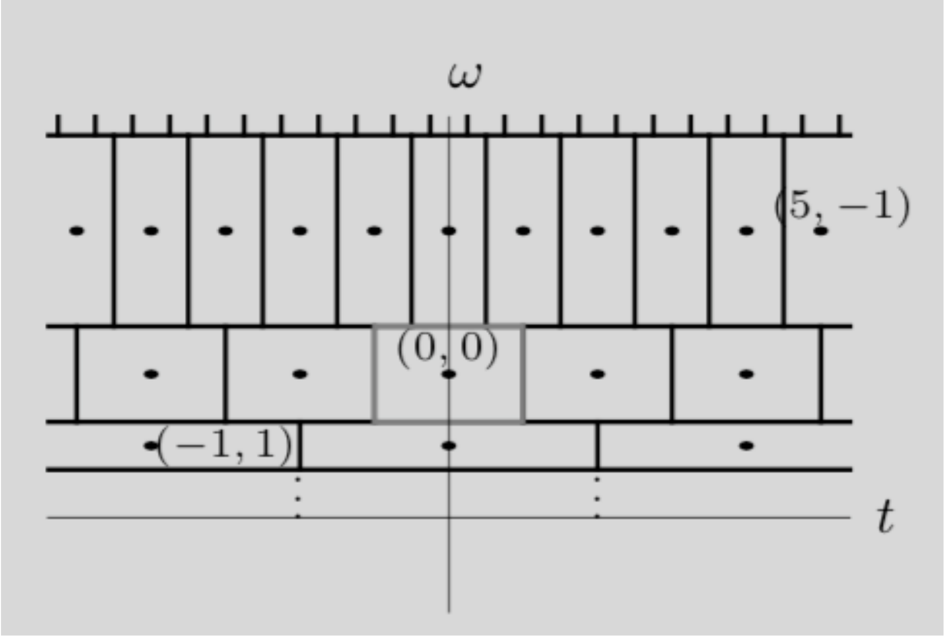
Fourier vs wavelet atoms

Fourier atoms
(time shift and modulation)



$$\varphi_{k,m}(t) = e^{jk\omega_0 t}\varphi(t - mt_0) \quad k, m \in \mathbb{Z}$$

wavelet atoms
(time shift and scaling)



$$\varphi_{l,m}(t) = \varphi(a^{-l}t - mt_0) \quad l, m \in \mathbb{Z}$$

Vetterli et al., "Foundations of signal processing," Cambridge University Press, 2014.

Spectral graph wavelet transform

$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right)$$

$$W_f(s,a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\,dx$$

$$\bar{\psi}_s(x) = \frac{1}{s}\psi^*\left(\frac{-x}{s}\right)$$

$$(T^s f)(a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\,dx = \int_{-\infty}^{\infty} \bar{\psi}_s(a-x) f(x)\,dx$$

$$= (\bar{\psi}_s \star f)(a)$$

$$\widehat{T^s f}(\omega) = \hat{\bar{\psi}}_s(\omega)\hat{f}(\omega) = \hat{\psi}^*(s\omega)\hat{f}(\omega)$$

$$(T^s f)(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega x}\,\hat{\psi}^*(s\omega)\,\hat{f}(\omega)\,d\omega$$

Fourier multiplier operator: scaled kernel $\hat{\psi}^*(s\omega)$

SGWT atom

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

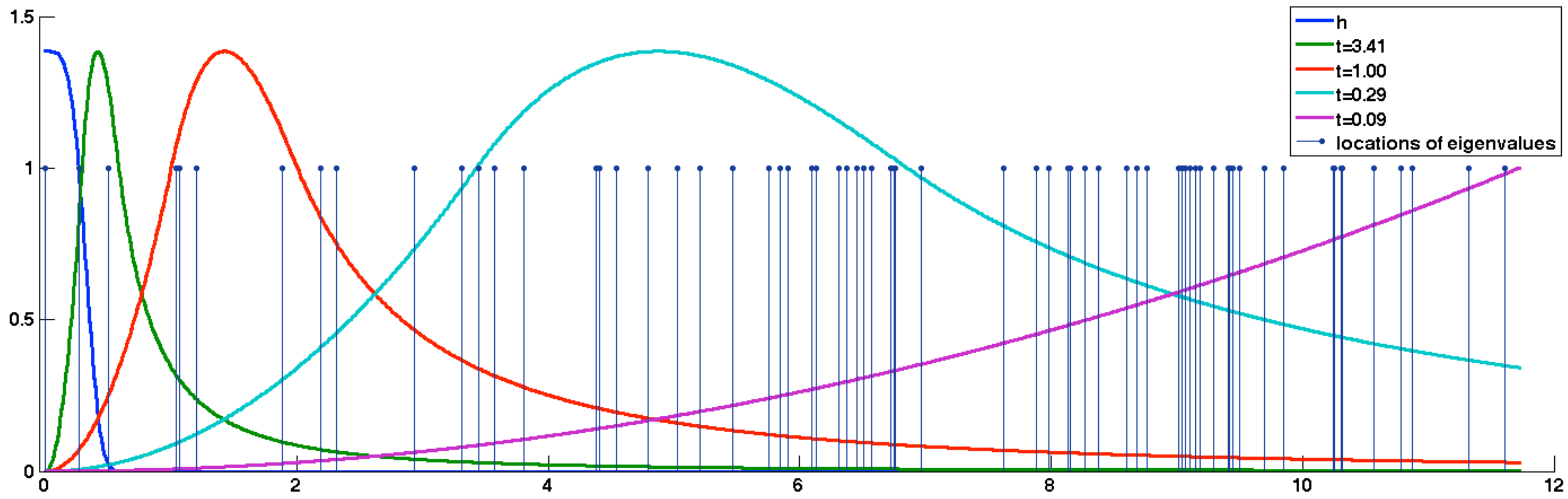
spectral graph wavelet transform (SGWT)

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

$$(\widehat{T_g^s f})(\ell) = \hat{g}(s\lambda_\ell)\hat{f}(\ell)$$

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

Spectral graph wavelet transform



SGWT kernel functions $\hat{g}(s\lambda_\ell)$

Spectral graph wavelet transform

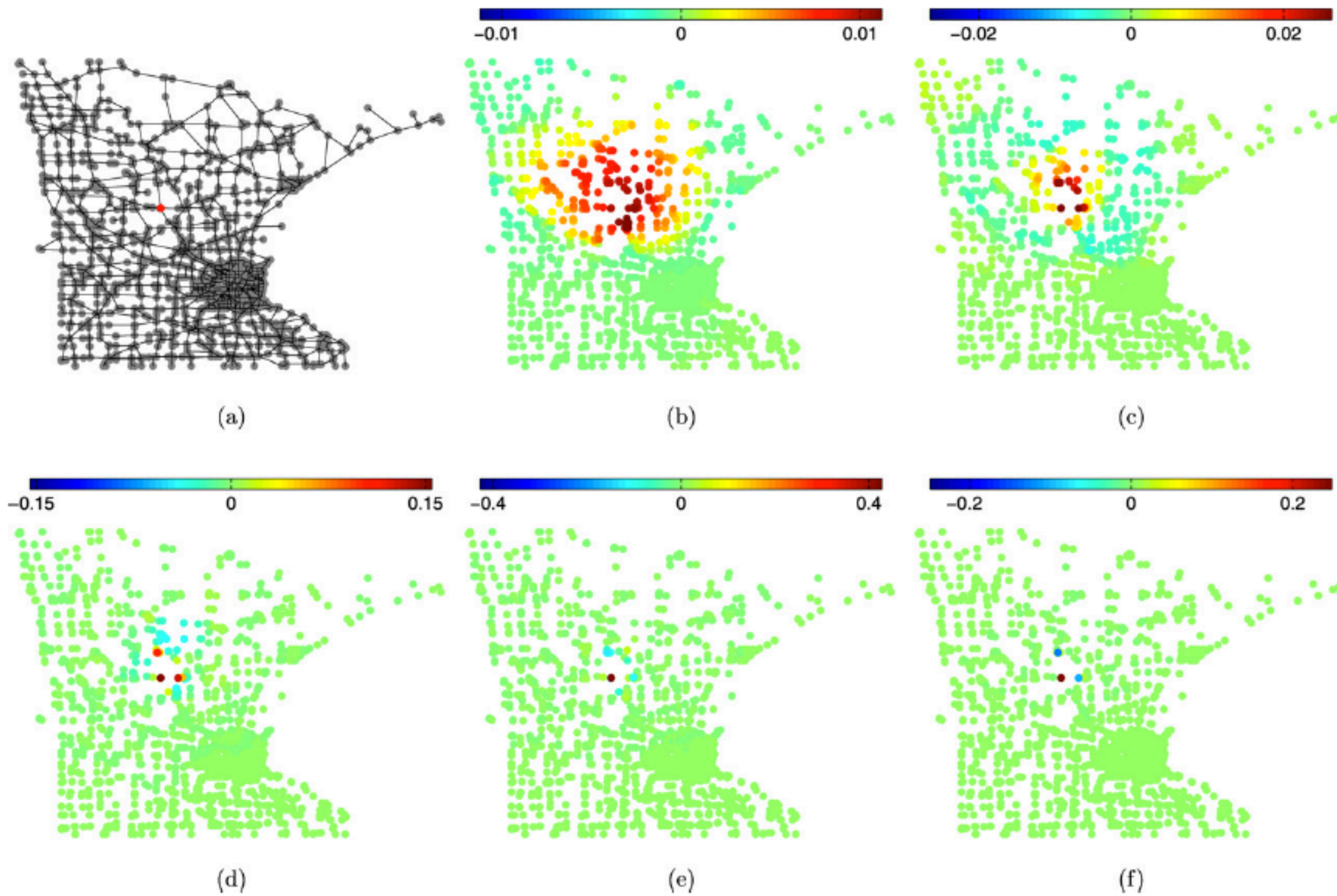


Fig. 4. Spectral graph wavelets on Minnesota road graph, with $K = 100$, $J = 4$ scales. (a) Vertex at which wavelets are centered, (b) scaling function, (c)–(f) wavelets, scales 1–4.

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

Spectral graph wavelet transform

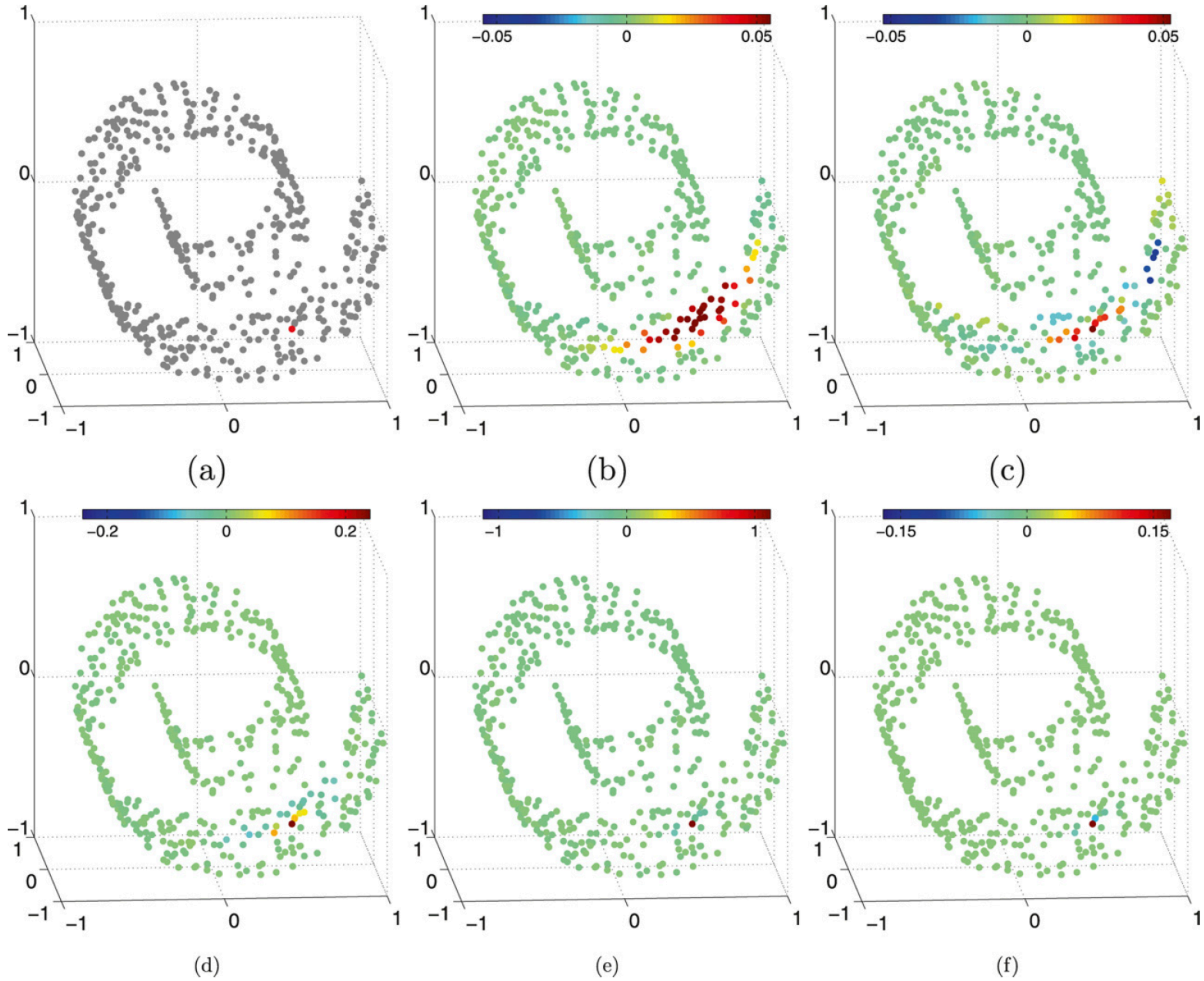
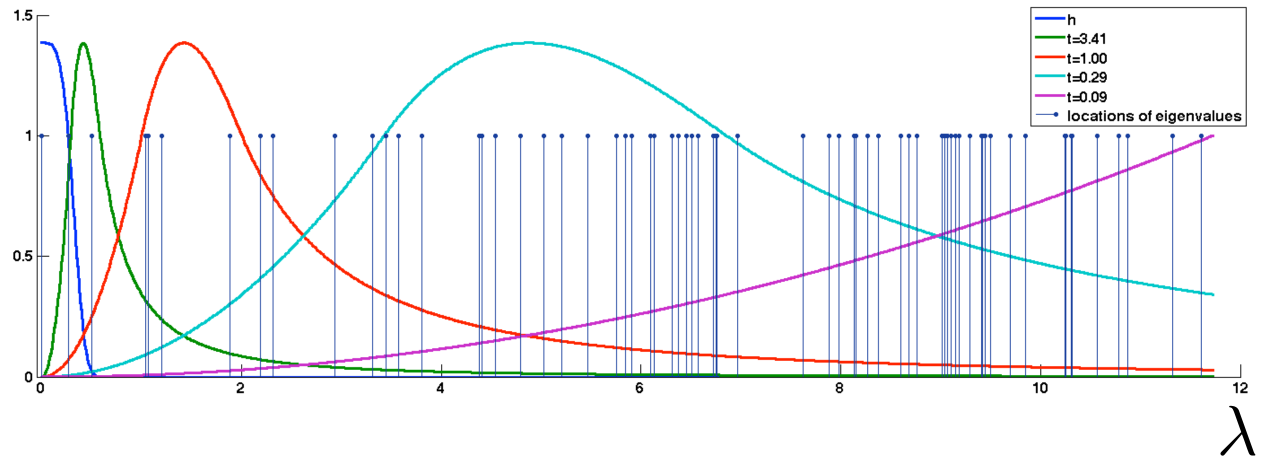
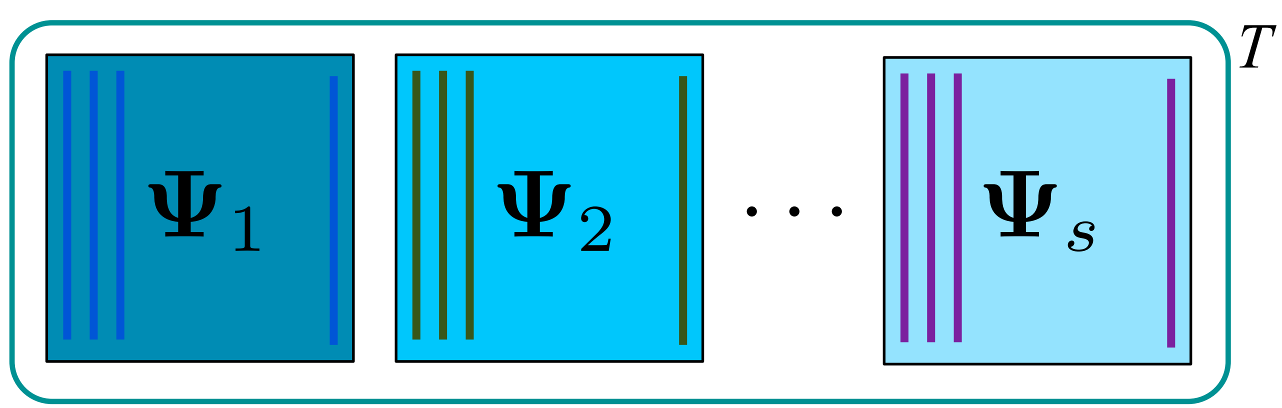


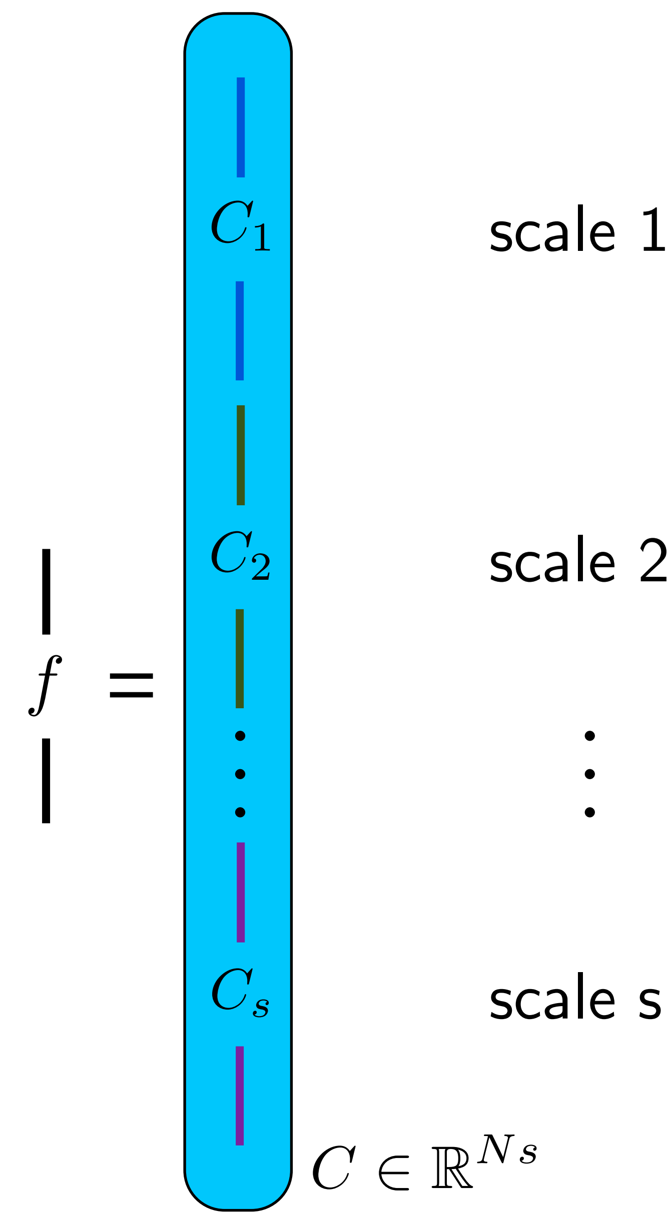
Fig. 3. Spectral graph wavelets on Swiss roll data cloud, with $J = 4$ wavelet scales. (a) Vertex at which wavelets are centered, (b) scaling function, (c)–(f) wavelets, scales 1–4.

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

Spectral graph wavelet transform







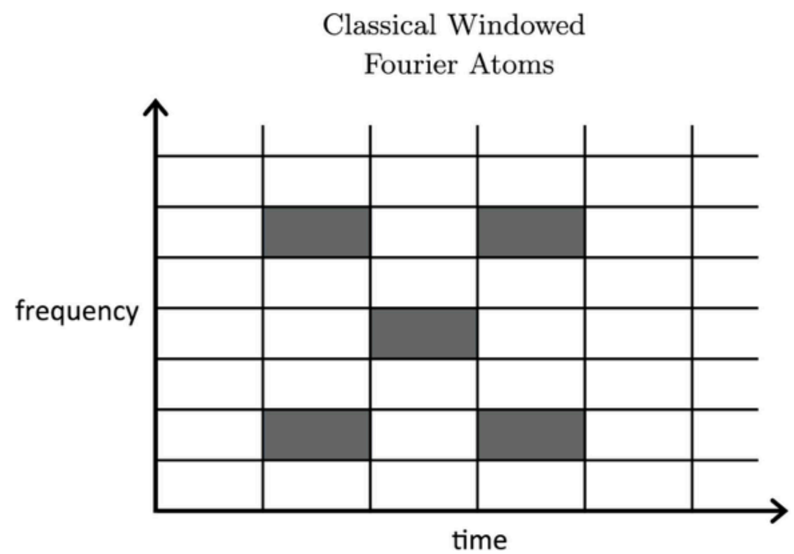
- different "subbands" capture behaviour at **different scales**
- SGWT is an **overcomplete** transform

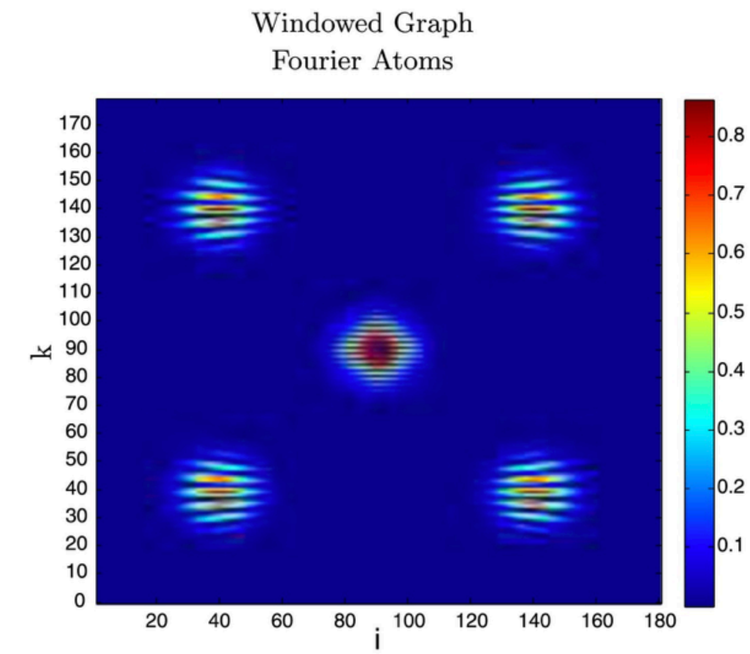
WGFT vs SGWT atoms

WGFT atom

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

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

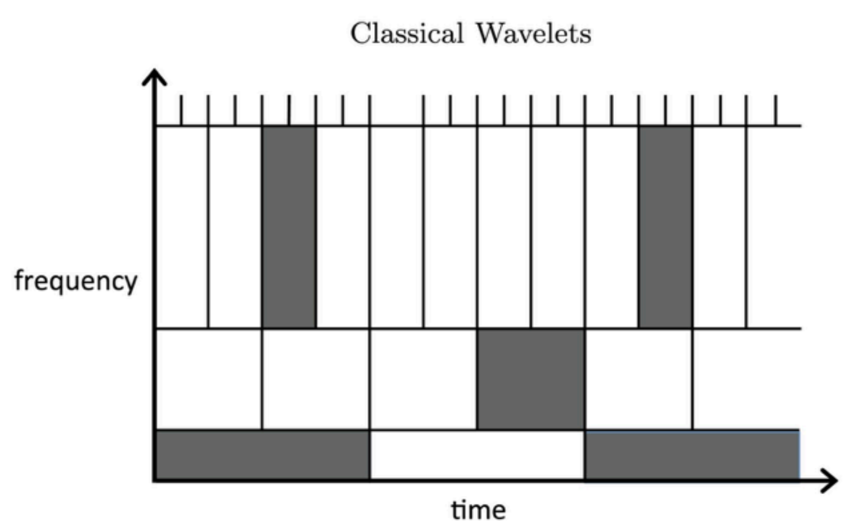


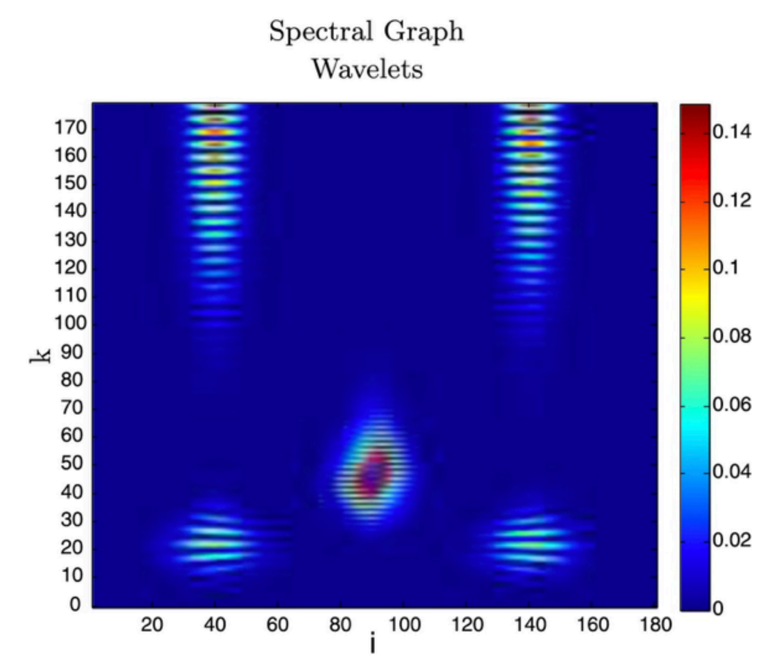


SGWT atom

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

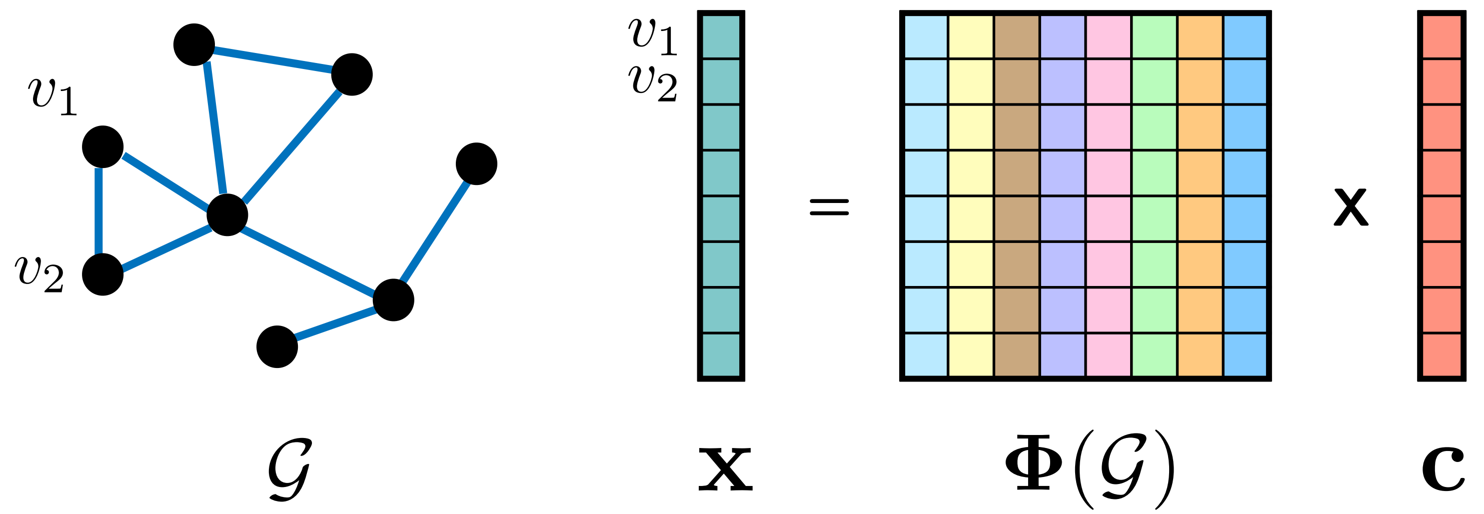
$$= \sum_{\ell=0}^{N-1} \hat{g}(s\lambda_\ell) \chi_\ell^*(i) \chi_\ell(n)$$

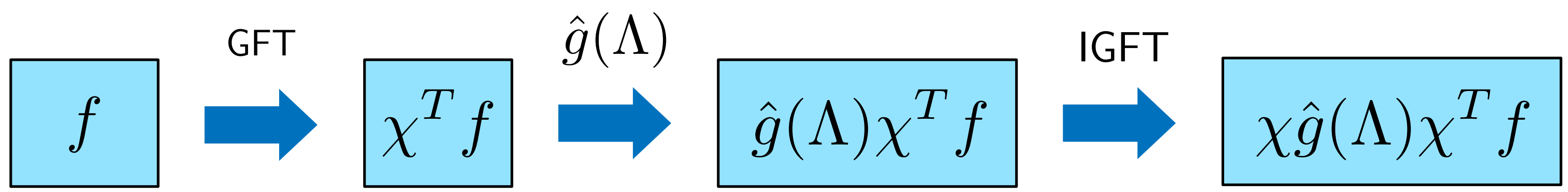




Shuman et al., "Vertex-frequency analysis on graphs," ACHA, 2016.

From analytical to trained graph dictionaries





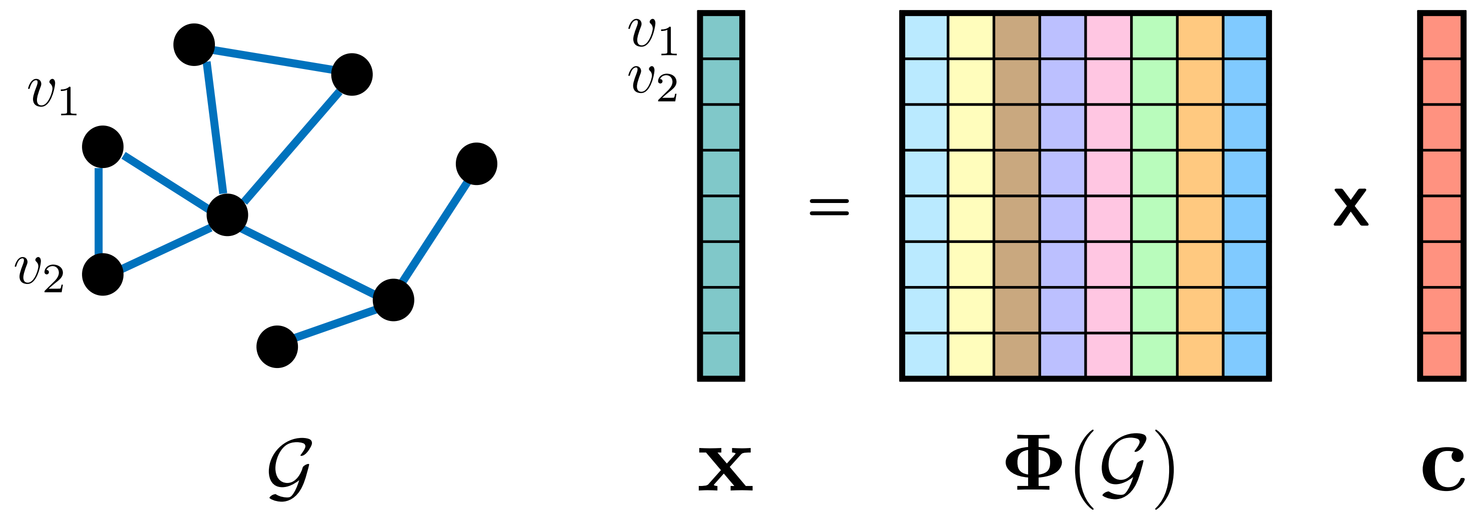
analytical graph dictionaries

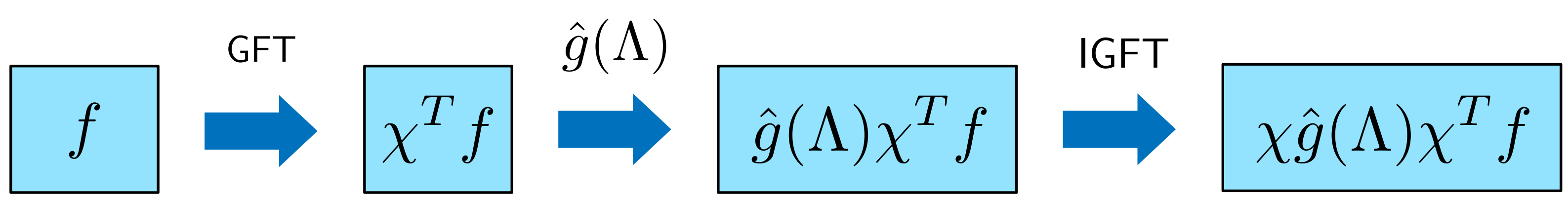
$$\mathbf{\Phi}(\mathcal{G}) = \hat{g}(L) = \chi\hat{g}(\Lambda)\chi^T$$

trained graph dictionaries

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

From analytical to trained graph dictionaries





analytical graph dictionaries

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

trained graph dictionaries

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

A parametric graph dictionary

learning a parametric kernel:

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

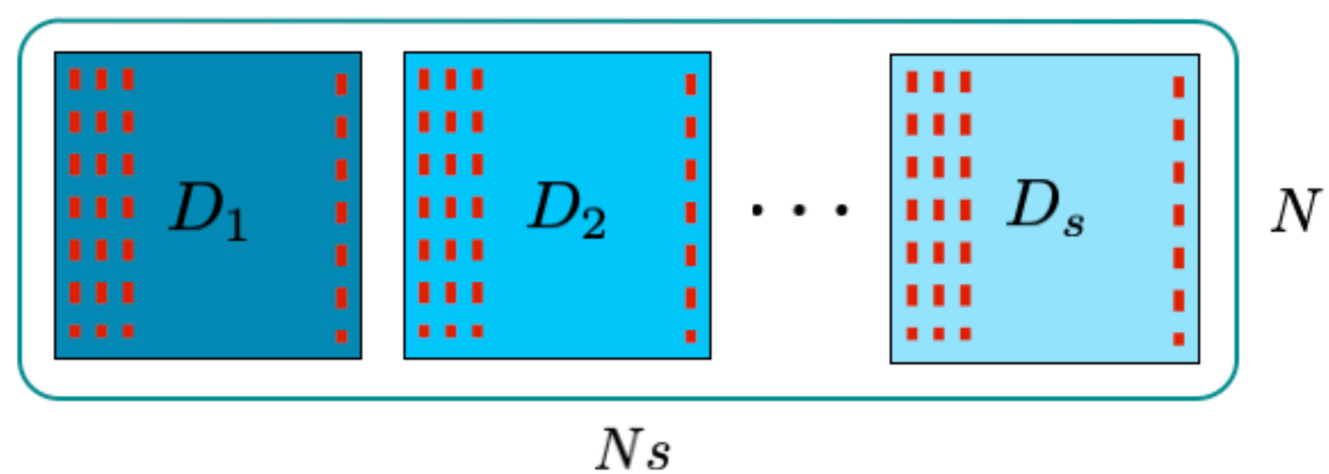
powers of graph Laplacian guarantee localisation

$$\hat{g}_\theta(\Lambda) = \sum_{j=0}^{K} \theta(j)\Lambda^j$$

consider the following dictionary:

$$D = [D_1 \ D_2 \ \cdots \ D_s] = [\chi \hat{g}_{\theta_1}(\Lambda)\chi^T \ \chi \hat{g}_{\theta_2}(\Lambda)\chi^T \ \cdots \ \chi \hat{g}_{\theta_s}(\Lambda)\chi^T]$$

- several filters to identify different **localised** spectral components



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

A parametric graph dictionary

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

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

structured graph dictionaries

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

sparsity constraint

constraints guaranteeing that D is a frame

two-stage iterative approach:

sparse approximation → dictionary updating → sparse approximation → . . . → convergence reached

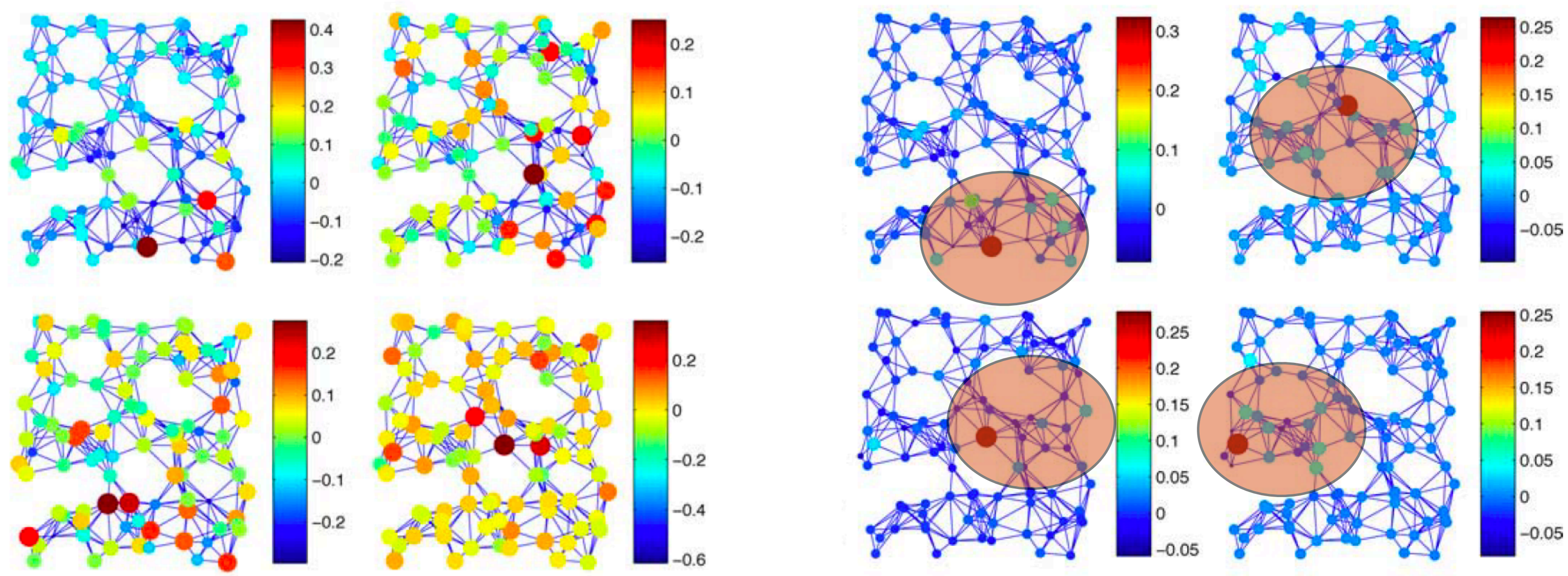
fixing D and solve for C | **fixing C and solve for D** | **fixing D and solve for C**

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

Comparison with non-graph dictionaries

non-graph dictionary atoms (KSVD)

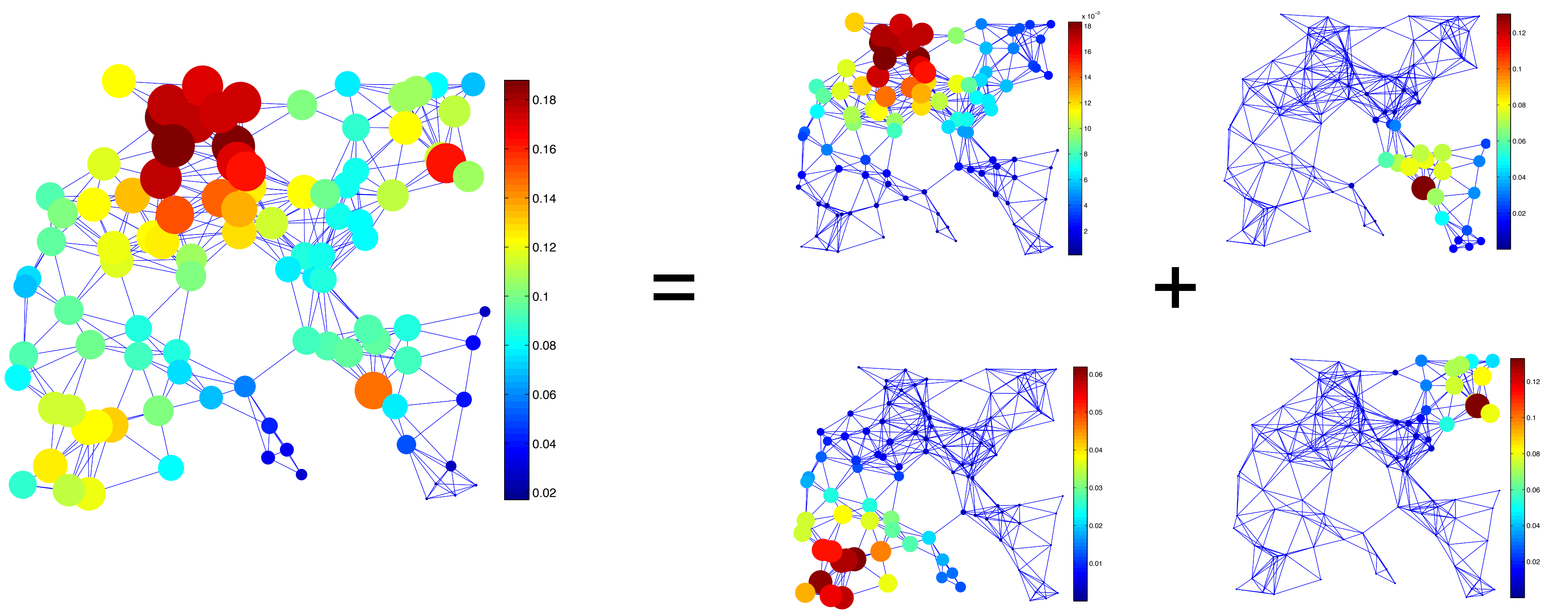
parametric dictionary atoms



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

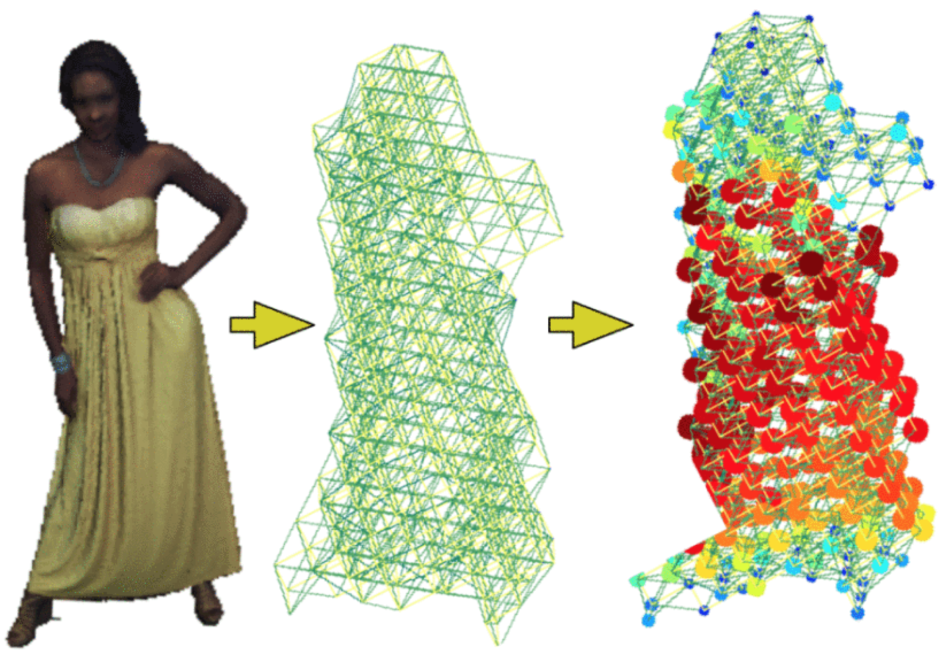
Graph dictionary design - Summary

- 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

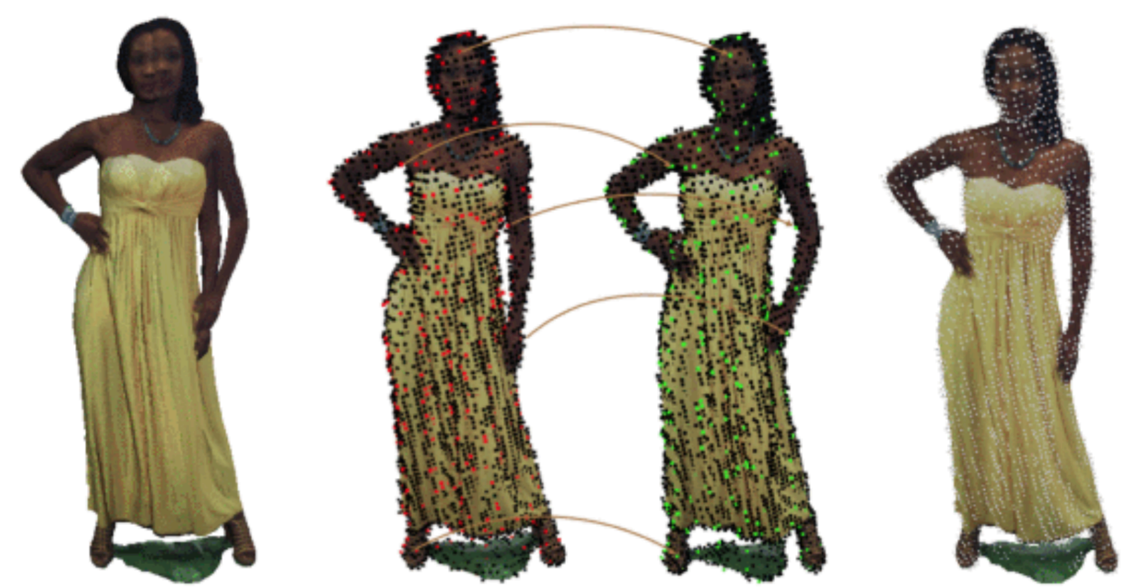
Outline

- Graph signal processing (GSP): Basic concepts
- Graph spectral filtering: Basic tools of GSP
- Connection with literature
- Representation of graph signals
- Applications

Application I: 3D point cloud analysis



(a) (b) (c)



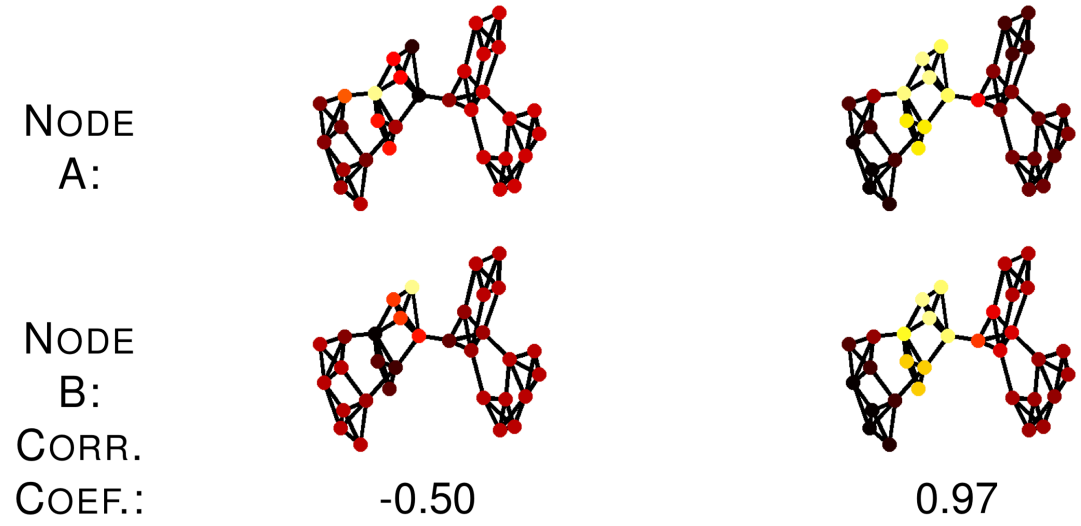
(a) $\mathcal{I}_t + \mathcal{I}_{t+1}$ (b) Correspondence between $\mathcal{I}_t$ and $\mathcal{I}_{t+1}$ (c) $\mathcal{I}_{t,mc} + \mathcal{I}_{t+1}$

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

Application II: Community detection

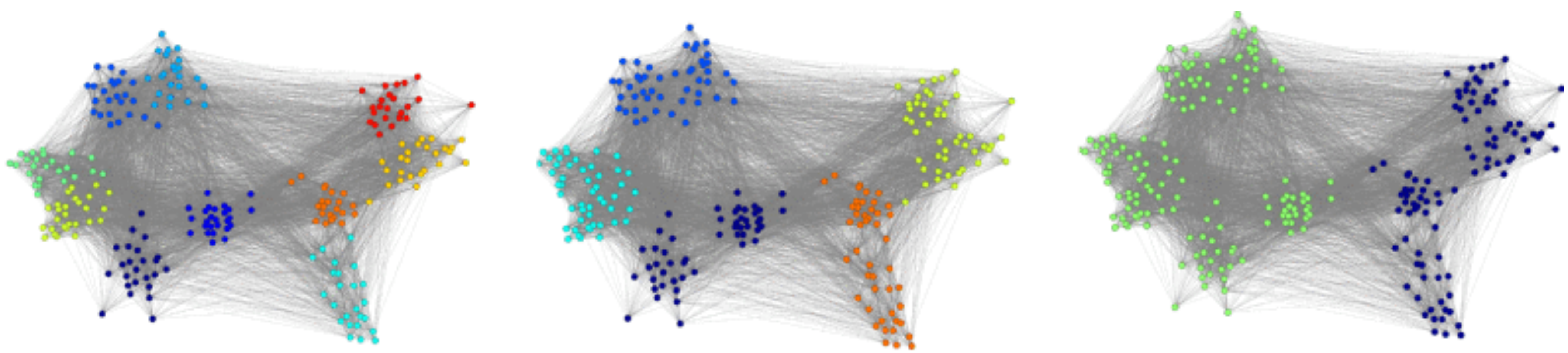
spectral graph wavelets
at different scales:

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



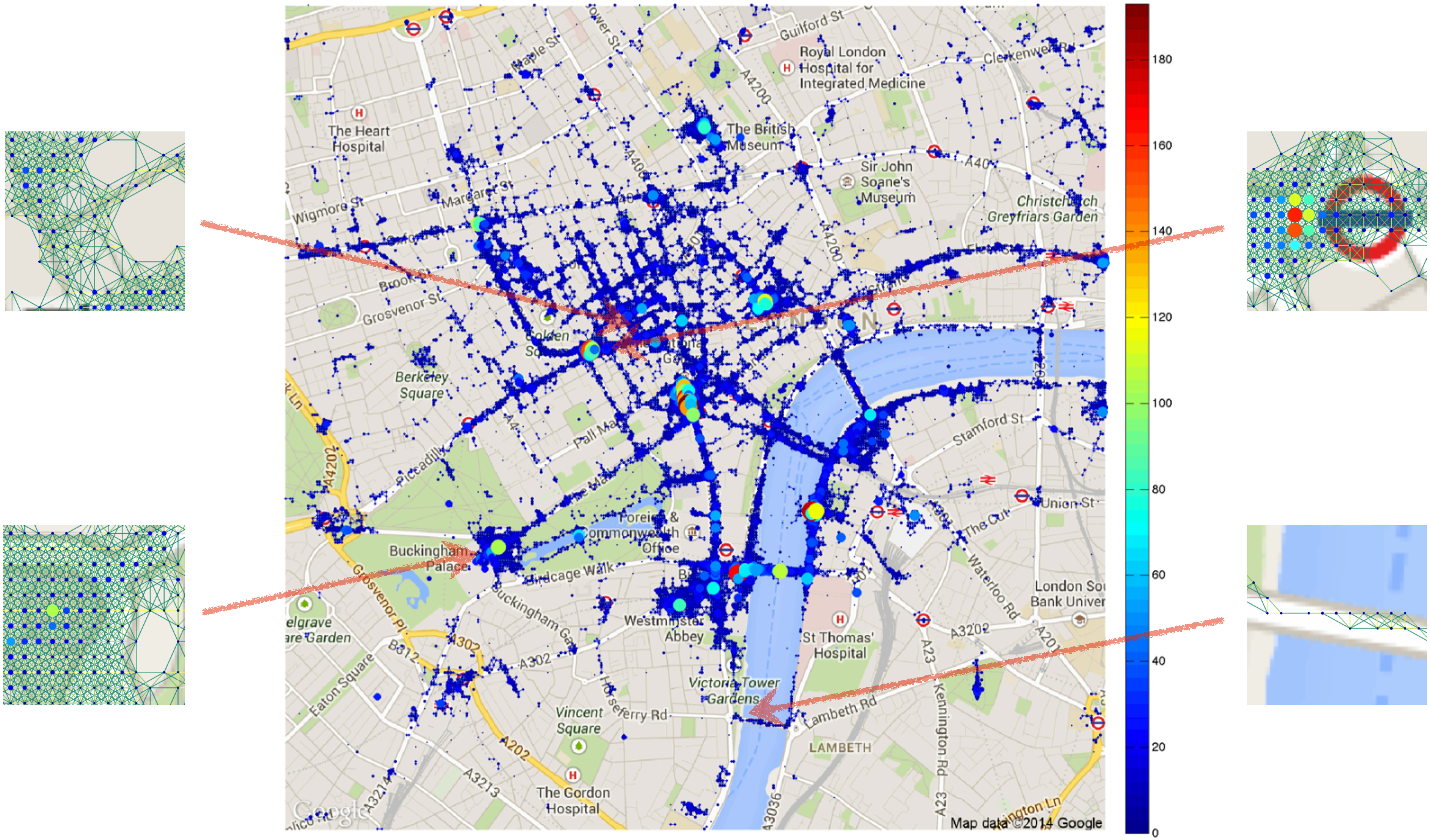
small scale **large scale**

multi-scale community
detection:



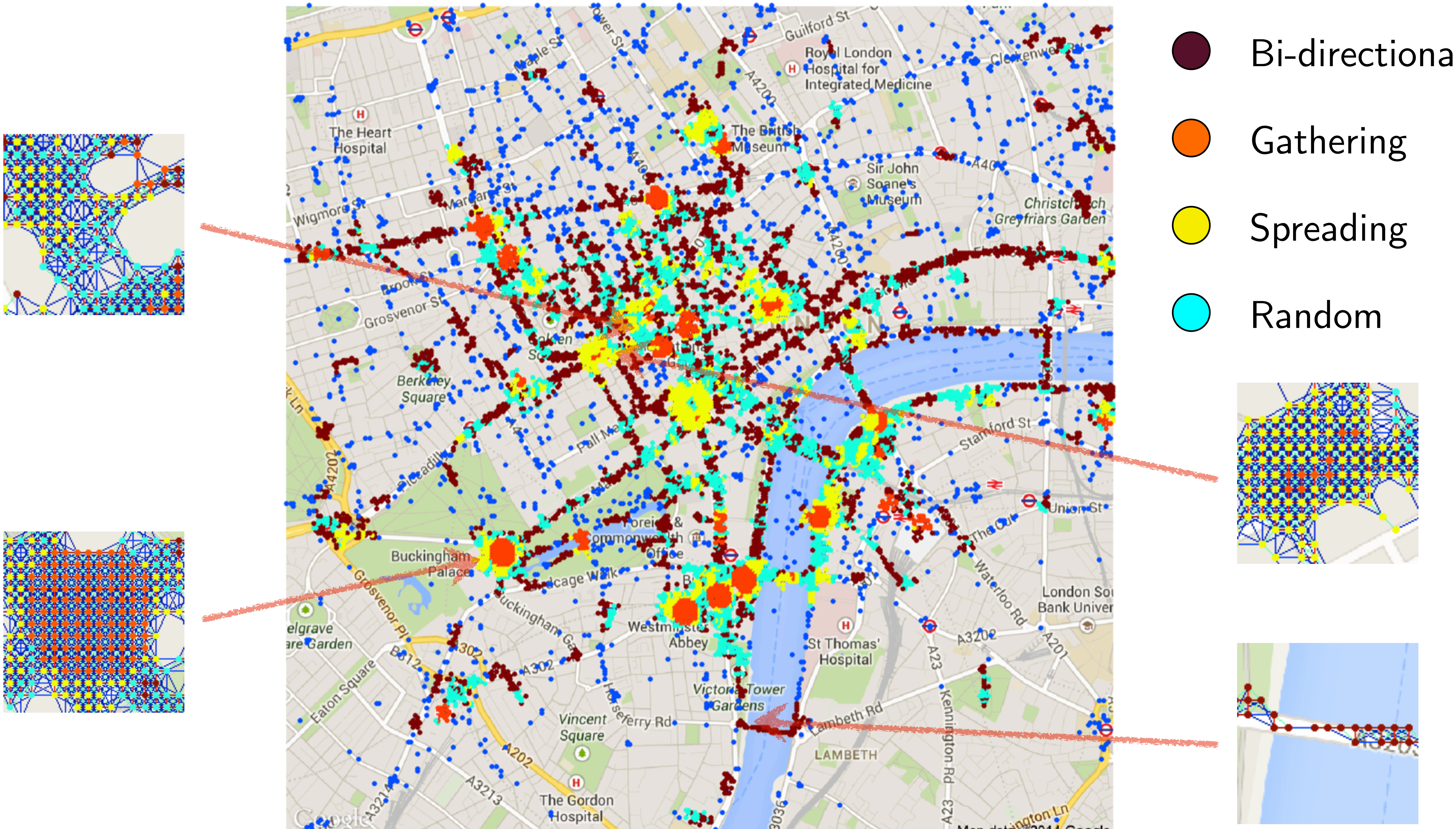
Tremblay et al., "Graph wavelets for multiscale community mining," IEEE TSP, 2014.

Application III: Mobility inference



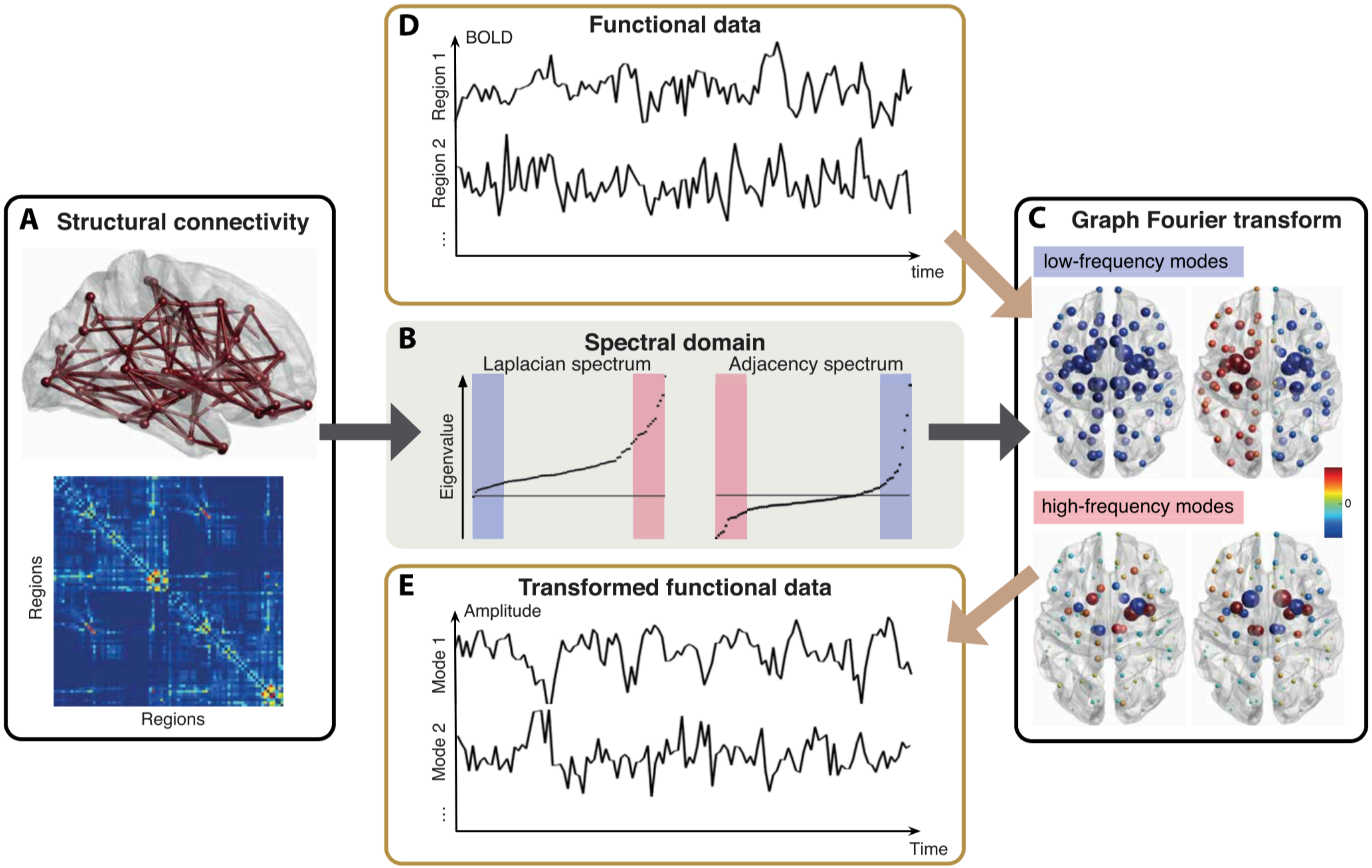
Dong et al., "Inference of mobility patterns via spectral graph wavelets," IEEE ICASSP, 2013.

Application III: Mobility inference



Dong et al., "Inference of mobility patterns via spectral graph wavelets," IEEE ICASSP, 2013.

Application IV: Neuroscience



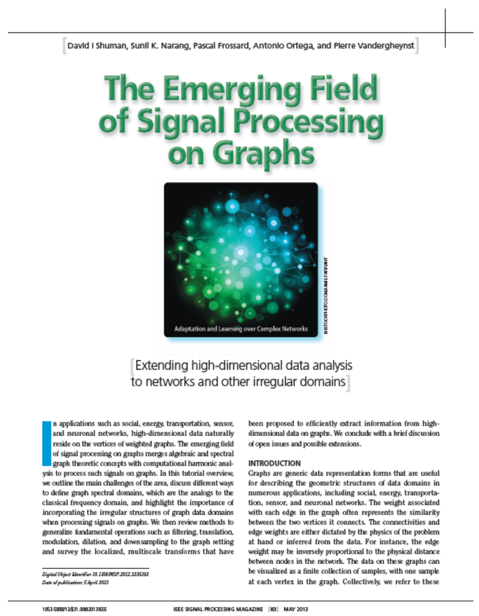
Huang et al., "A graph signal processing perspective on functional brain imaging," Proceedings of the IEEE, 2018.

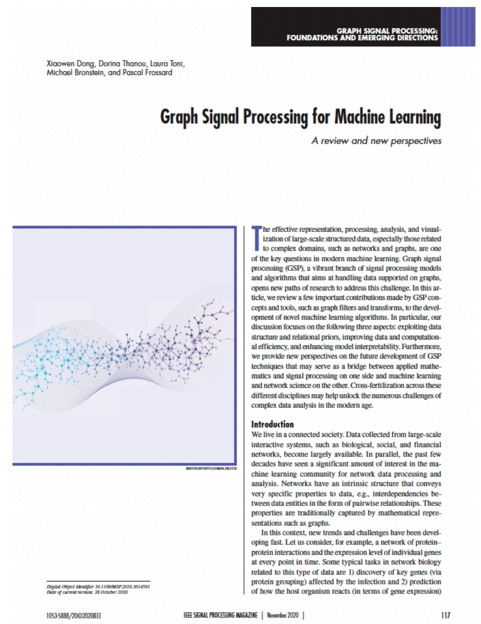
Future of GSP

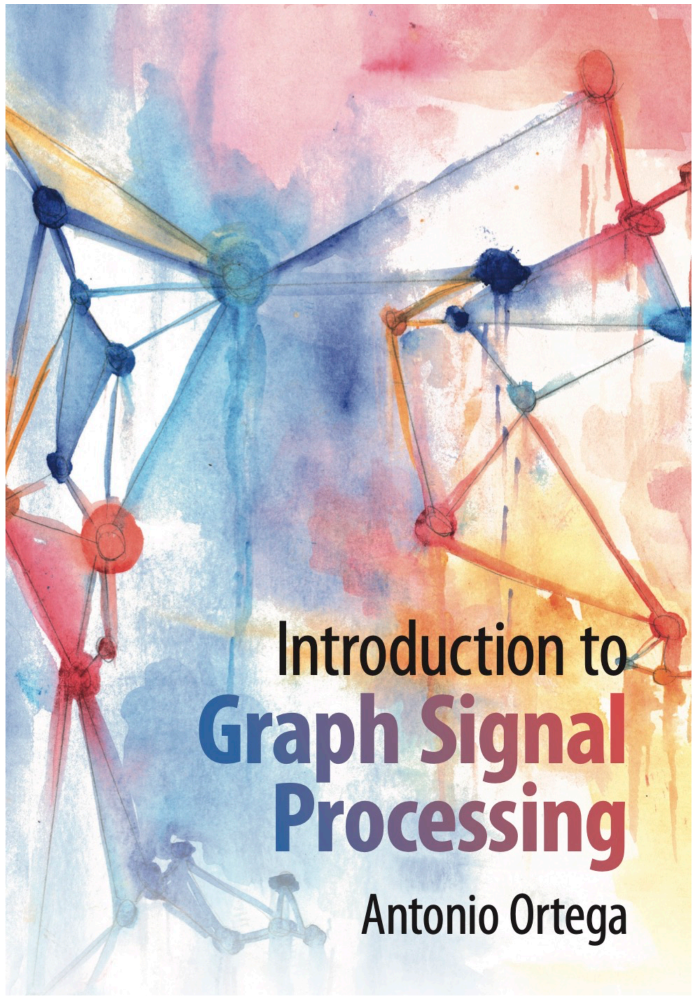
- Mathematical models for graph signals
  - global and local smoothness / regularity
  - incorporating underlying physical processes
- Graph construction
  - how to infer graph topology given observed data?
  - how to handle temporal dynamics?
- Fast implementation
  - fast graph Fourier transform
  - distributed processing
- Connection with other fields
  - machine learning on graphs
  - complex networks and systems
- Applications

References

- Four tutorial/overview papers and one textbook







Resources

- Graph signal processing
  - MATLAB toolbox
    - https://github.com/epfl-lts2/gspbox
    - https://github.com/STAC-USC/GraSP
  - Python toolbox
    - https://github.com/epfl-lts2/pygsp

- More at: http://www.robots.ox.ac.uk/~xdong/resource.html