

Graph signal processing

Concepts, tools and applications in neuroscience

Xiaowen Dong

Oxford-Man Institute

Department of Engineering Science

NISOx, Big Data Institute, February 2019



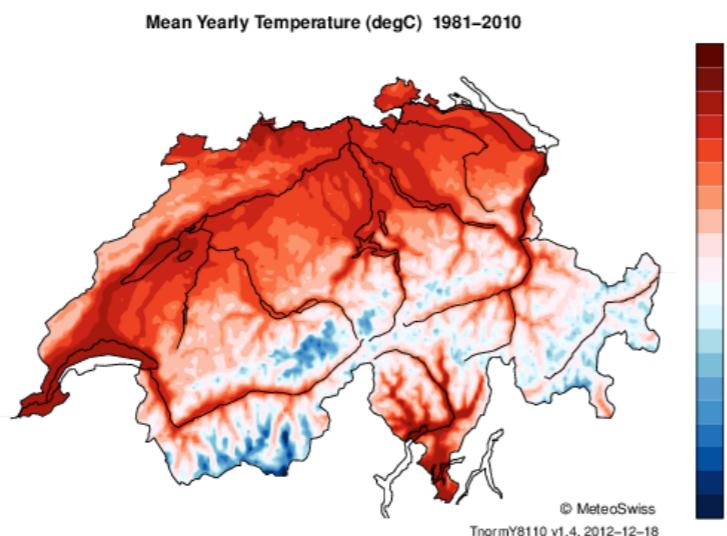
Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

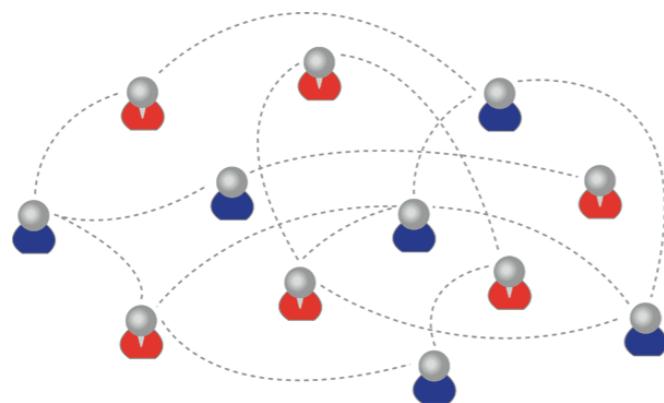
Data are often structured



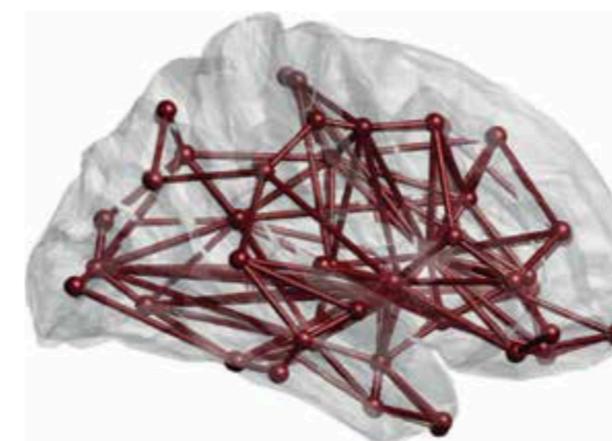
Temperature data



Traffic data

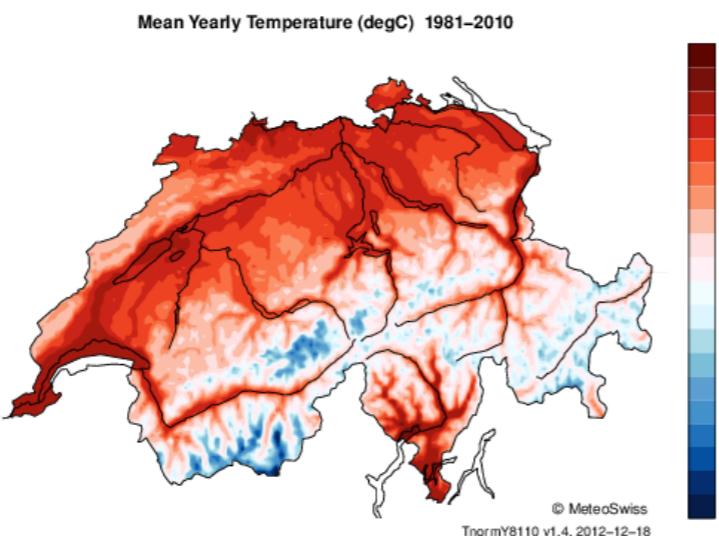


Social network data



Neuroimaging data

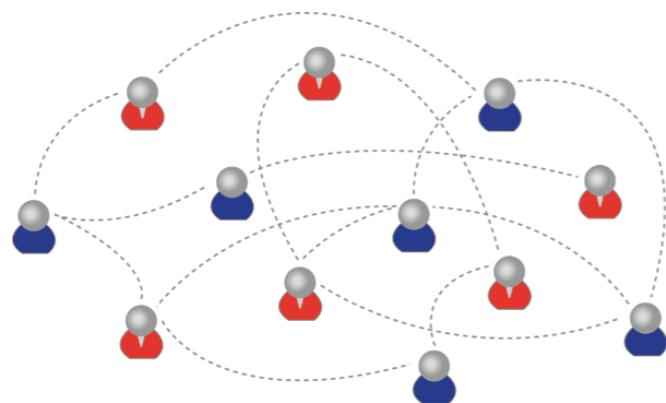
Data are often structured



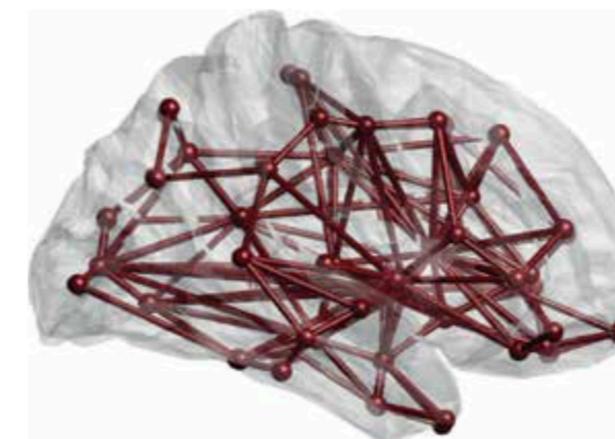
Temperature data



Traffic data



Social network data

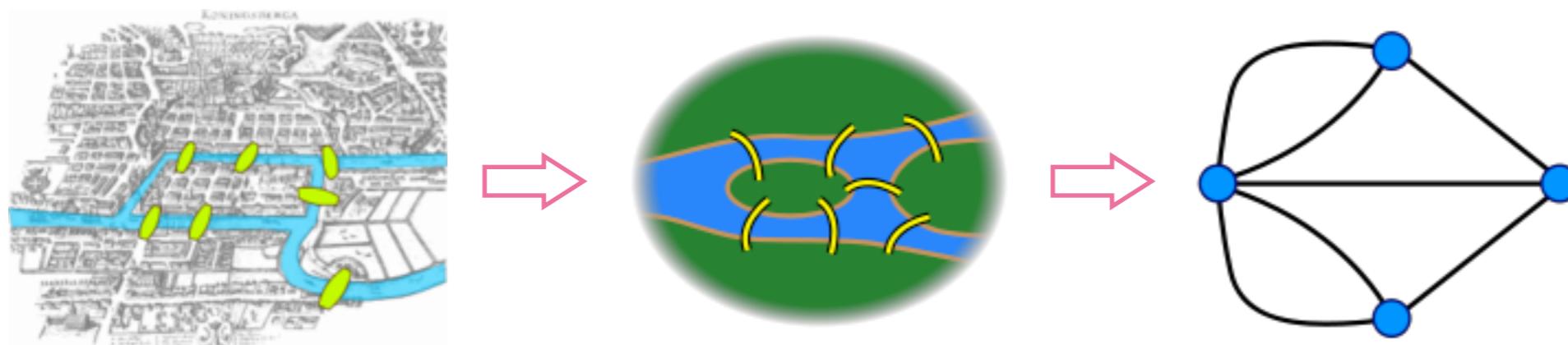


Neuroimaging data

We need to take into account the structure behind the data

Graphs are appealing tools

- Efficient representations for **pairwise relations** between entities

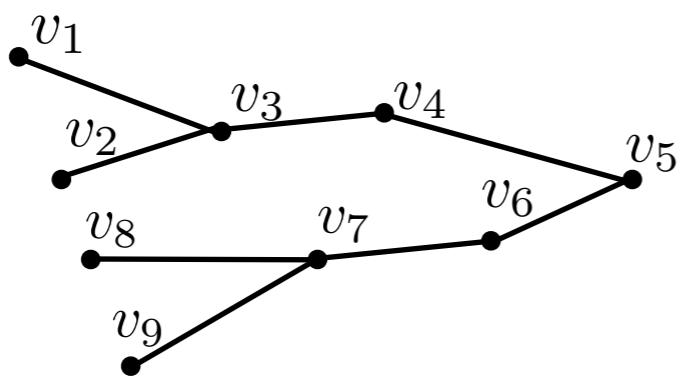


The Königsberg Bridge Problem
[Leonhard Euler, 1736]



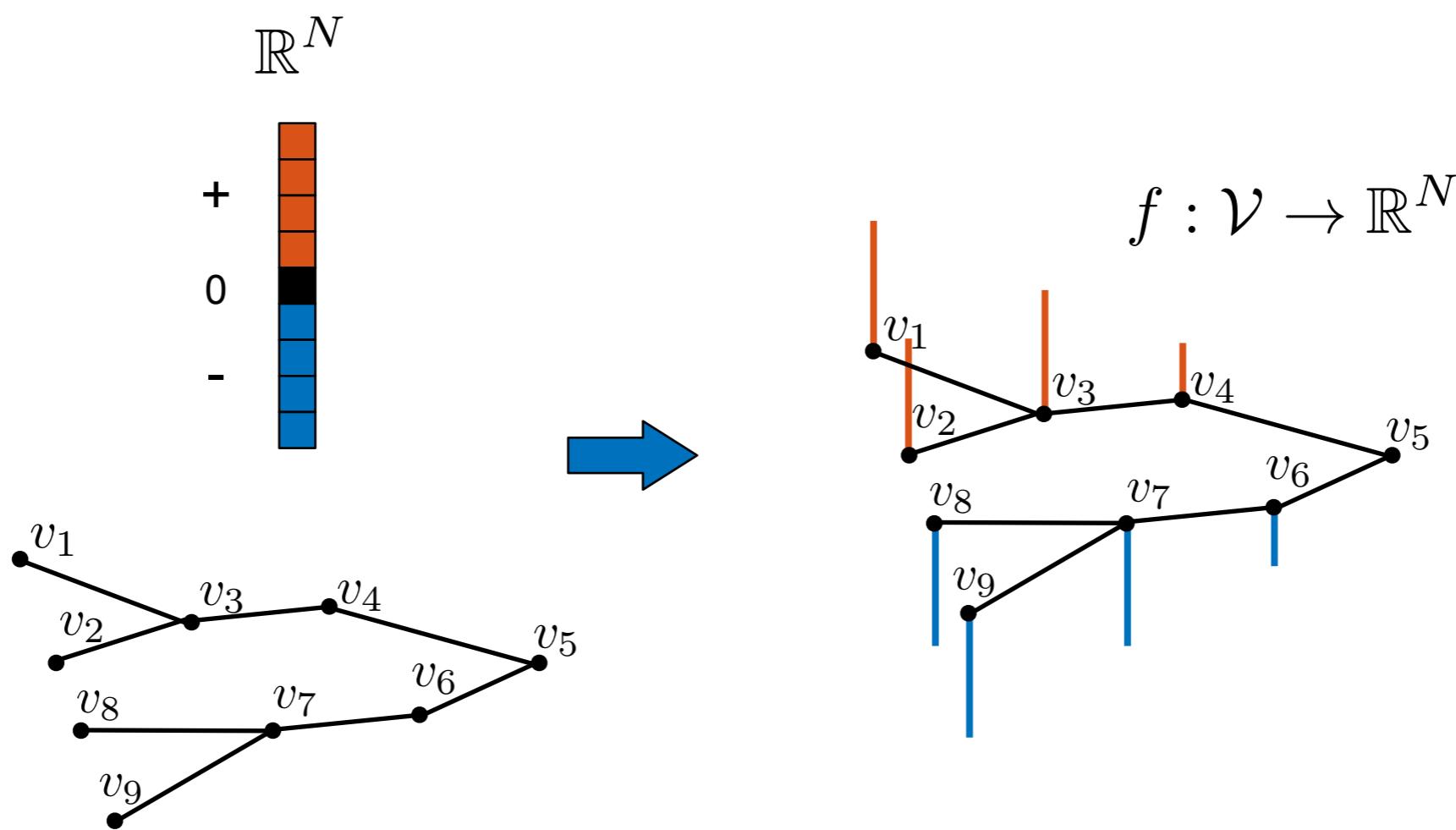
Graphs are appealing tools

- Efficient representations for pairwise relations between entities



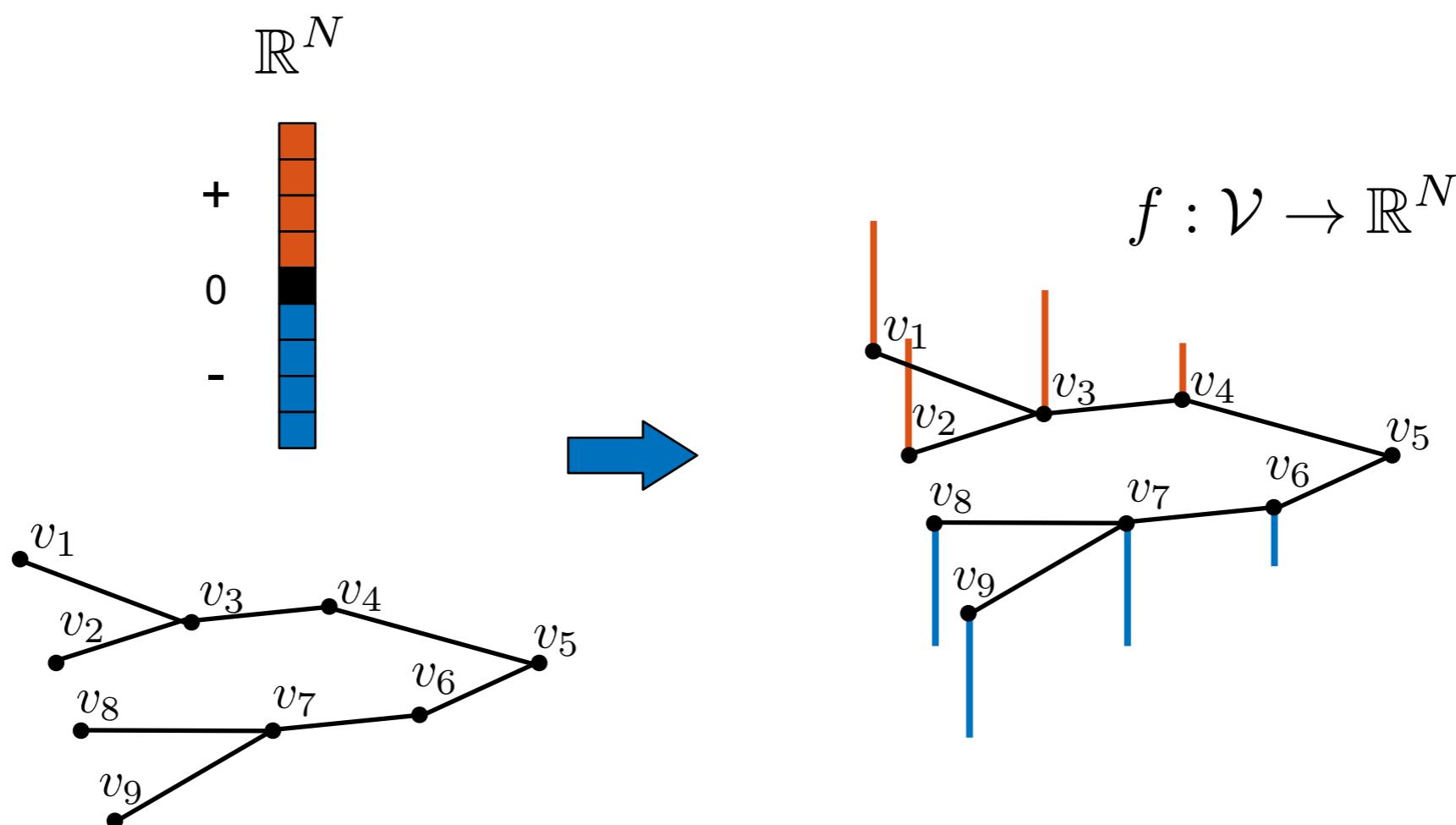
Graphs are appealing tools

- Efficient representations for pairwise relations between entities
- Structured data can be represented by **graph signals**



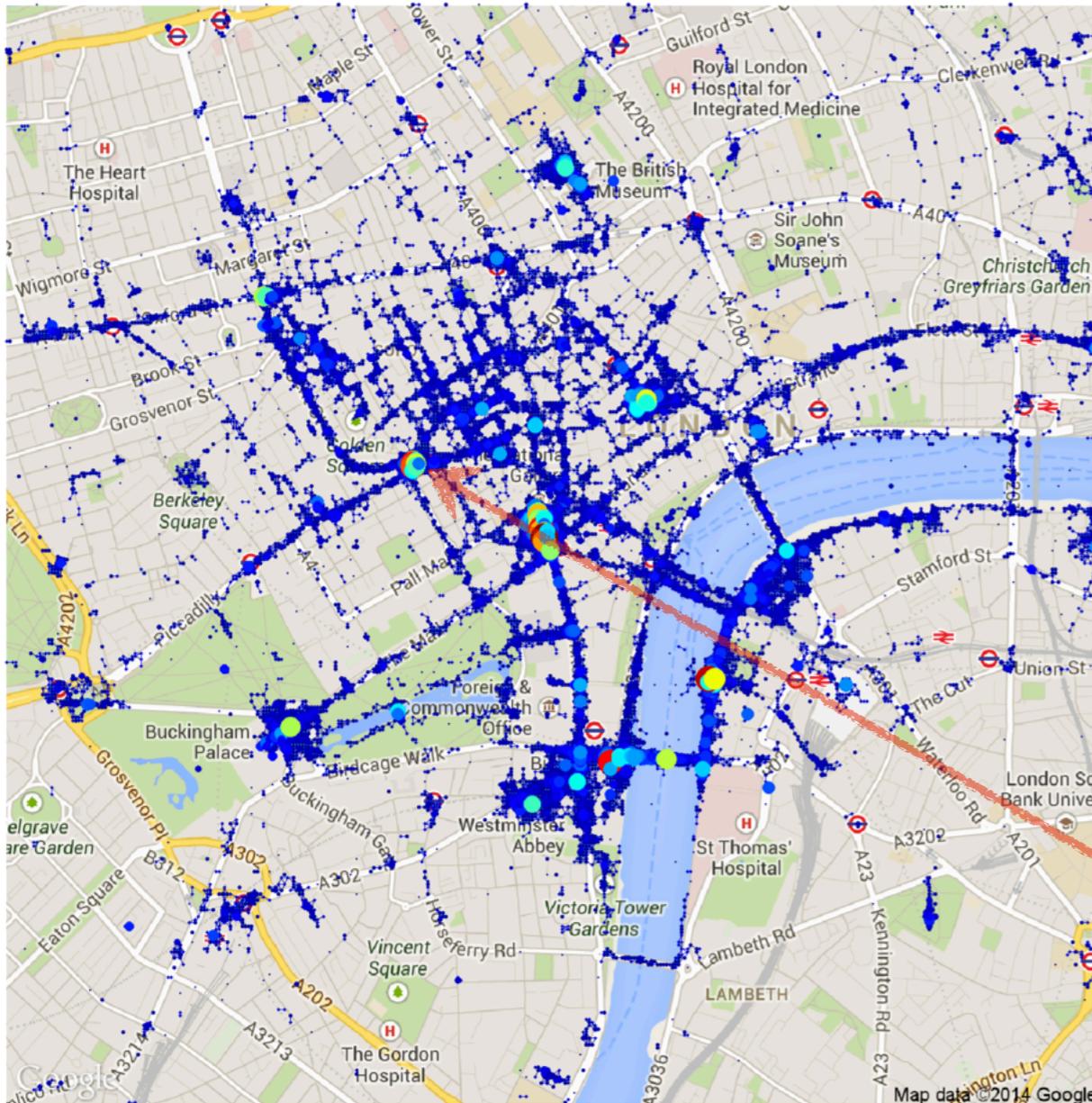
Graphs are appealing tools

- Efficient representations for pairwise relations between entities
- Structured data can be represented by **graph signals**

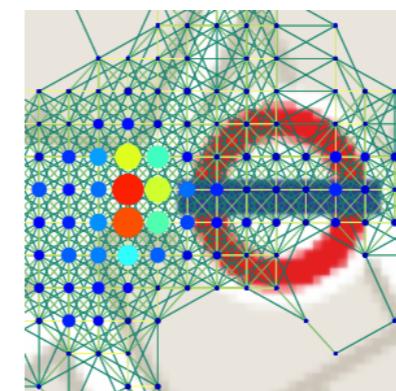


Takes into account both structure (edges) and data (values at vertices)

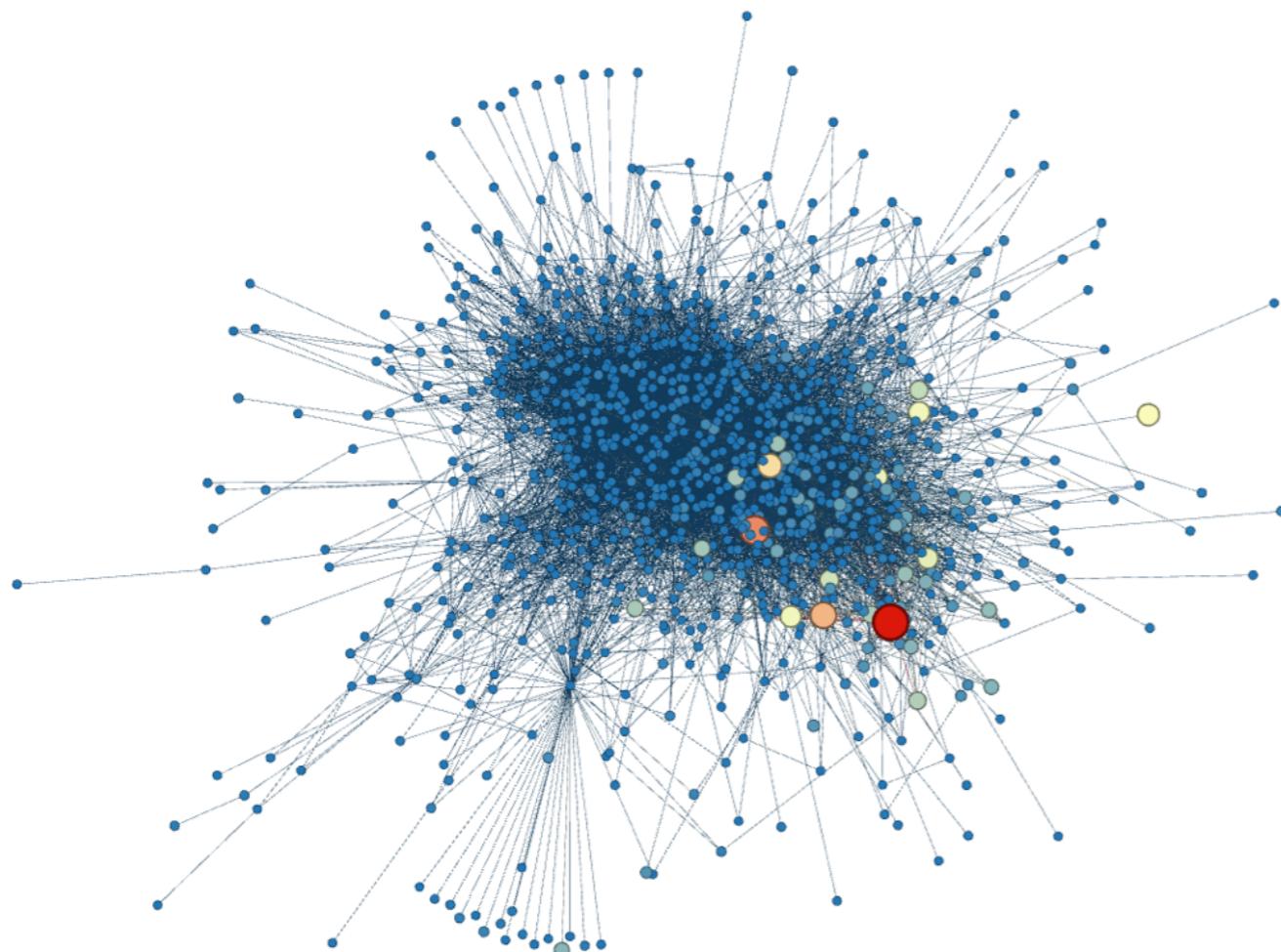
Graph signals are pervasive



- Vertices:
 - 9000 grid cells in London
- Edges:
 - geographical proximity of grid cells
- Signal:
 - # Flickr users who have taken photos in two and a half year

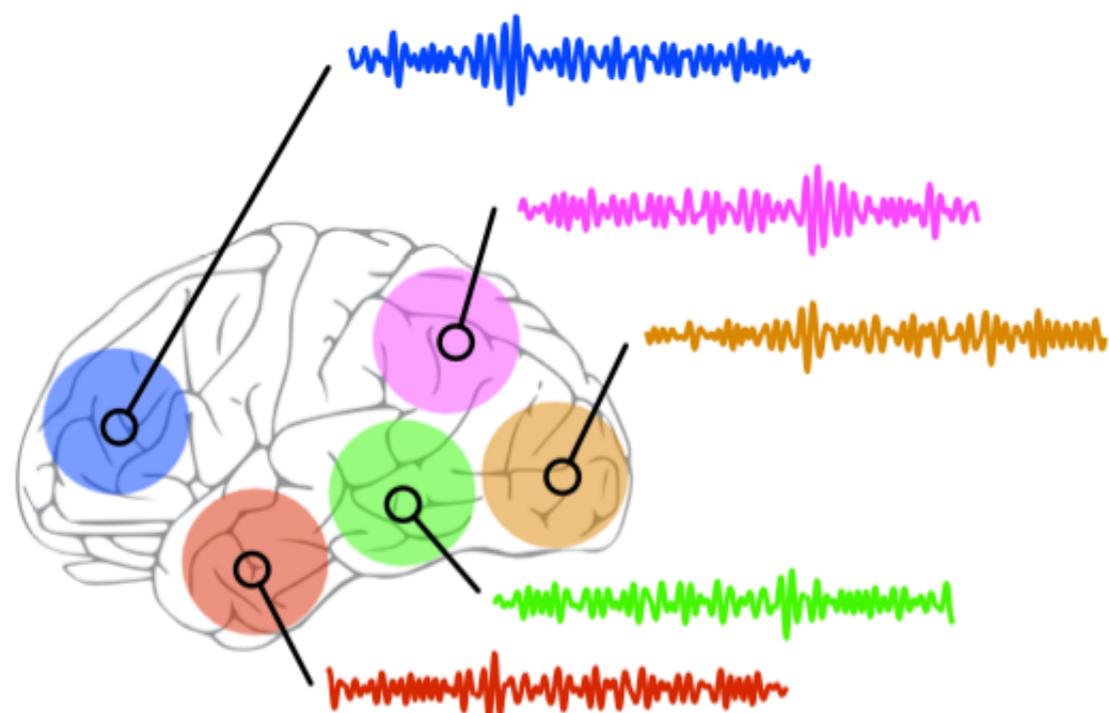


Graph signals are pervasive



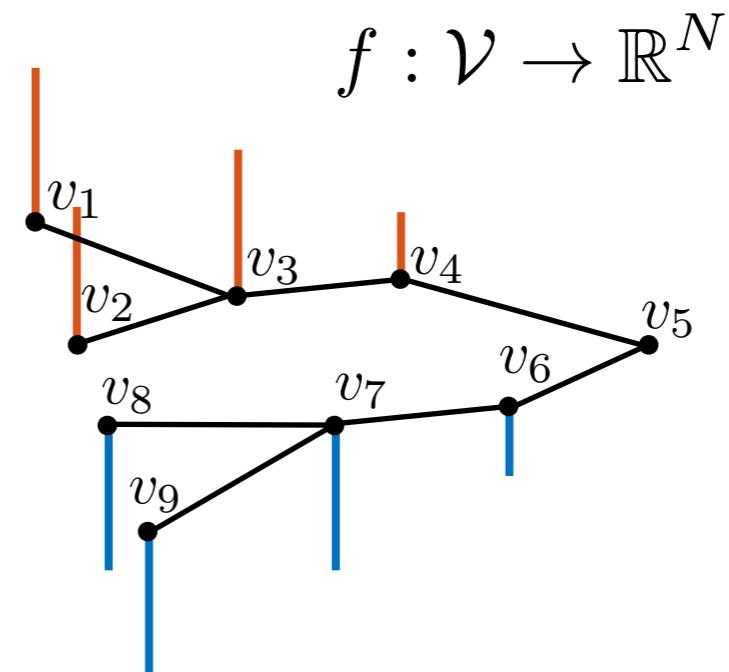
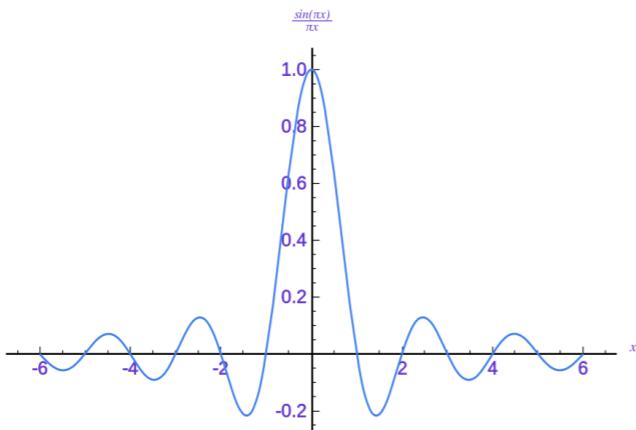
- Vertices:
 - 1000 Twitter users
- Edges:
 - following relationship among users
- Signal:
 - # Apple-related hashtags they have posted in six weeks

Graph signals are pervasive



- Vertices:
 - brain regions
- Edges:
 - structural connectivity between brain regions
- Signal:
 - blood-oxygen-level-dependent (BOLD) time series

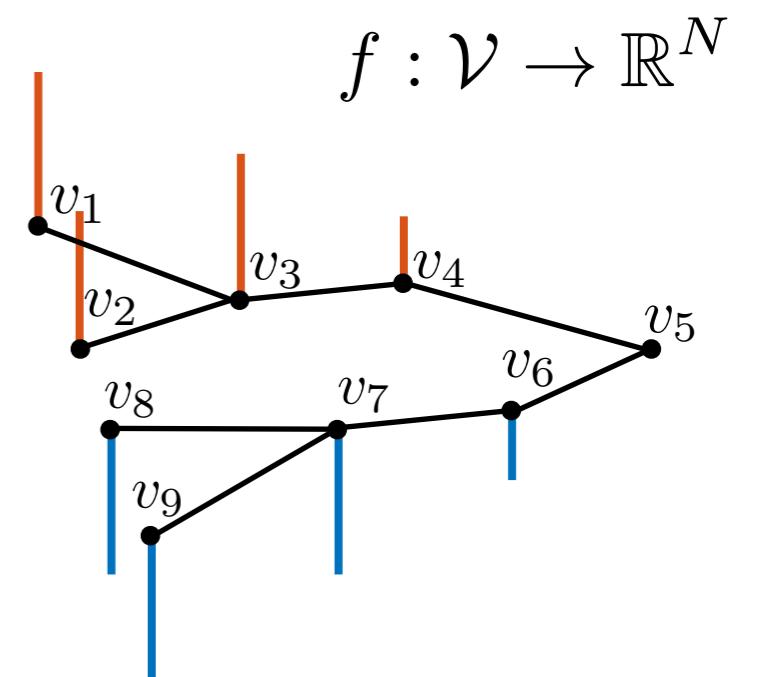
Research challenges



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

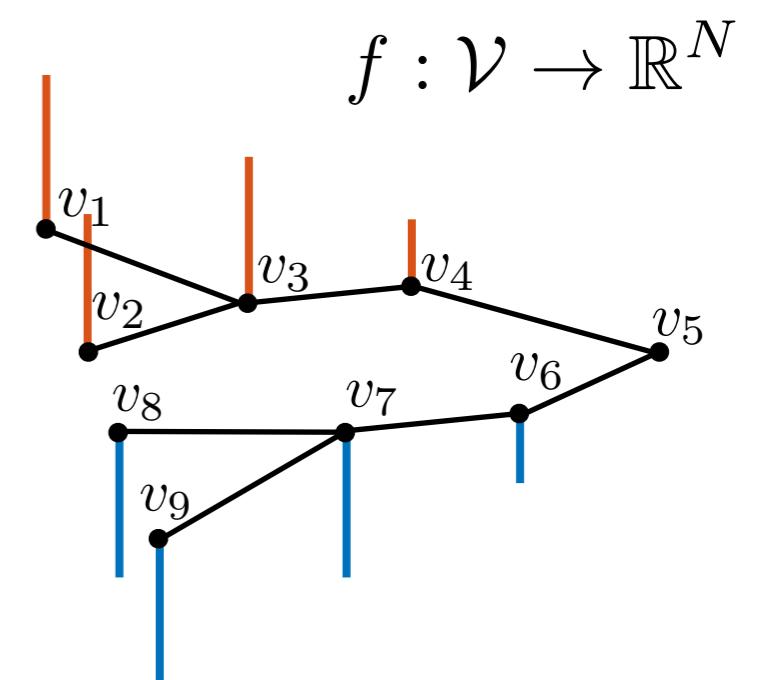
Graph signal processing

- 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 network analysis, etc.



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 network analysis, etc.
- An increasingly rich literature
 - classical signal processing
 - algebraic and spectral graph theory
 - computational harmonic analysis
 - machine learning

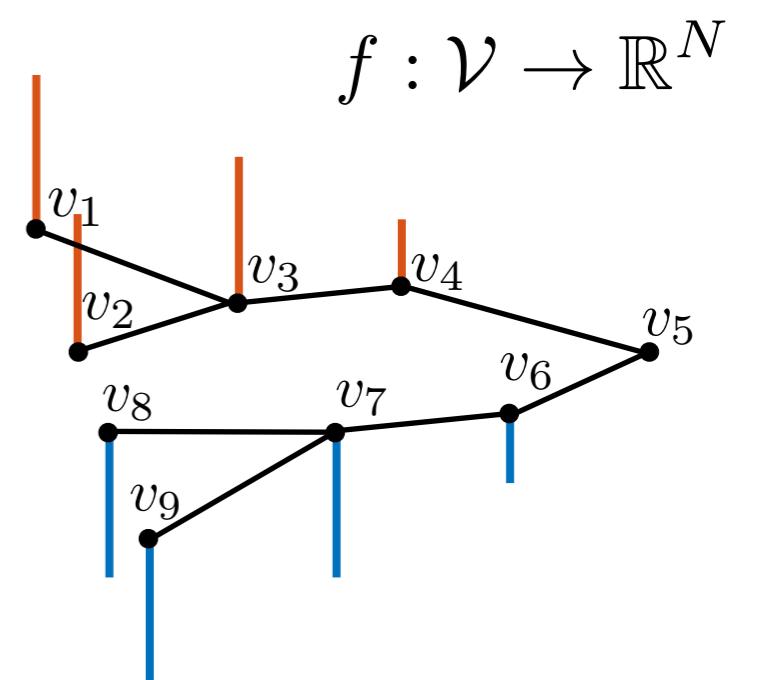


Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

Two paradigms

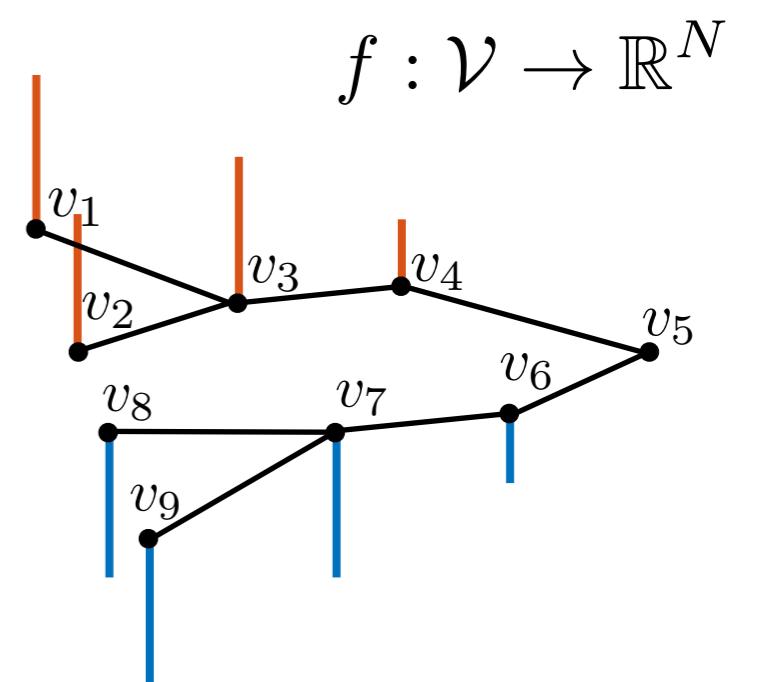
- The main approaches can be categorised into two families:
 - vertex (spatial) domain designs
 - frequency (graph spectral) domain designs



Two paradigms

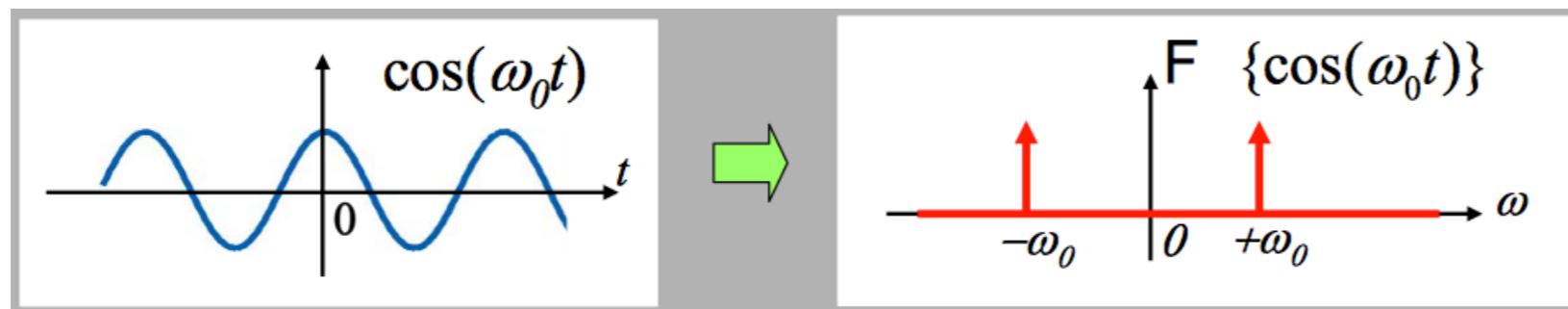
- The main approaches can be categorised into two families:
 - vertex (spatial) domain designs
 - frequency (graph spectral) domain designs

Important for analysis of signal properties

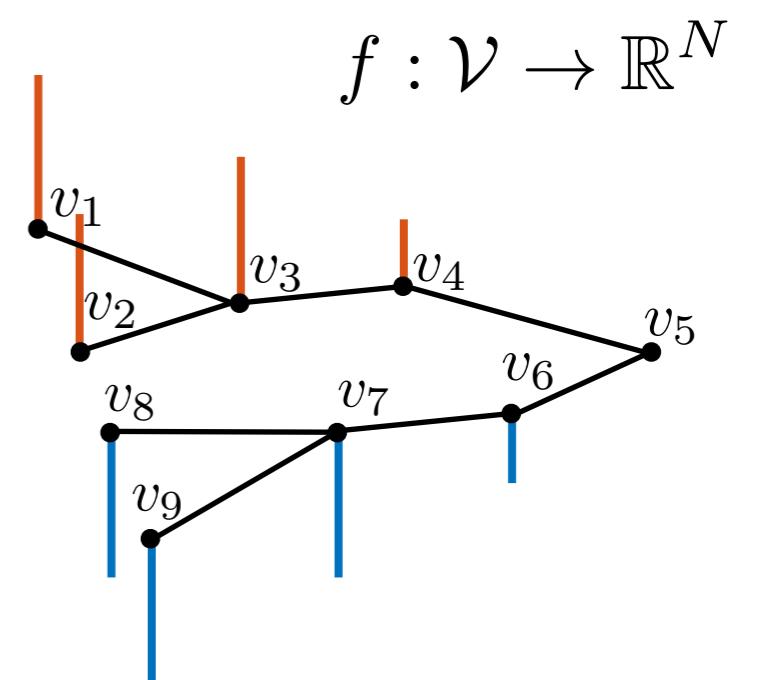


Need for frequency

- Classical Fourier transform provides the frequency domain representation of the signals

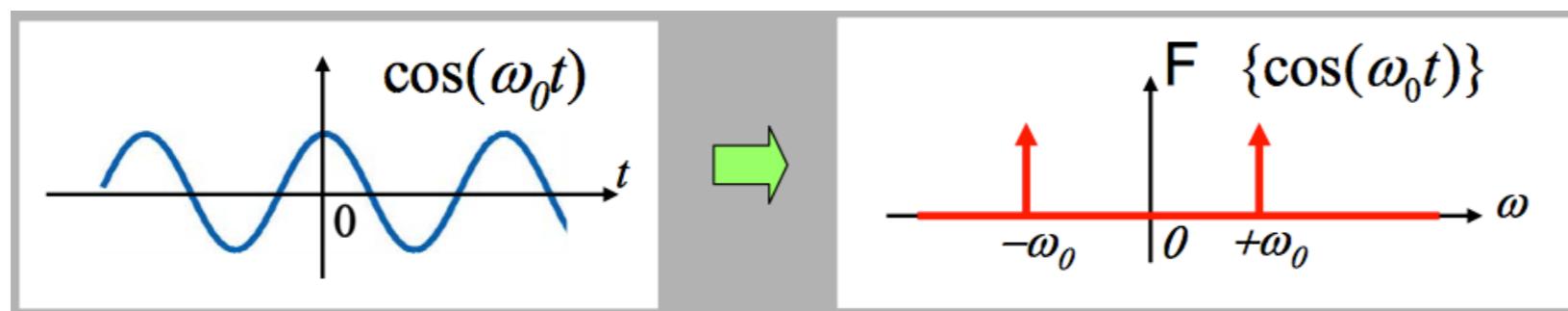


Source: <http://www.physik.uni-kl.de>



Need for frequency

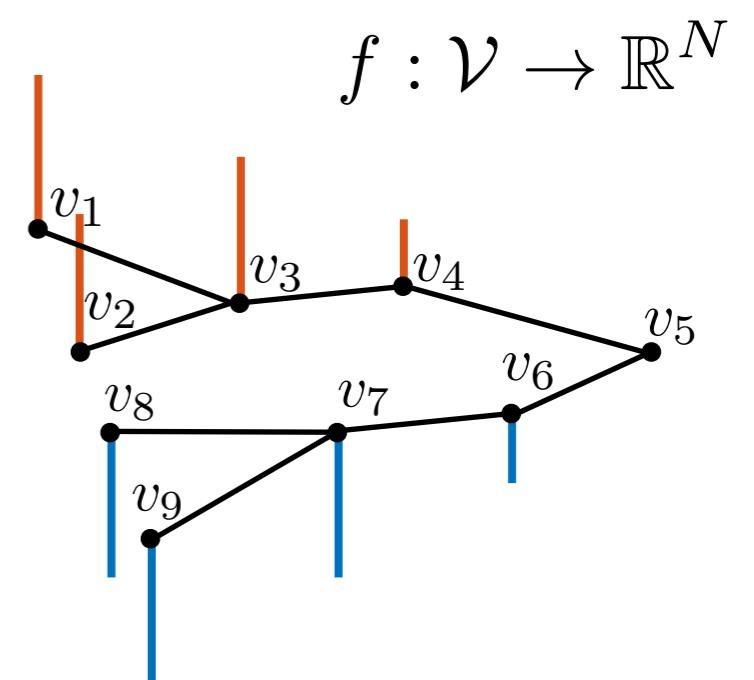
- Classical Fourier transform provides the frequency domain representation of the signals



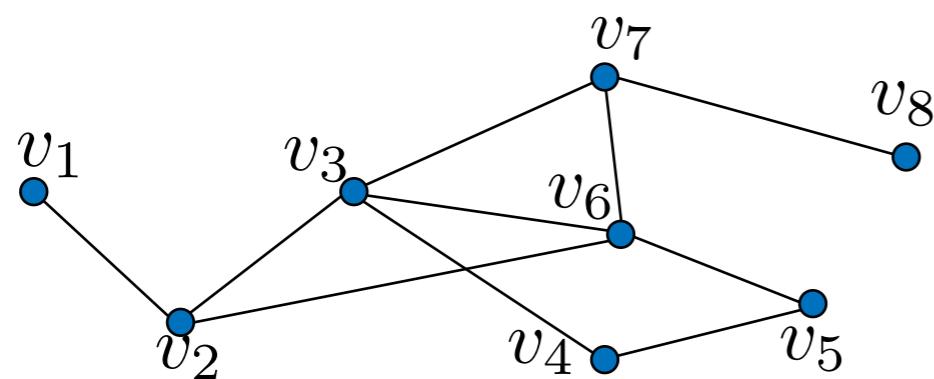
Source: <http://www.physik.uni-kl.de>

A notion of frequency for graph signals:

We need the graph Laplacian matrix



Graph Laplacian



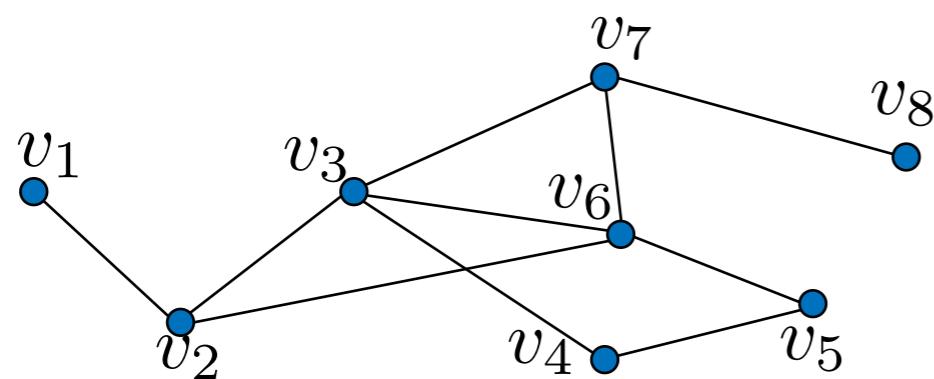
Weighted and undirected graph:

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

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

$$W$$

Graph Laplacian



Weighted and undirected graph:

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

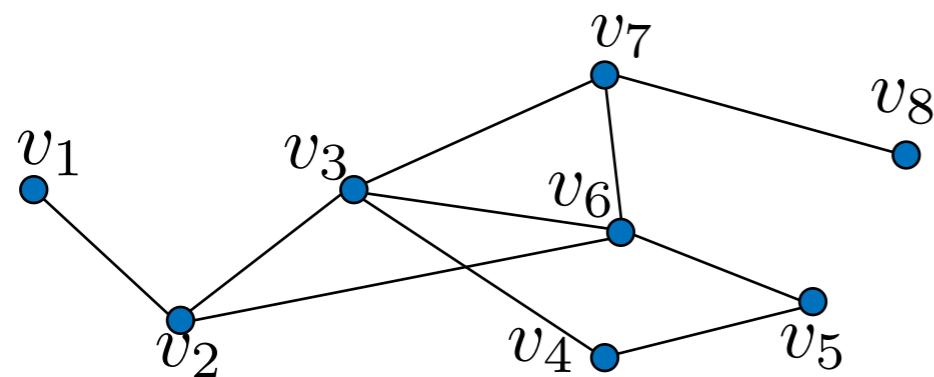
$$D = \text{diag}(d(v_1), \dots, d(v_N))$$

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

$$D$$

$$W$$

Graph Laplacian



Weighted and undirected graph:

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

$$D = \text{diag}(d(v_1), \dots, d(v_N))$$

$$L = D - W \quad \text{Equivalent to G!}$$

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

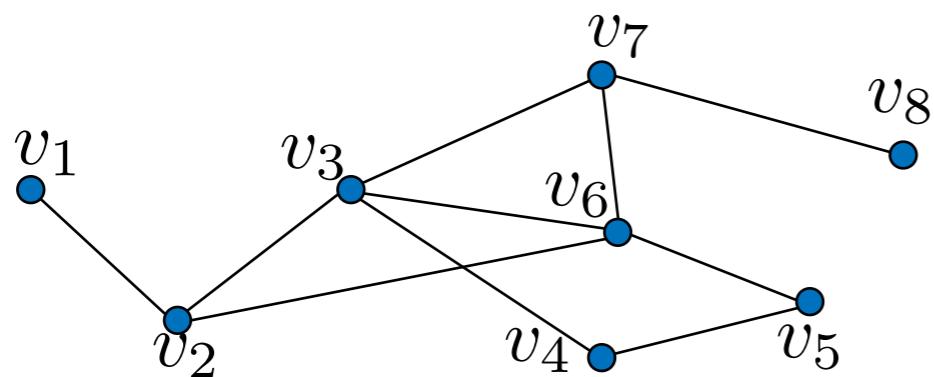
$$D$$

$$W$$

$$L$$

- Symmetric
- Off-diagonal entries non-positive
- Rows sum up to zero

Graph Laplacian



Weighted and undirected graph:

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

$$D = \text{diag}(d(v_1), \dots, d(v_N))$$

$$L = D - W \quad \text{Equivalent to G!}$$

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

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

$$D$$

$$W$$

$$L$$

- Symmetric
- Off-diagonal entries non-positive
- Rows sum up to zero

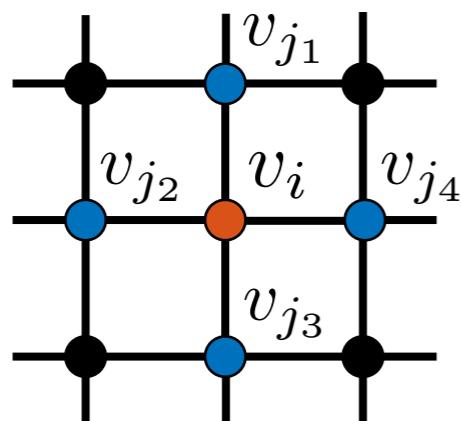
Graph Laplacian

Why graph Laplacian?

Graph Laplacian

Why graph Laplacian?

- approximation of the Laplace operator



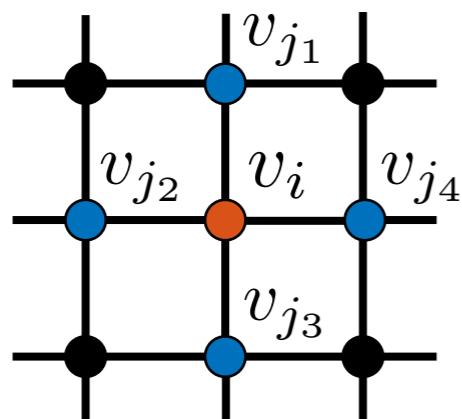
$$(Lf)(i) = 4f(i) - [f(j_1) + f(j_2) + f(j_3) + f(j_4)]$$

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

Graph Laplacian

Why graph Laplacian?

- approximation of the Laplace operator

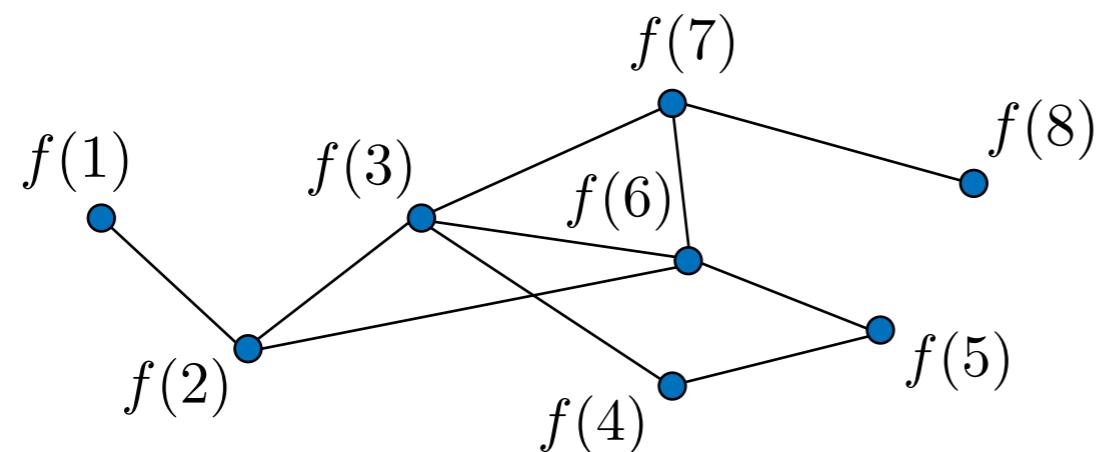


$$(Lf)(i) = 4f(i) - [f(j_1) + f(j_2) + f(j_3) + f(j_4)]$$

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

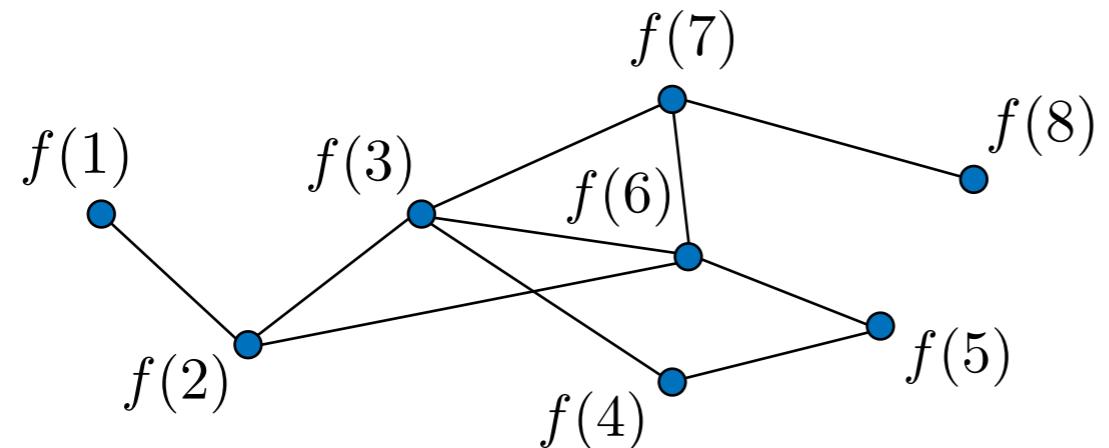
- converges to the Laplace-Beltrami operator (given certain conditions)
- provides a notion of “frequency” on graphs

Graph Laplacian



Graph signal $f : \mathcal{V} \rightarrow \mathbb{R}^N$

Graph Laplacian

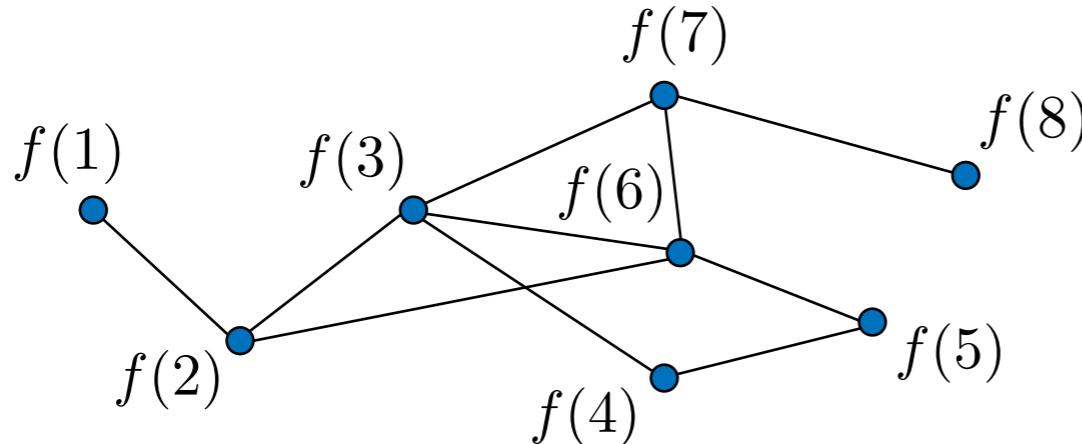


Graph signal $f : \mathcal{V} \rightarrow \mathbb{R}^N$

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

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

Graph Laplacian



Graph signal $f : \mathcal{V} \rightarrow \mathbb{R}^N$

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

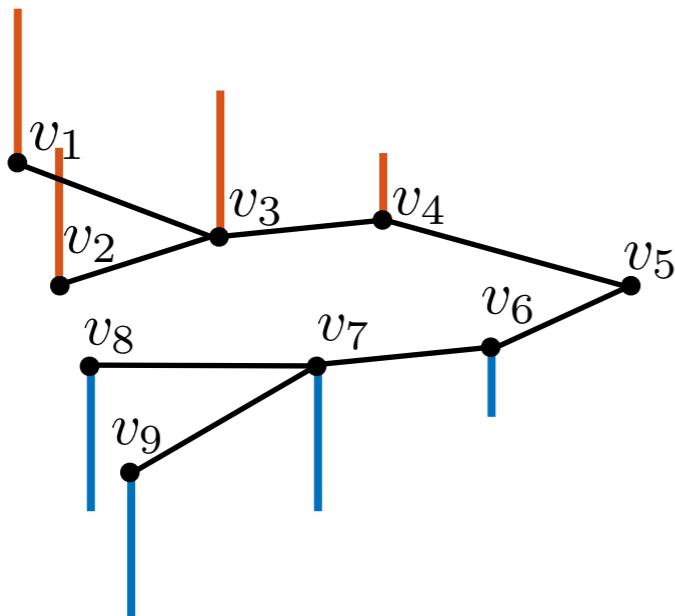
$$\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}^T \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 &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

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

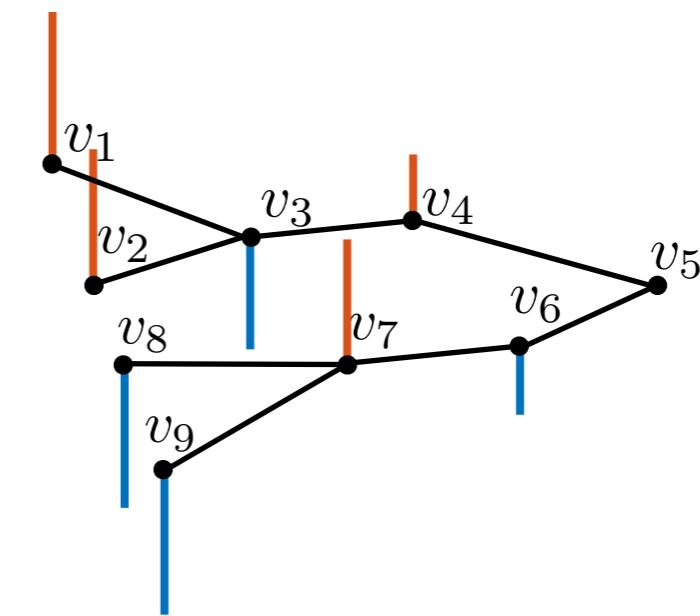
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^N W_{ij} (f(i) - f(j))^2$$

A measure of “smoothness”

Graph Laplacian



$$f^T L f = 1$$



$$f^T L f = 21$$

Graph Laplacian

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

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

Graph Laplacian

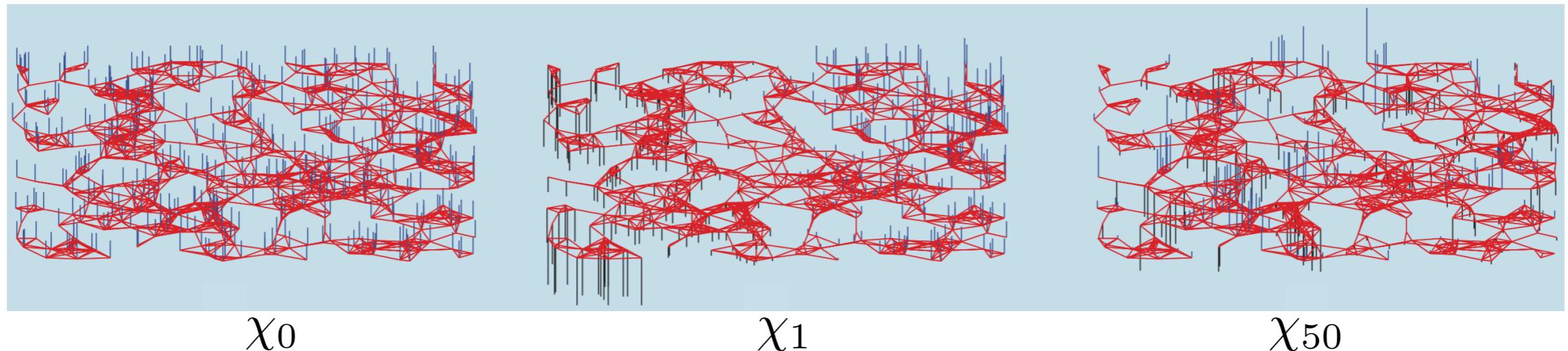
- L has a complete set of orthonormal eigenvectors: $L = \chi \Lambda \chi^T$

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

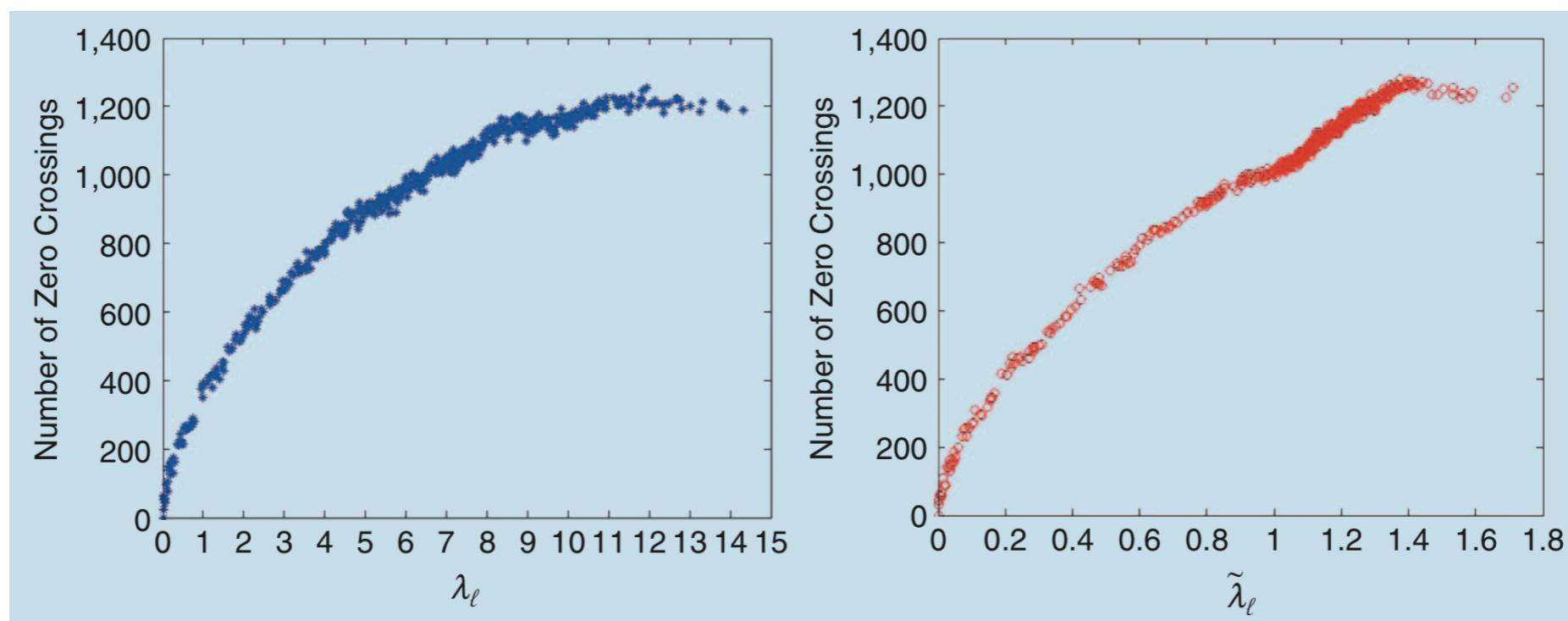
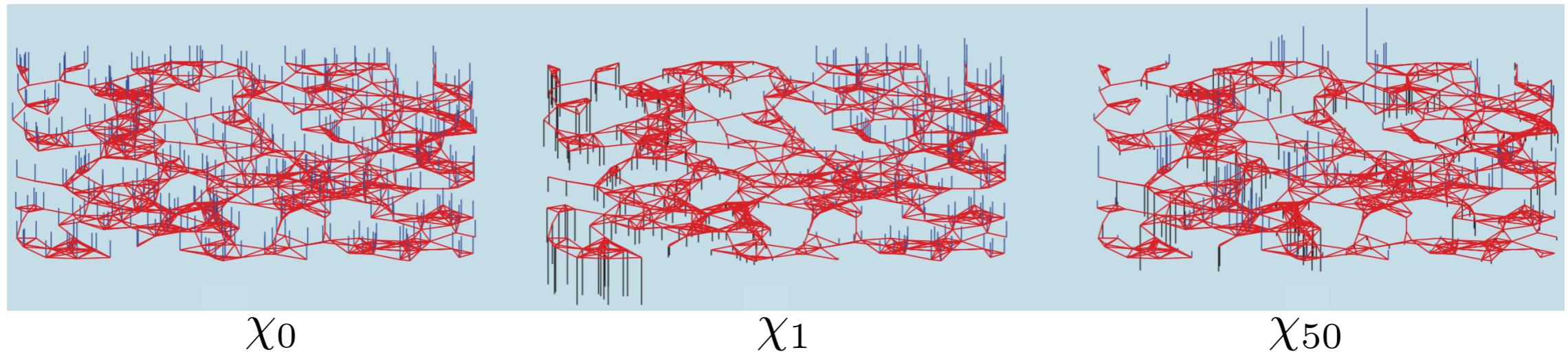
$\chi \quad \Lambda \quad \chi^T$

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

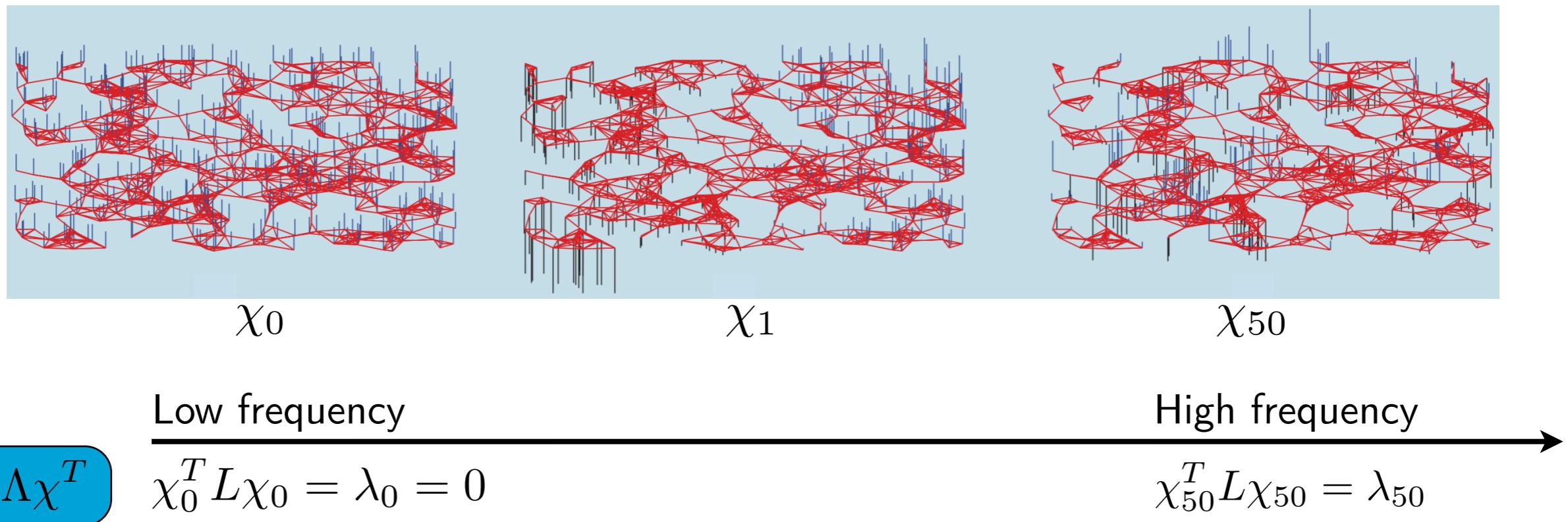
Graph Fourier transform



Graph Fourier transform

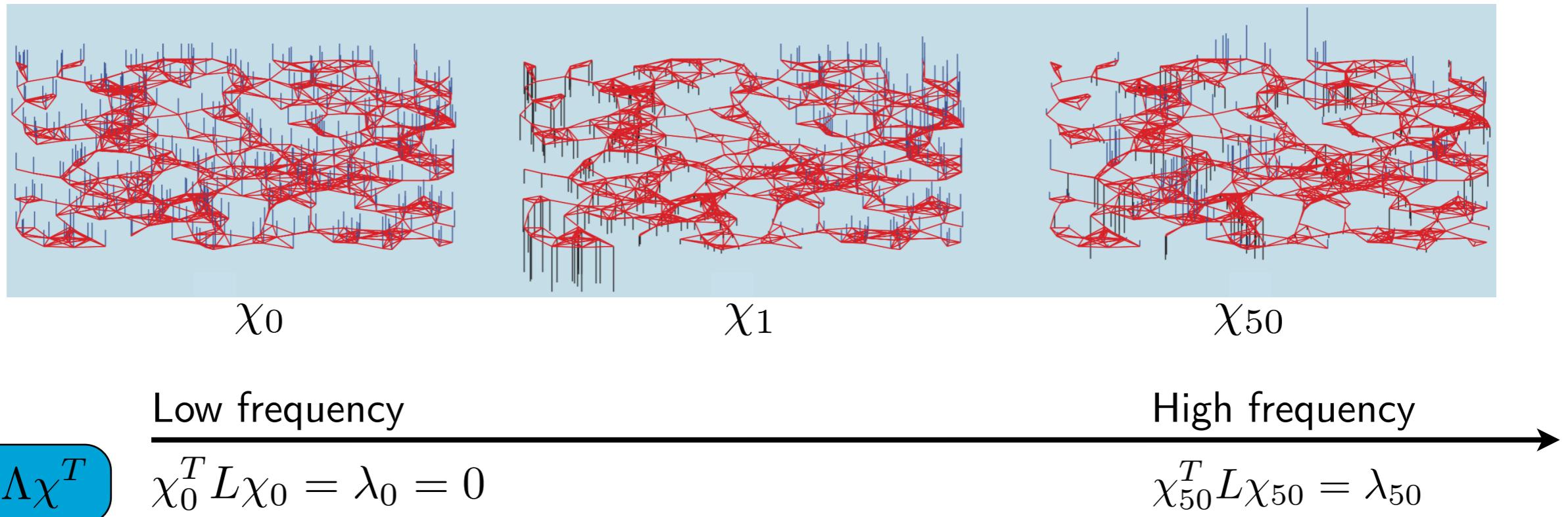


Graph Fourier transform



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

Graph Fourier transform

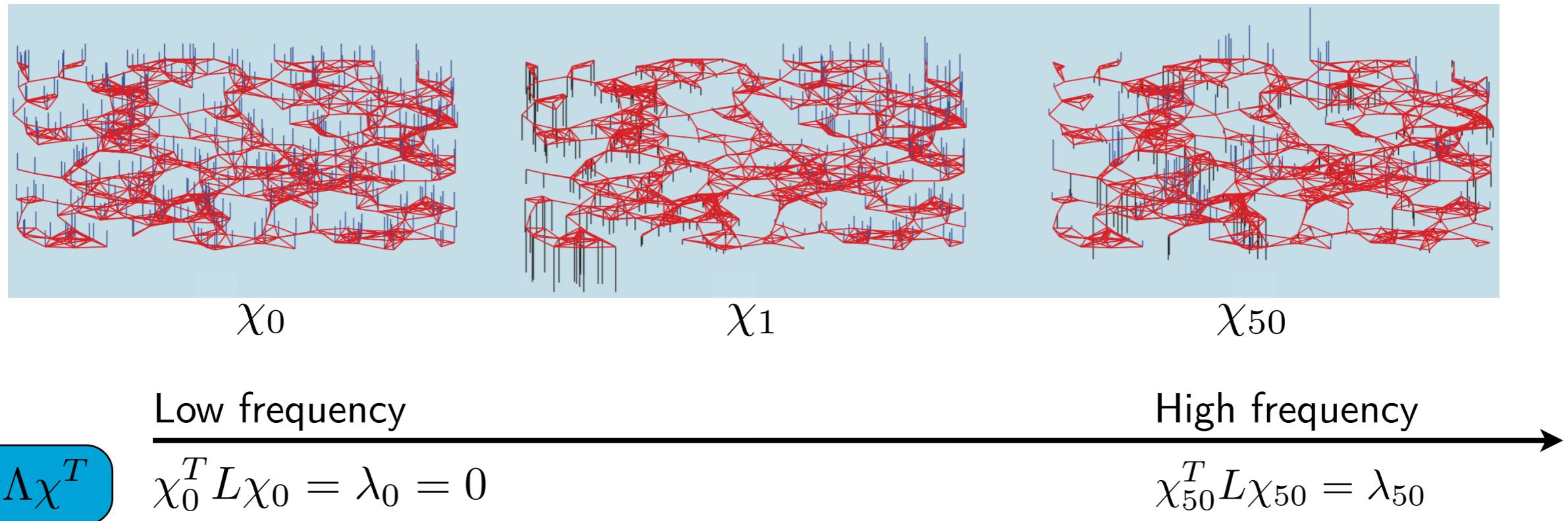


Graph Fourier transform:

[Hammond11]

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

Graph Fourier transform



Graph Fourier transform: [Hammond11]

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: $-\nabla^2$



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 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



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: $-\nabla^2$



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



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

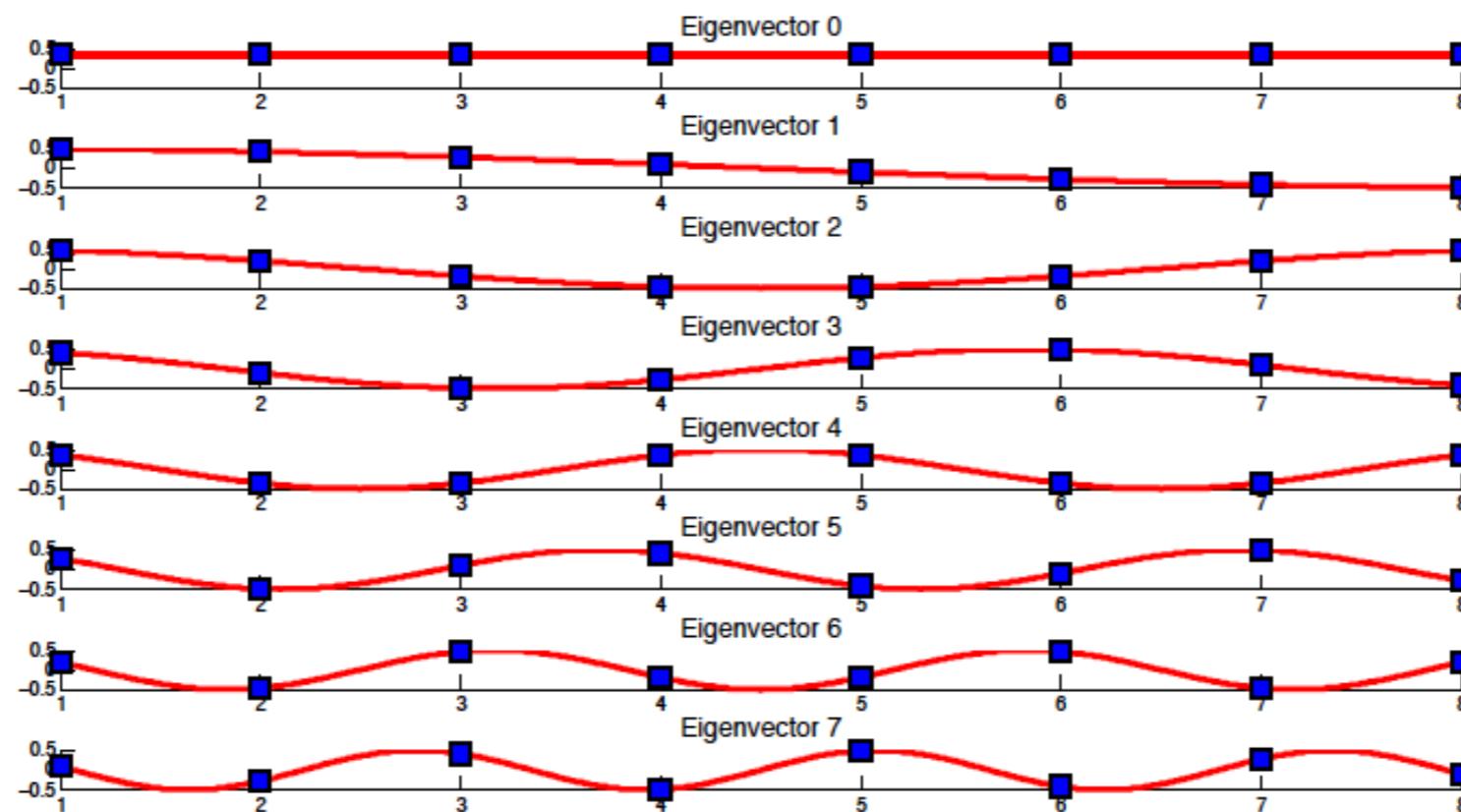


- (Unordered) Laplacian eigenvalues: $\lambda_\ell = 2 - 2 \cos\left(\frac{2\ell\pi}{N}\right)$
- One possible choice of orthogonal Laplacian eigenvectors:
$$\chi_\ell = [1, \omega^\ell, \omega^{2\ell}, \dots, \omega^{(N-1)\ell}]$$
, where $\omega = e^{\frac{2\pi j}{N}}$
- $\begin{bmatrix} & | & & | \\ & x_0 & \dots & x_{N-1} \\ & | & & | \end{bmatrix}$ is the Discrete Fourier Transform (DFT) matrix

Two special cases



$$\begin{array}{ll} \text{if } \lambda_\ell = 2 - 2 \cos \left(\frac{\pi \ell}{N} \right) & \text{if } \chi_0(i) = \frac{1}{\sqrt{N}}, \quad \chi_\ell(i) = \sqrt{\frac{2}{N}} \cos \left(\frac{\pi \ell(i-0.5)}{N} \right), \quad \ell = 1, 2, \dots, N-1 \end{array}$$


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

Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

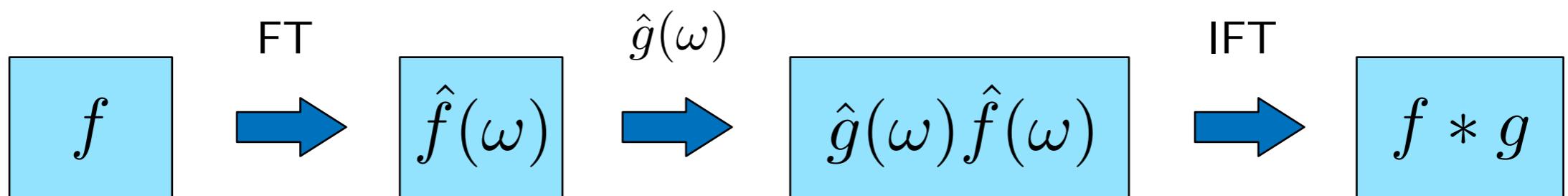
Classical frequency filtering

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

Classical frequency filtering

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

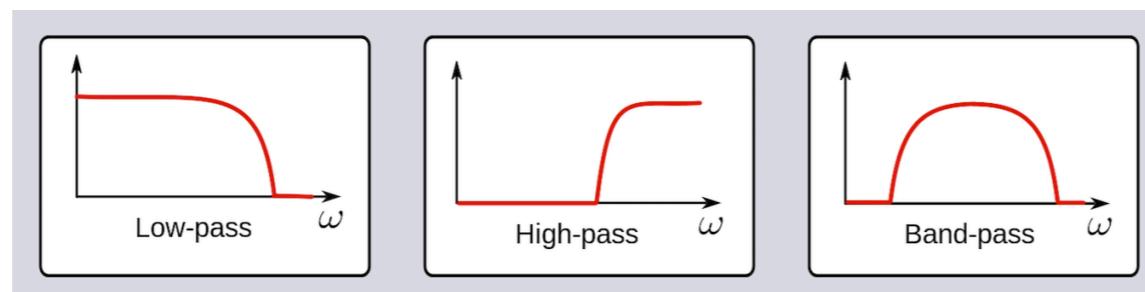
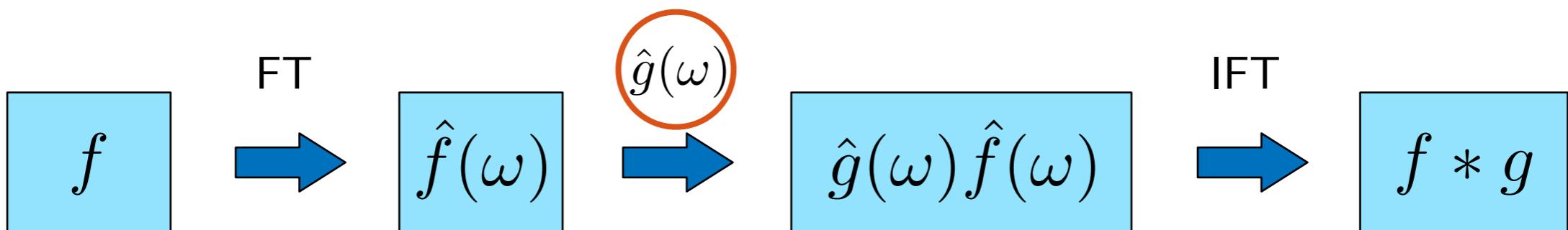
Apply filter with transfer function $\hat{g}(\cdot)$ to a signal f



Classical frequency filtering

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

Apply filter with transfer function $\hat{g}(\cdot)$ to a signal f



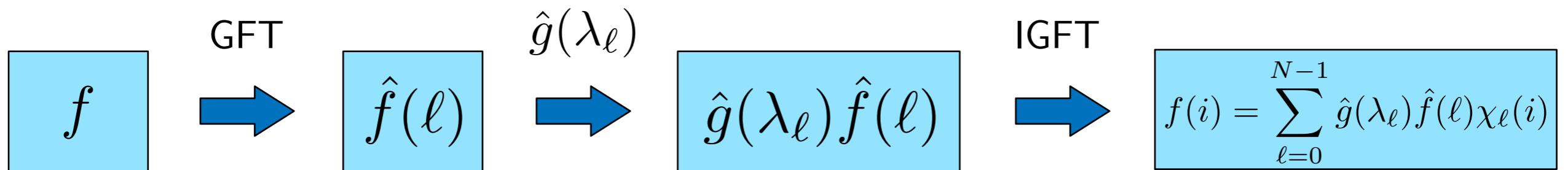
Graph spectral filtering

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

Graph spectral filtering

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

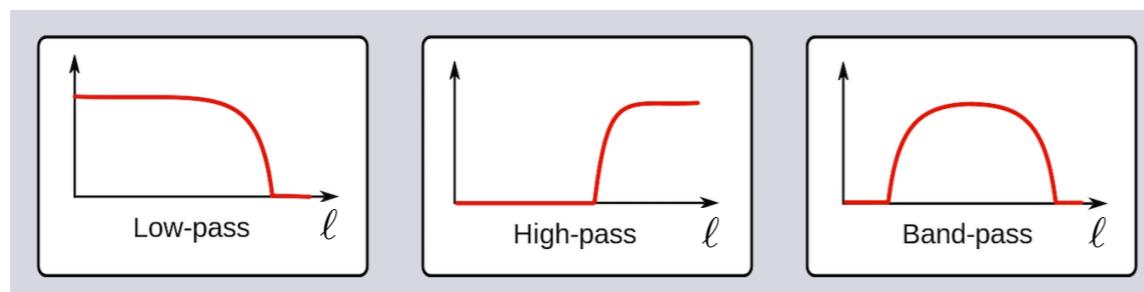
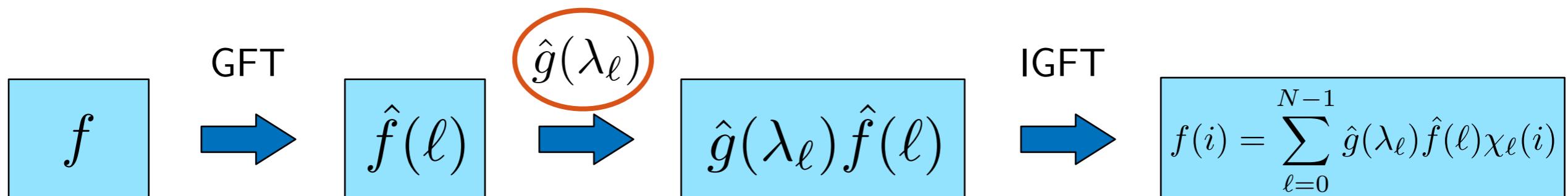
Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : \mathcal{V} \rightarrow \mathbb{R}^N$



Graph spectral filtering

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

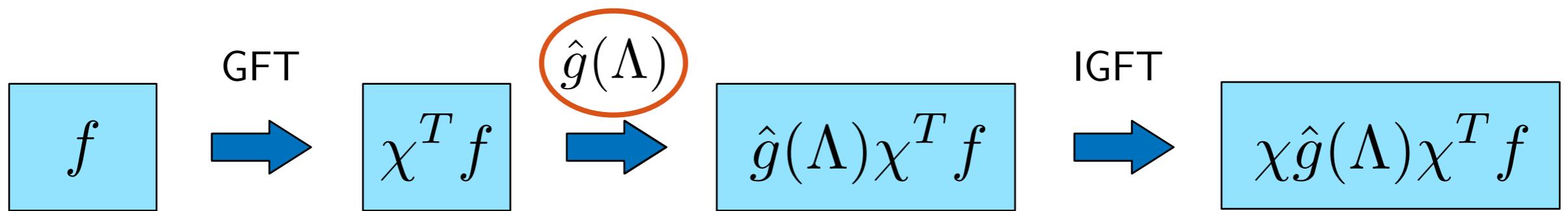
Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : \mathcal{V} \rightarrow \mathbb{R}^N$



Graph spectral filtering

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

Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : \mathcal{V} \rightarrow \mathbb{R}^N$

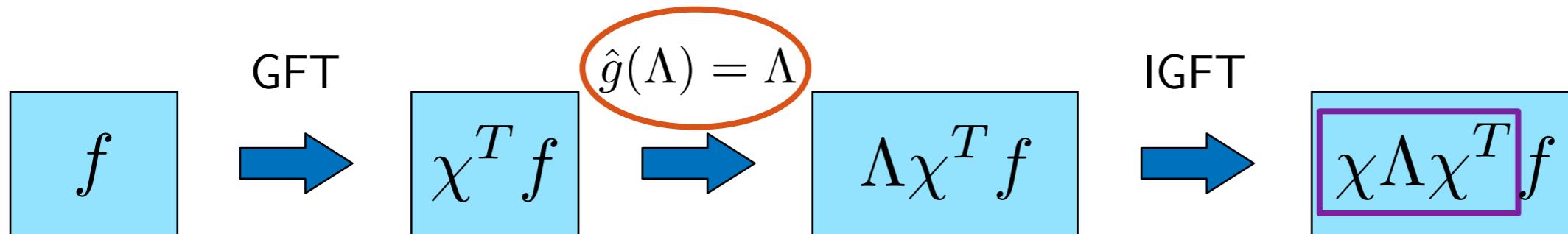


$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

Graph Laplacian revisited

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

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

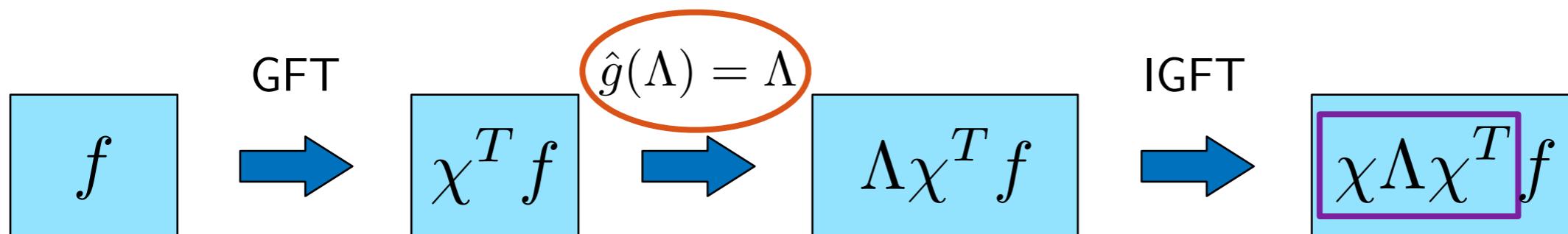


$$\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}$$

Graph Laplacian revisited

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

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



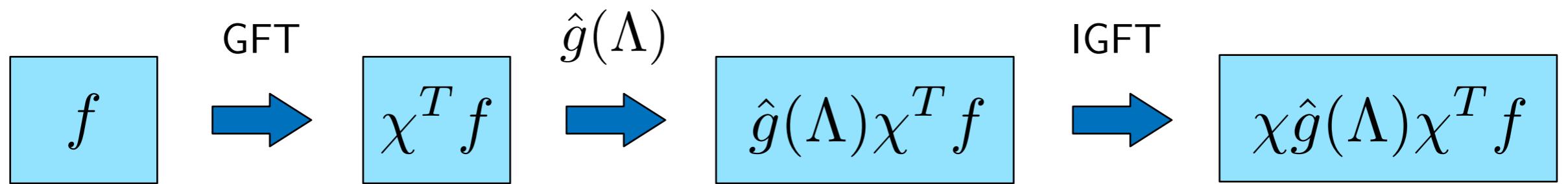
$$\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}$$

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

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

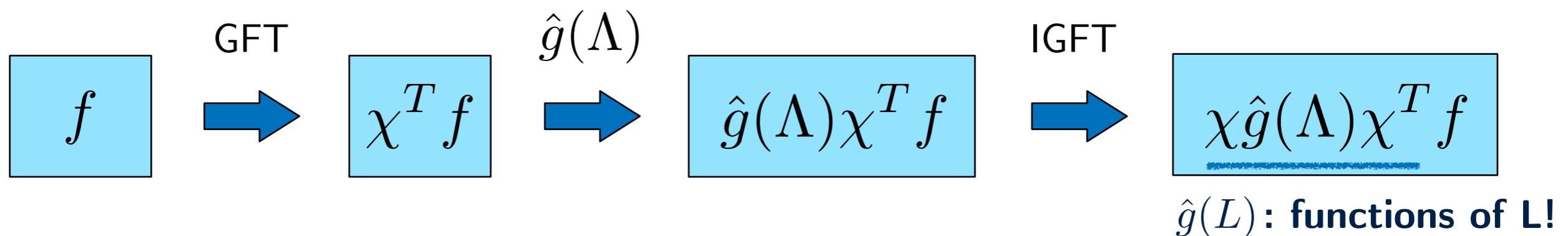
Graph transform/dictionary design

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



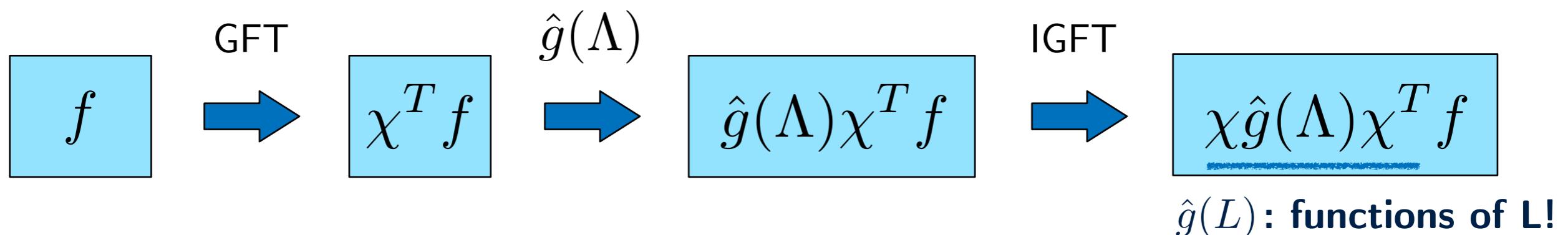
Graph transform/dictionary design

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



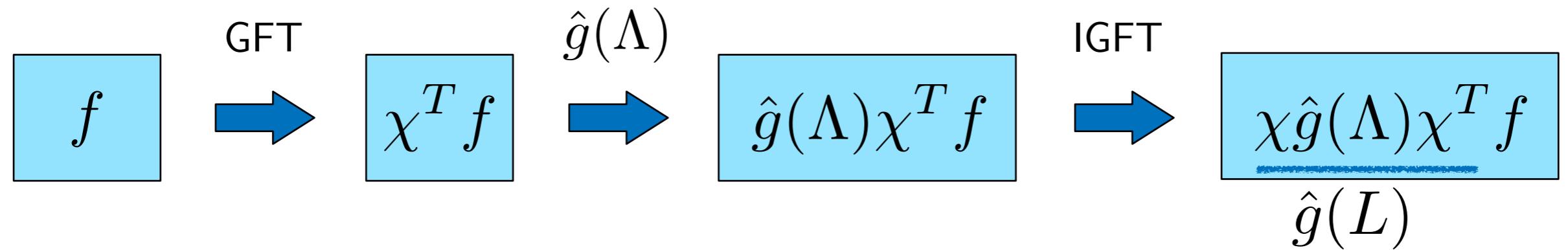
Graph transform/dictionary design

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

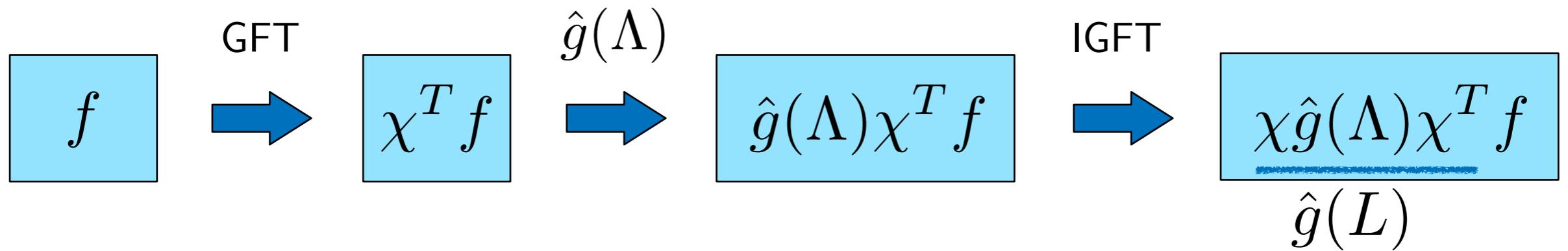


- Important properties can be achieved by properly defining $\hat{g}(L)$, such as localisation of atoms
- Closely related to kernels and regularisation on graphs

A simple example



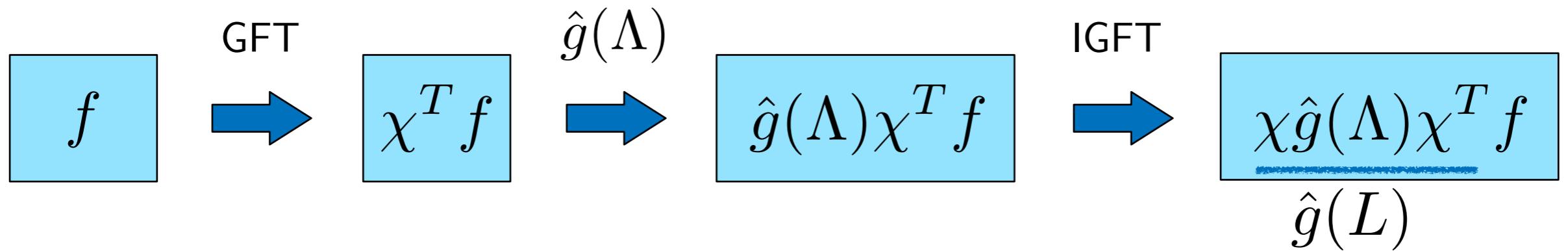
A simple example



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

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

A simple example



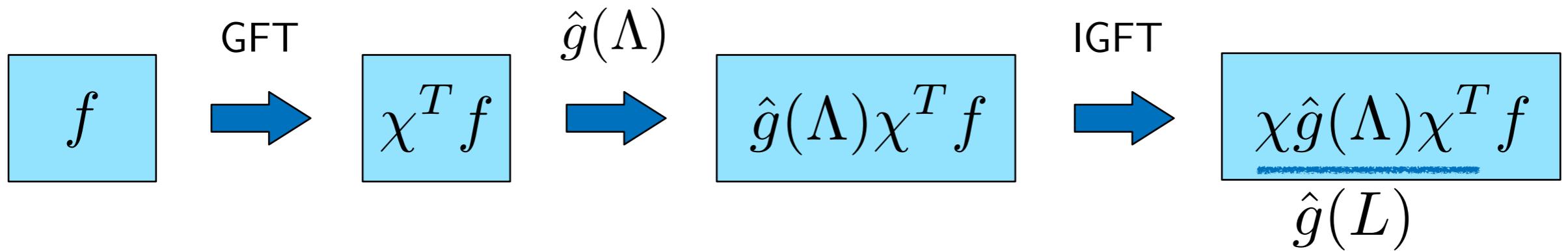
Problem: We observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

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

Annotations:

- Data fitting term: Points to the first term $\|y - f\|_2^2$.
- "Smoothness" assumption: Points to the second term $\gamma y^T L y$.

A simple example



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

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

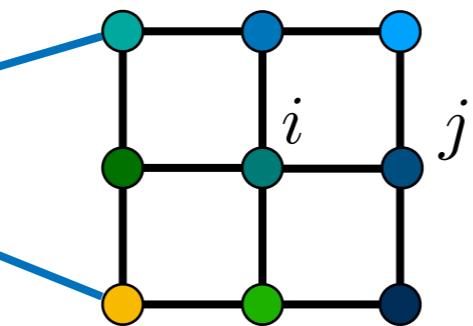
Data fitting term
“Smoothness” assumption

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

Laplacian (Tikhonov) regularisation is equivalent to low-pass filtering in the graph spectral domain!

A simple example

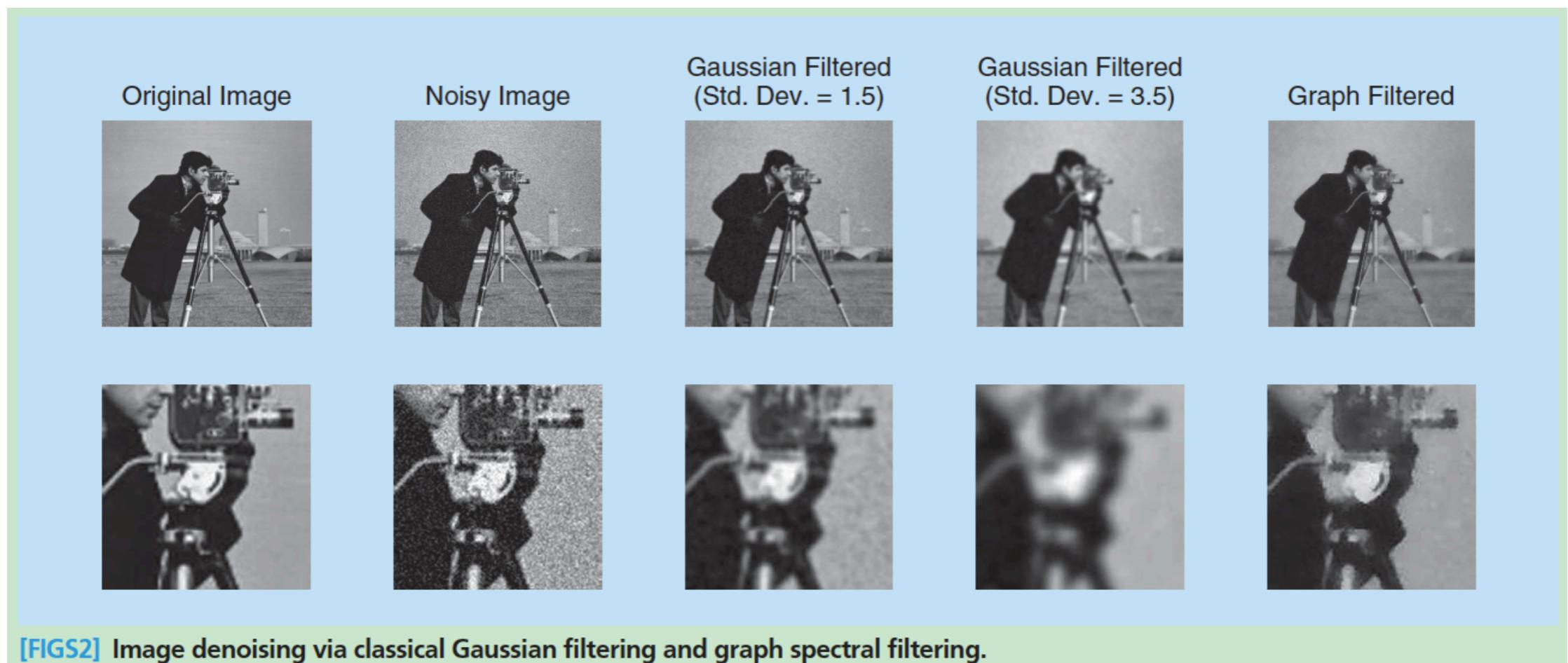
- noisy image as the observed noisy graph signal
- regular grid graph (weights inversely proportional to pixel value difference)



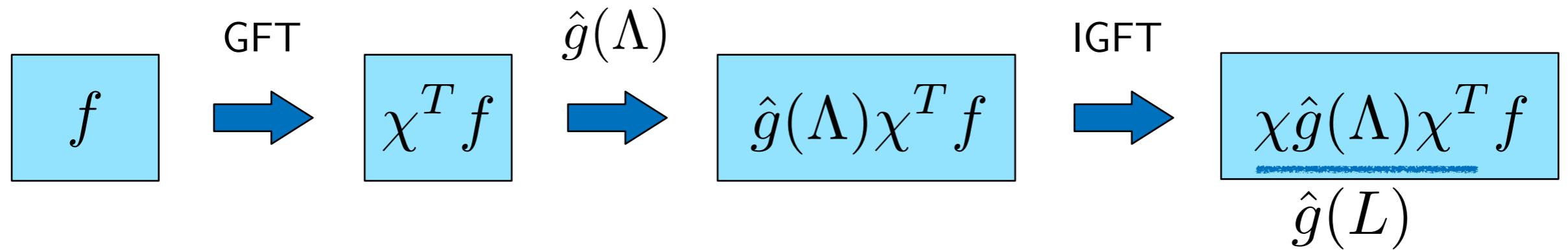
$$w_{ij} = \frac{1}{|f(i) - f(j)|}$$

A simple example

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

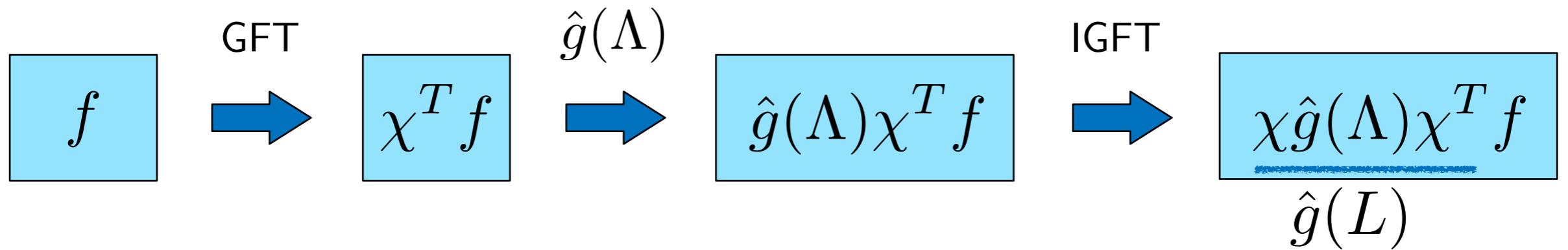


Example designs



Low-pass filters: $\hat{g}(L) = (I + \gamma L)^{-1} = \chi(I + \gamma\Lambda)^{-1}\chi^T$

Example designs



Low-pass filters: $\hat{g}(L) = (I + \gamma L)^{-1} = \chi(I + \gamma \Lambda)^{-1} \chi^T$

Window kernel: Windowed graph Fourier transform

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

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

Parametric polynomials: $\hat{g}_s(L) = \sum_{k=0}^K \alpha_{sk} L^k = \chi \left(\sum_{k=0}^K \alpha_{sk} \Lambda^k \right) \chi^T$

Outline

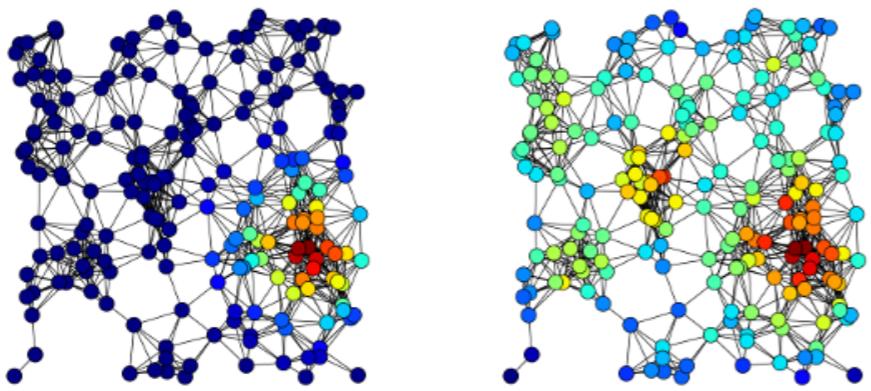
- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

GSP and the literature

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

GSP and the literature

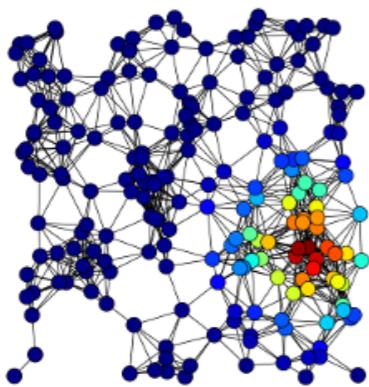
There is a rich literature about data analysis and learning on graphs



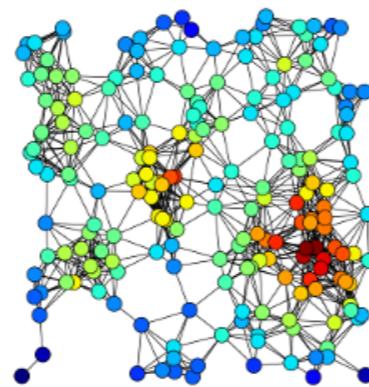
network science

GSP and the literature

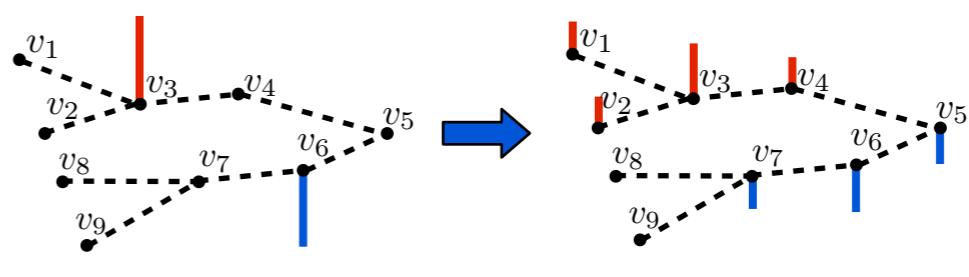
There is a rich literature about data analysis and learning on graphs



network science

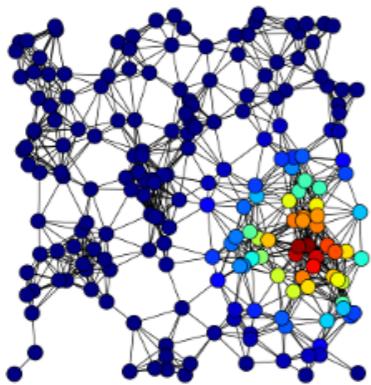


diffusion on graphs

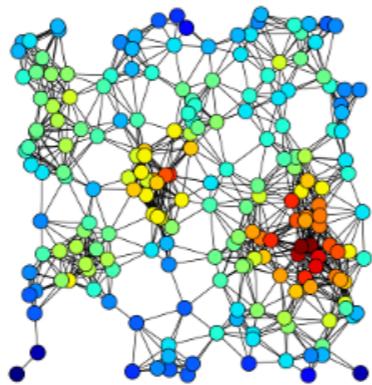


GSP and the literature

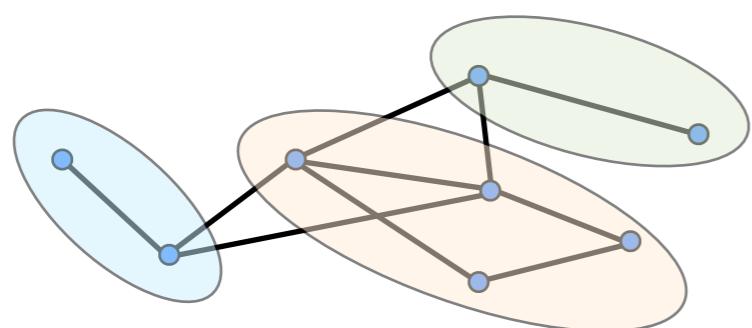
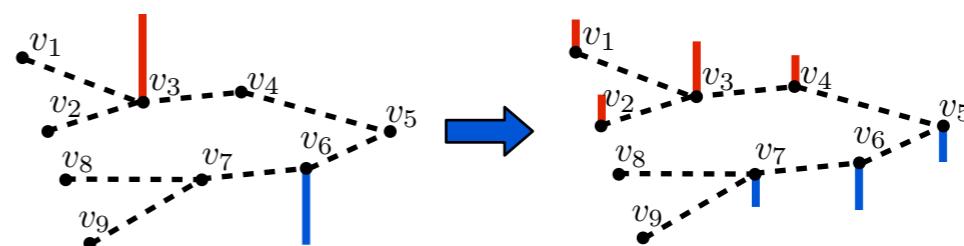
There is a rich literature about data analysis and learning on graphs



network science



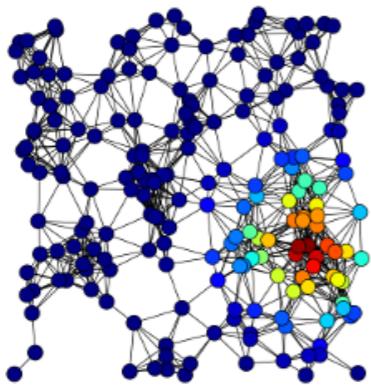
diffusion on graphs



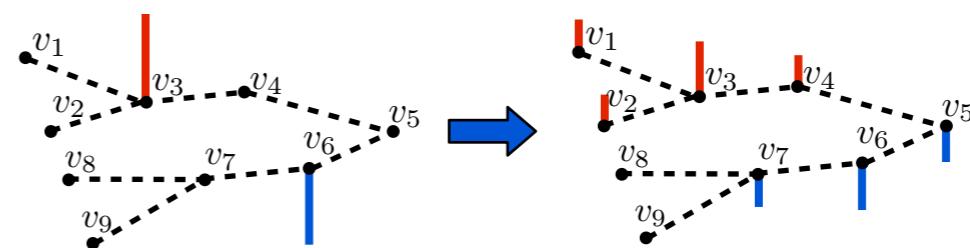
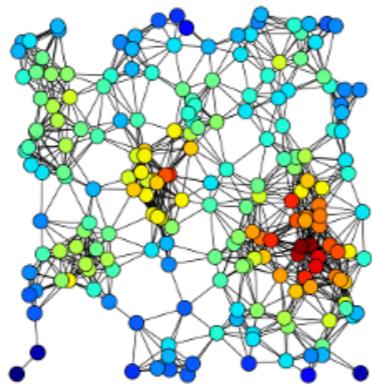
unsupervised learning (dimensionality reduction, clustering)

GSP and the literature

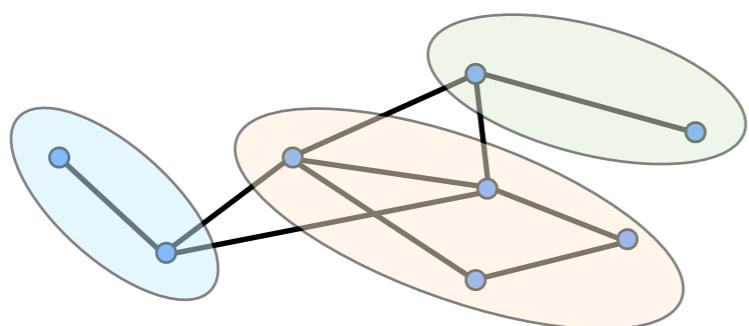
There is a rich literature about data analysis and learning on graphs



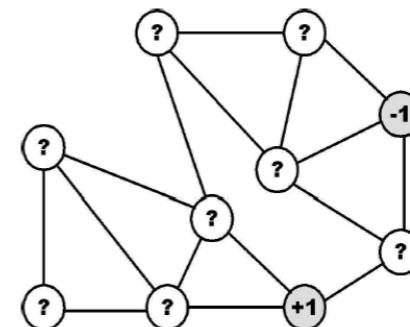
network science



diffusion on graphs



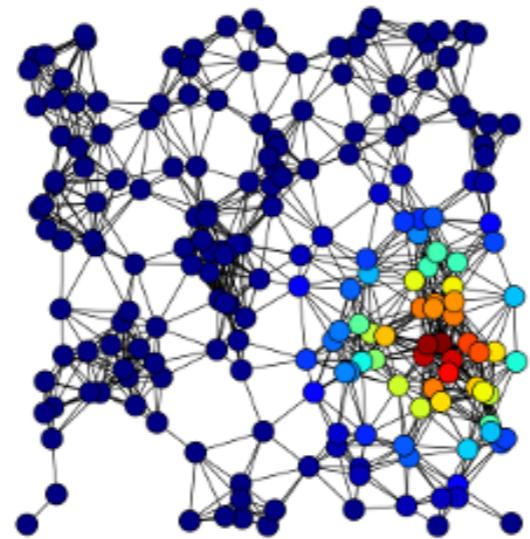
unsupervised learning (dimensionality reduction, clustering)



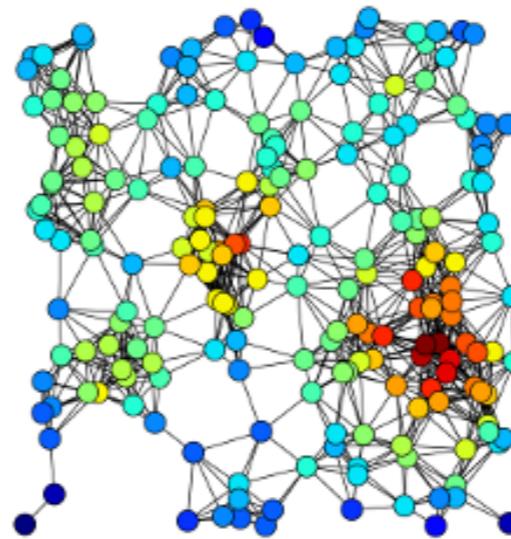
semi-supervised learning

Network centrality

eigenvector centrality



degree centrality

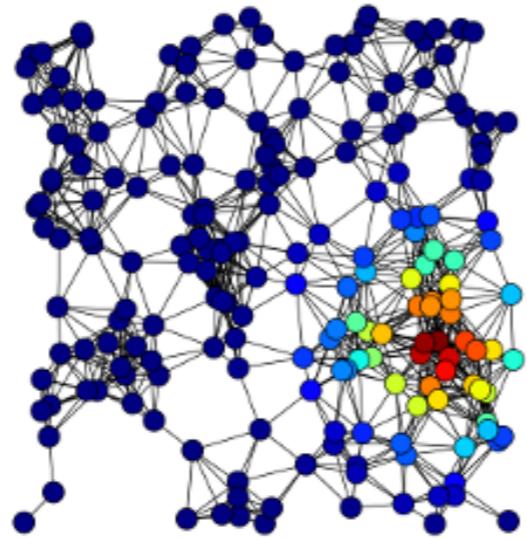


$$Wx = \lambda_{\max}x$$

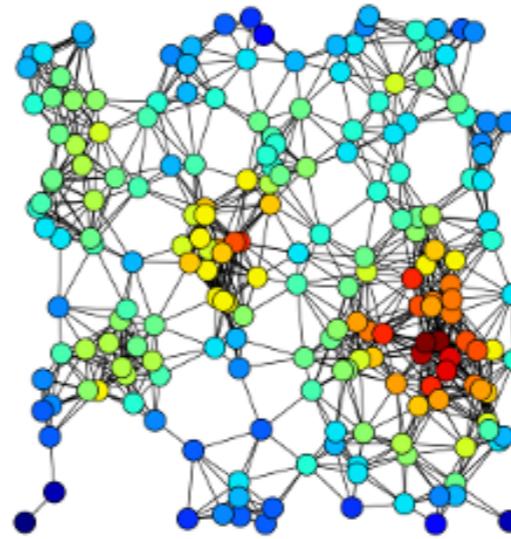
$$d = [d(v_1), \dots, d(v_N)]$$

Network centrality

eigenvector centrality



degree centrality

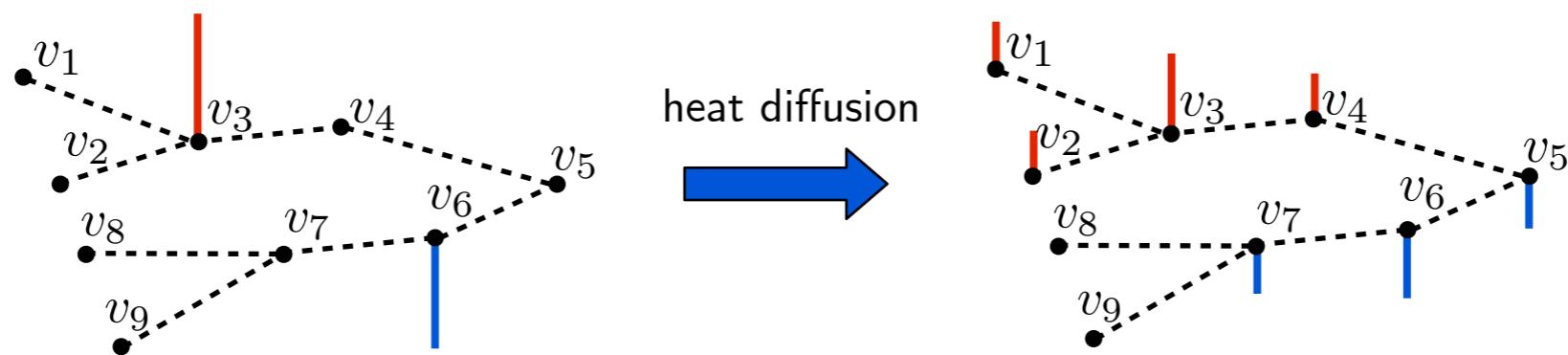


$$Wx = \lambda_{\max}x$$

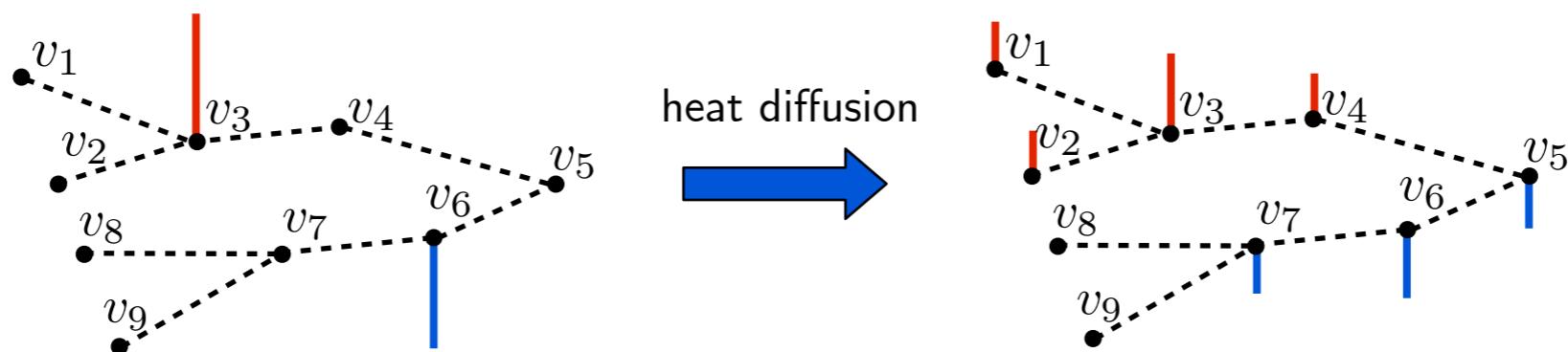
$$d = [d(v_1), \dots, d(v_N)]$$

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

Diffusion on graphs



Diffusion on graphs



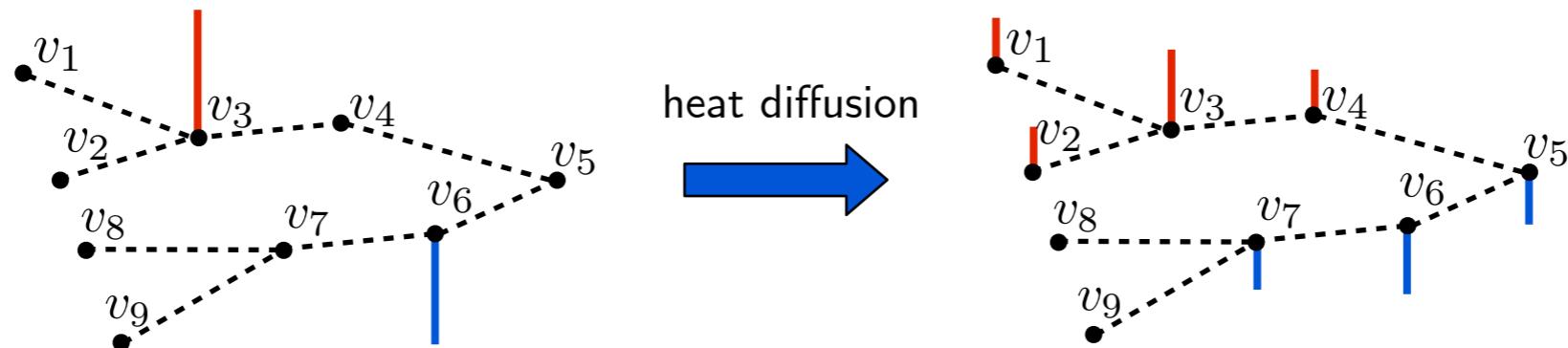
$$\frac{\partial x}{\partial \tau} - Lx = 0$$

$$x(v, 0) = x_0(v)$$



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

Diffusion on graphs



$$\frac{\partial x}{\partial \tau} - Lx = 0$$

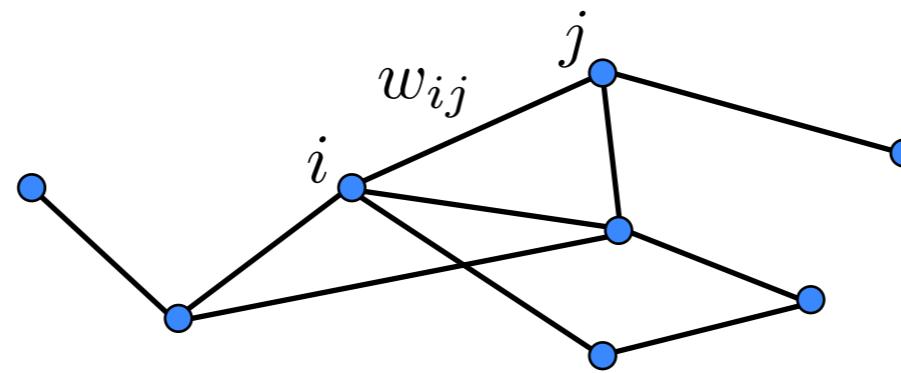
$$x(v, 0) = x_0(v)$$



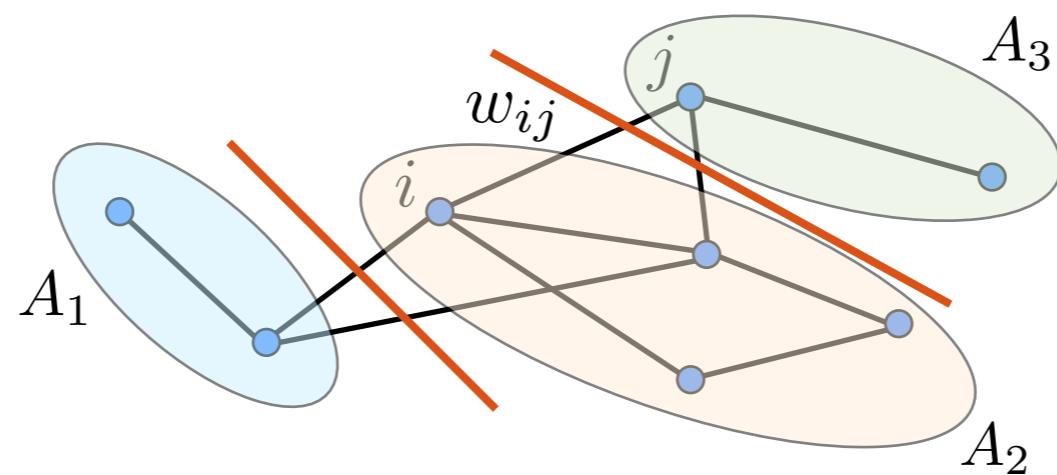
$$x(v, \tau) = e^{-\tau L} x_0(v)$$

- 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

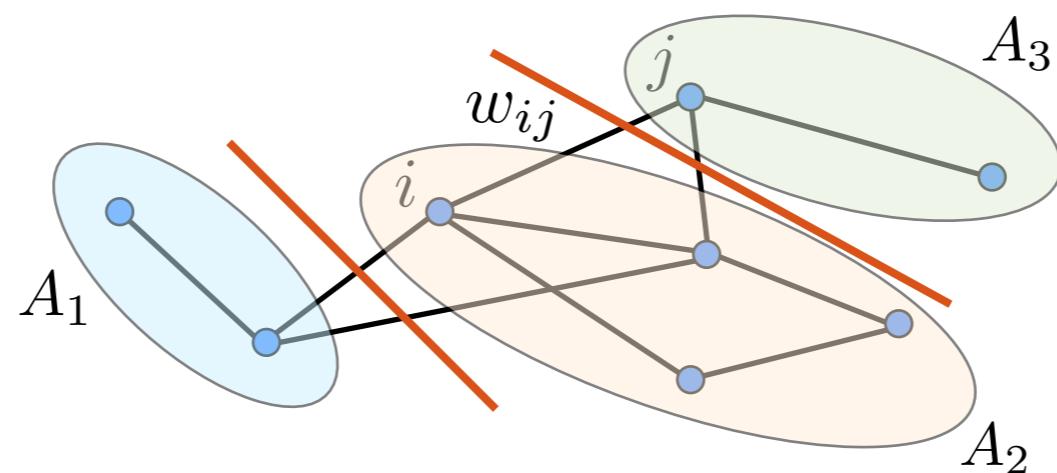
Graph clustering (community detection)



Graph clustering (community detection)

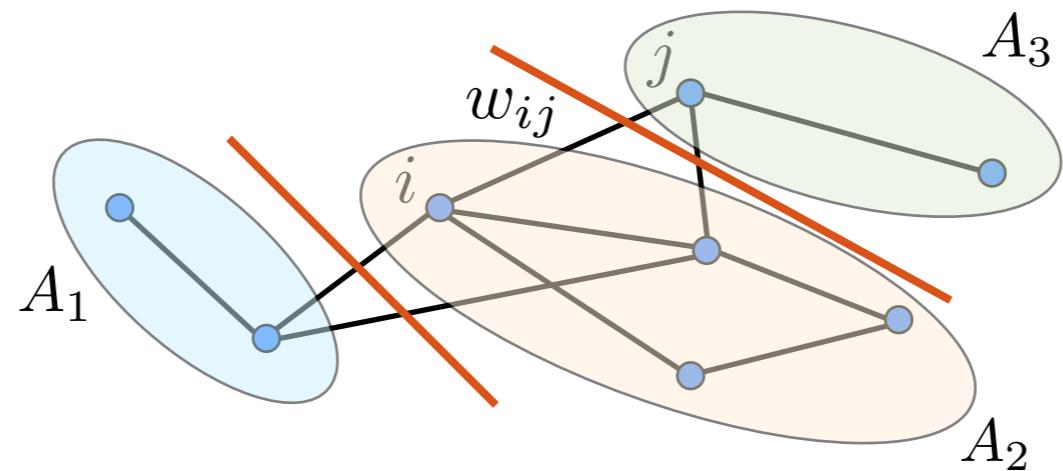


Graph clustering (community detection)



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

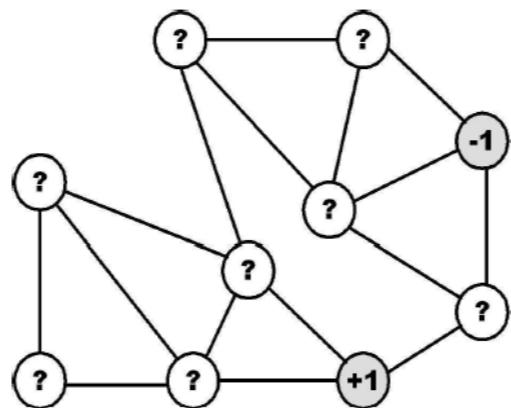
Graph clustering (community detection)



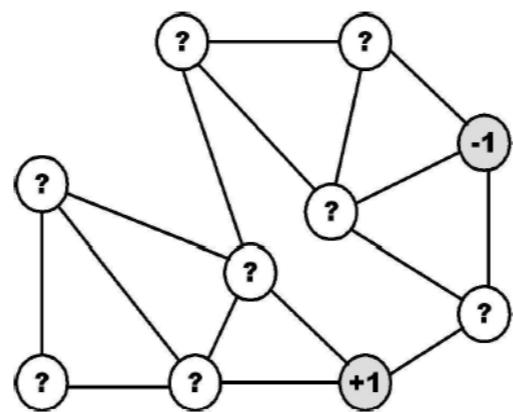
$$NCut(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \overline{A_i})}{vol(A_i)}$$

- first k eigenvectors of graph Laplacian minimise the graph cut
- eigenvectors of graph Laplacian enable a Fourier-like analysis for graph signals

Semi-supervised learning



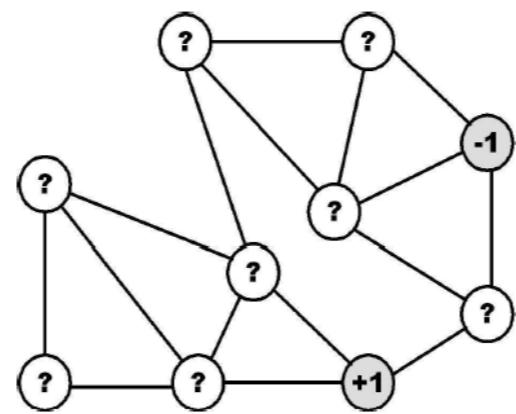
Semi-supervised learning



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

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

Semi-supervised learning



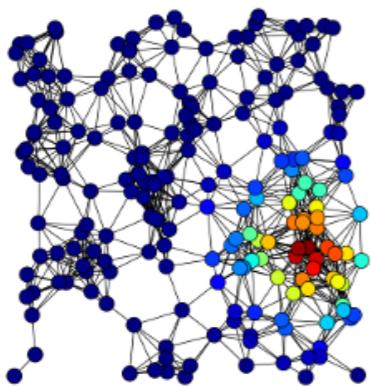
$$y : \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ +1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

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

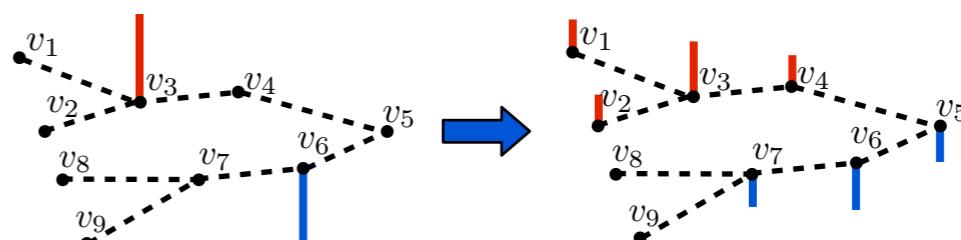
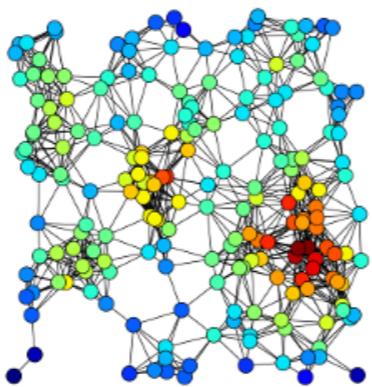
- learning by assuming smoothness of predicted labels
- this is equivalent to a denoising problem for graph signal y

GSP and the literature

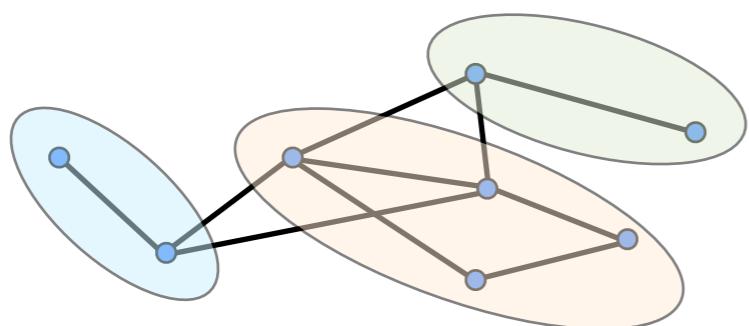
centrality, diffused information, class membership, node labels (and node-level features in general) can ALL be viewed as graph signals



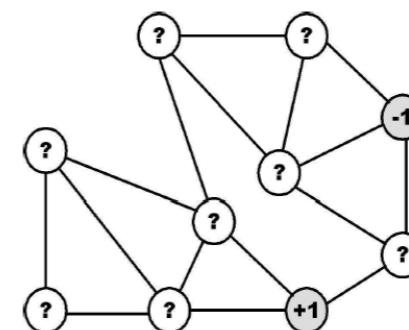
network science



network diffusion



unsupervised learning (dimensionality reduction, clustering)

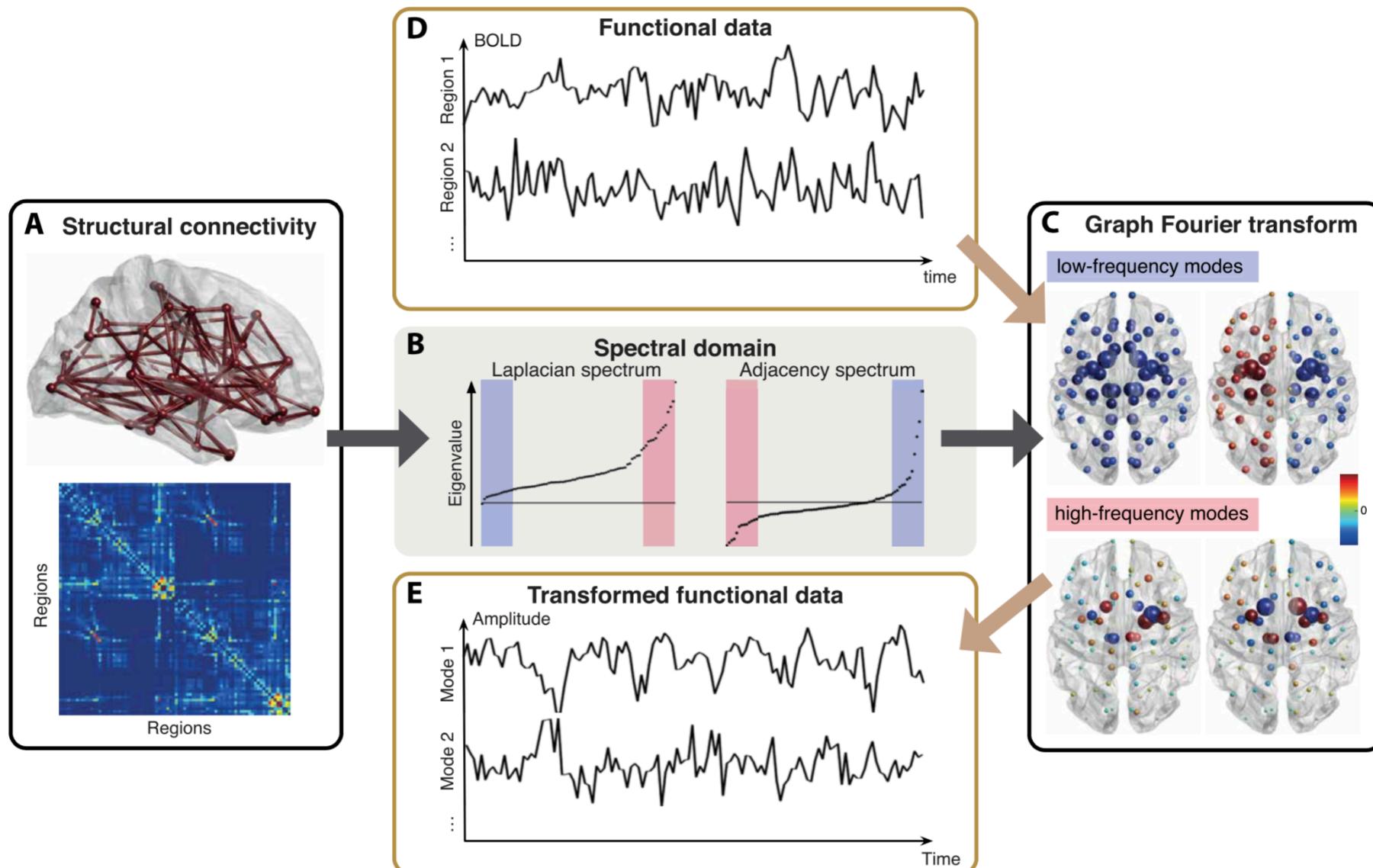


semi-supervised learning

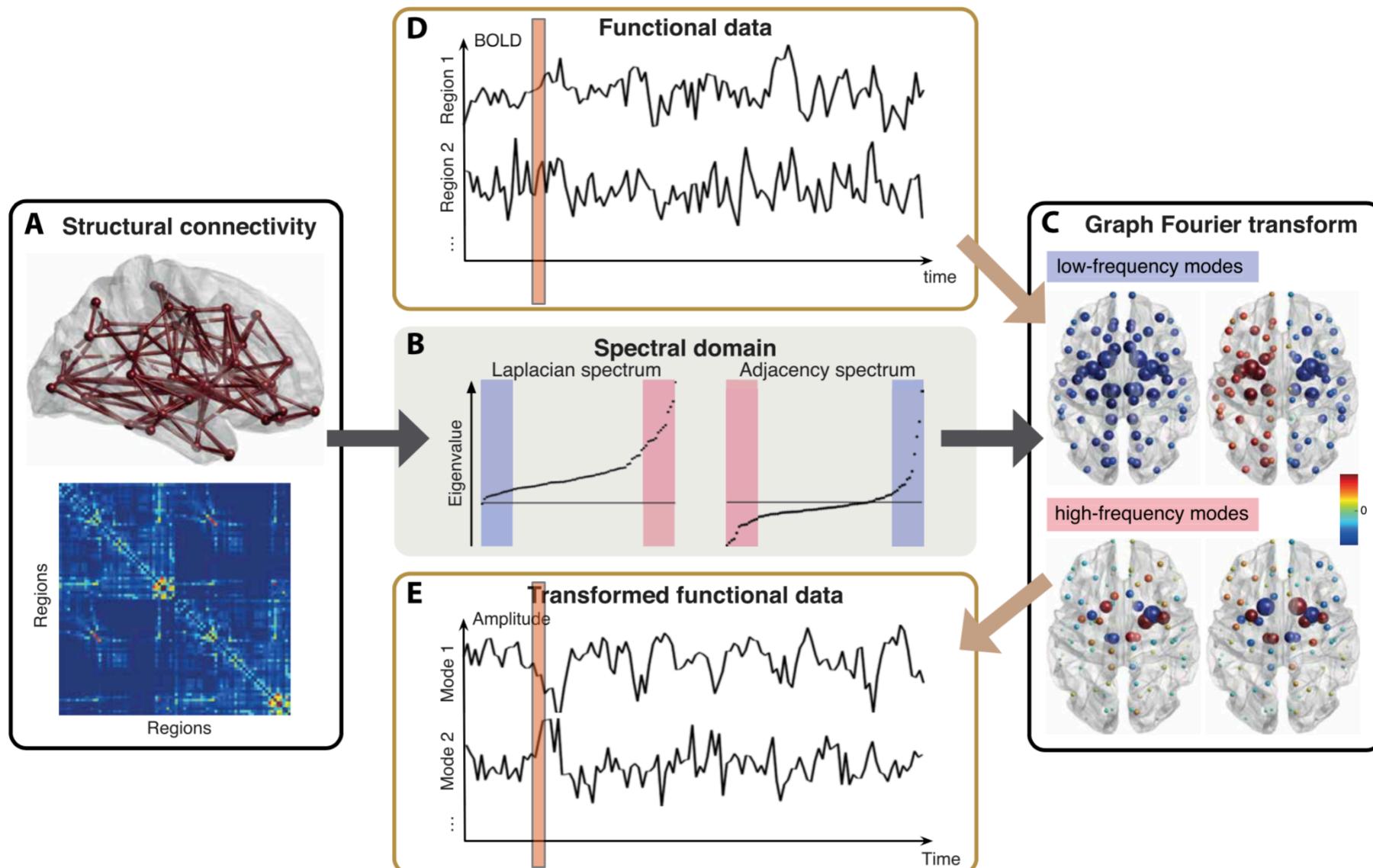
Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

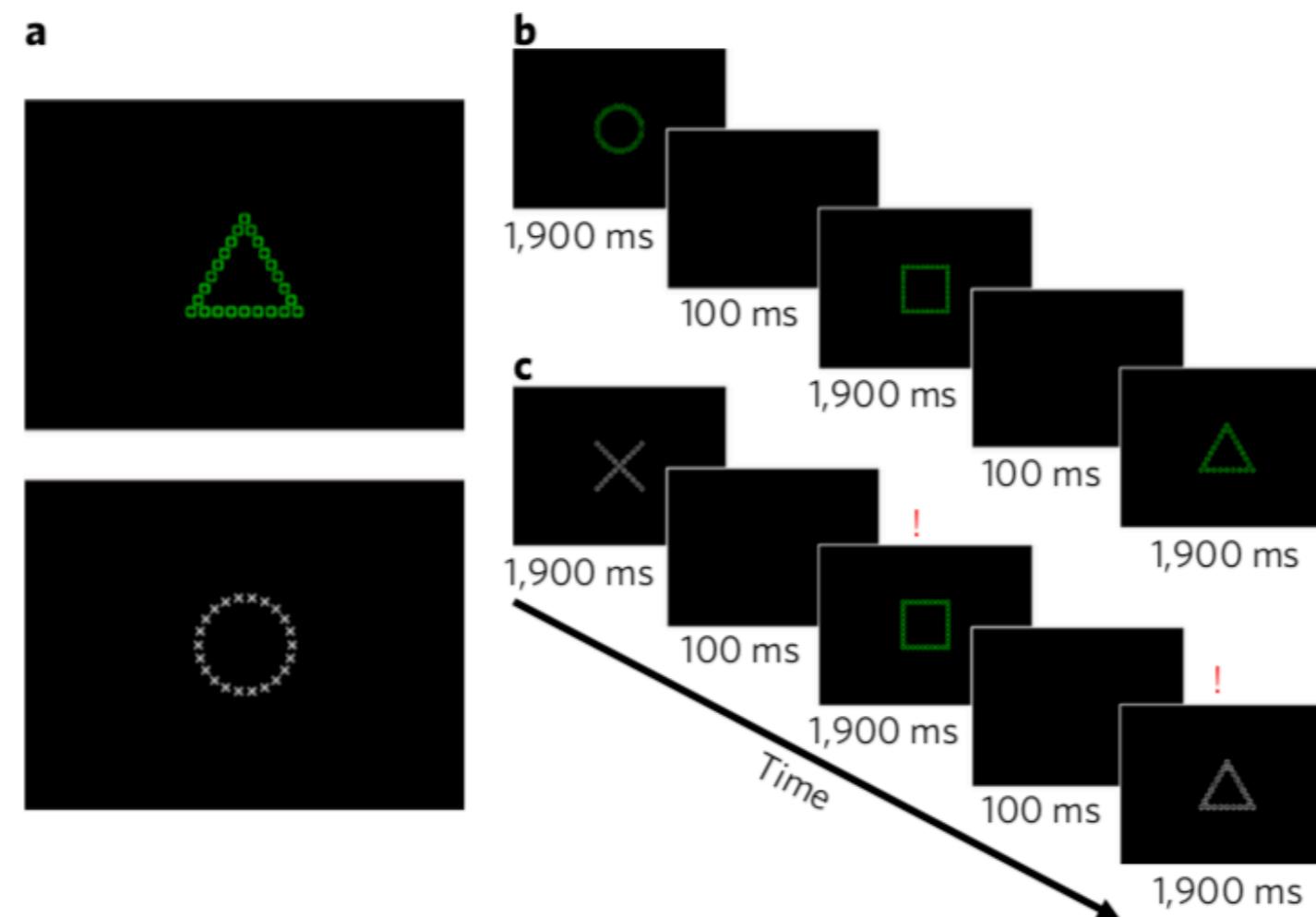
A typical analysis framework



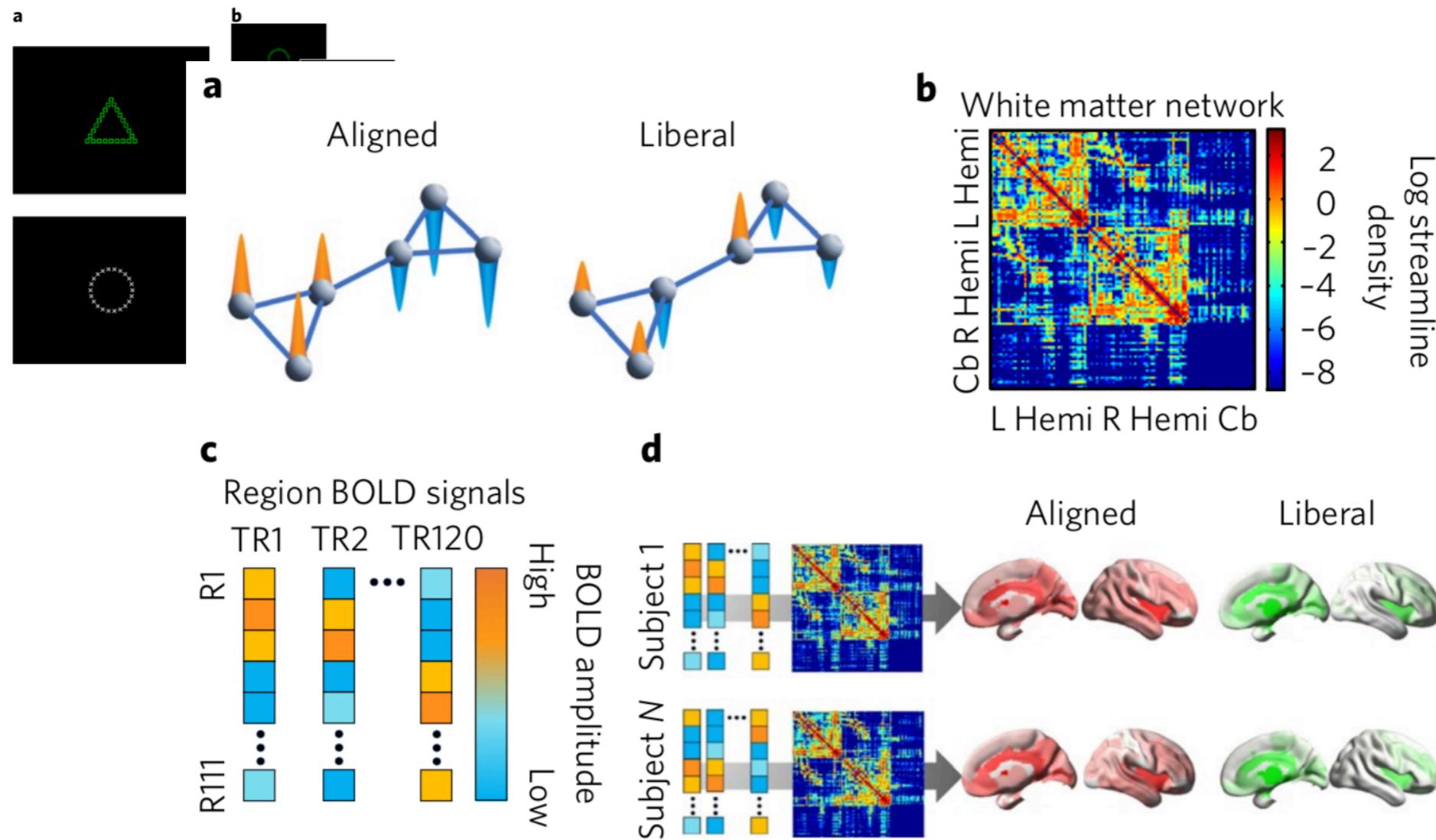
A typical analysis framework



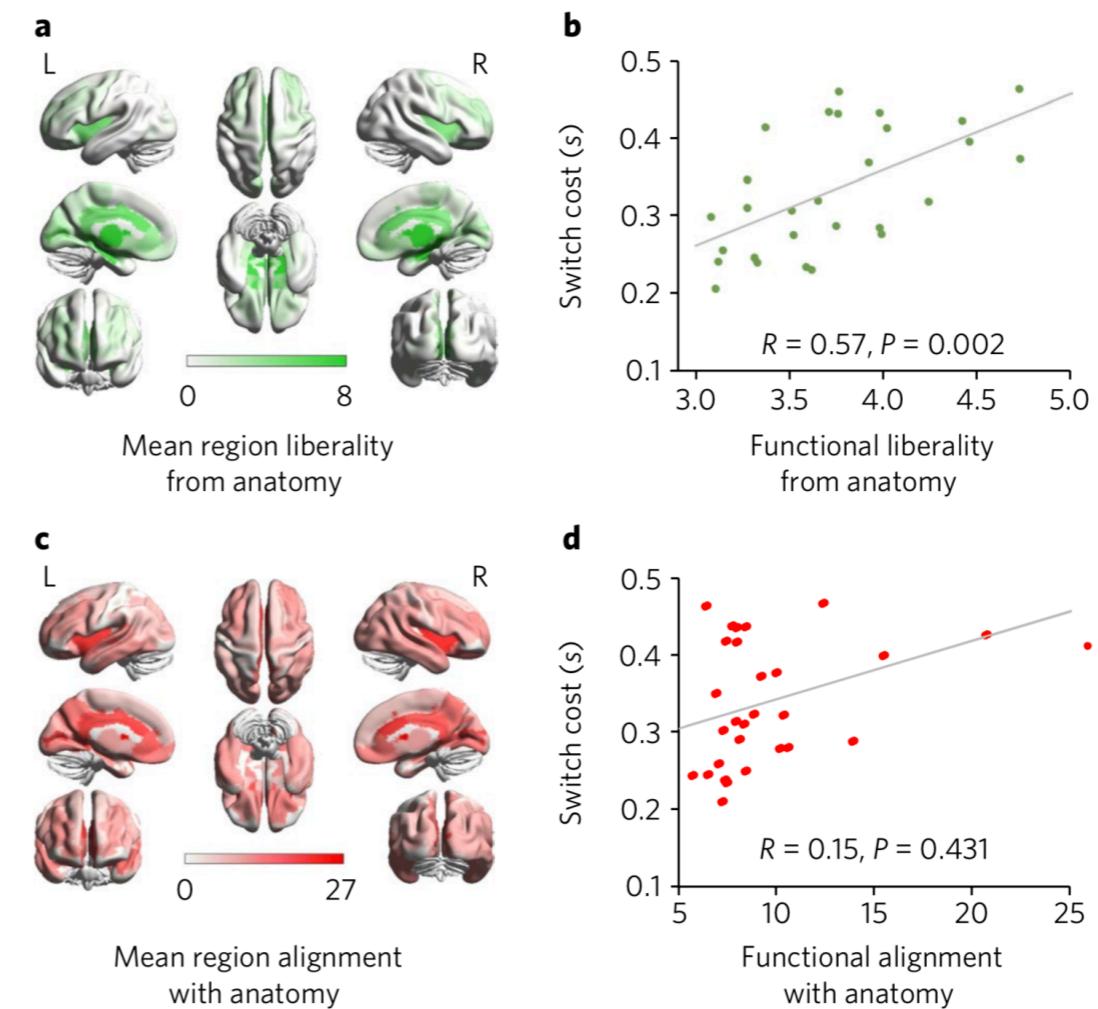
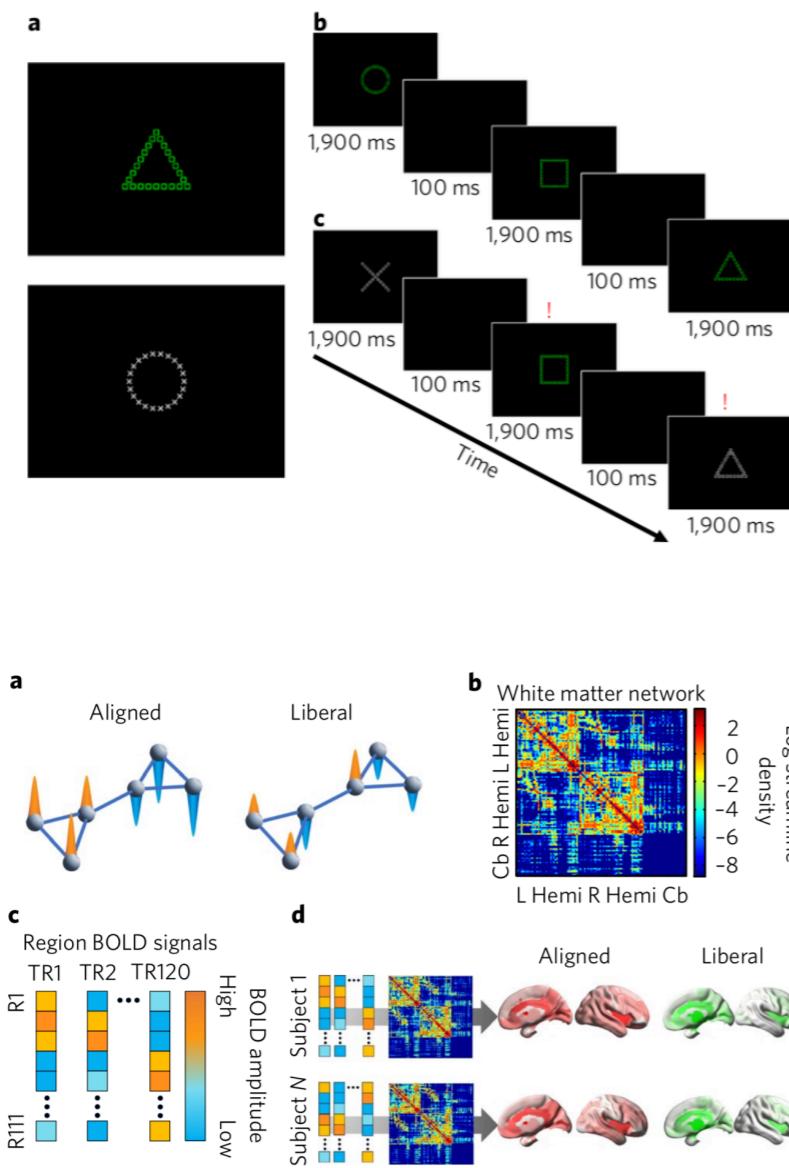
Application I: Understanding brain functioning



Application I: Understanding brain functioning

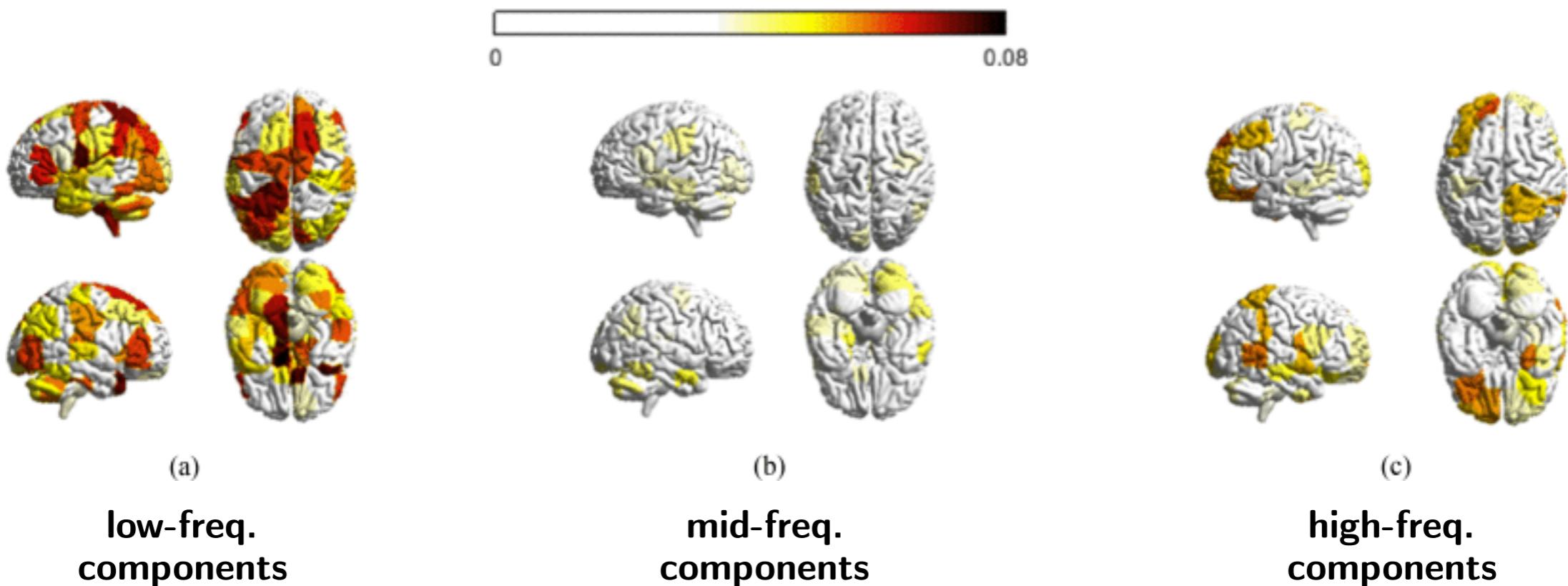


Application I: Understanding brain functioning



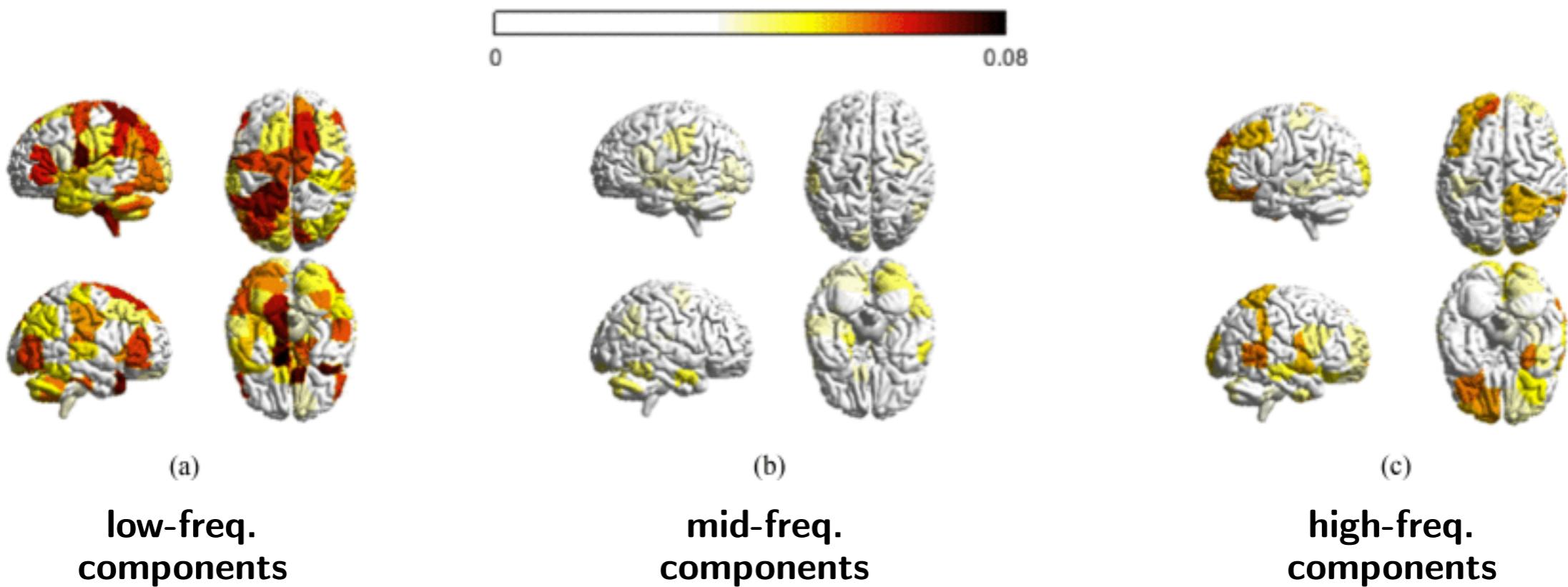
**liberality (large high-freq. components)
associated with high switching cost**

Application I: Understanding brain functioning



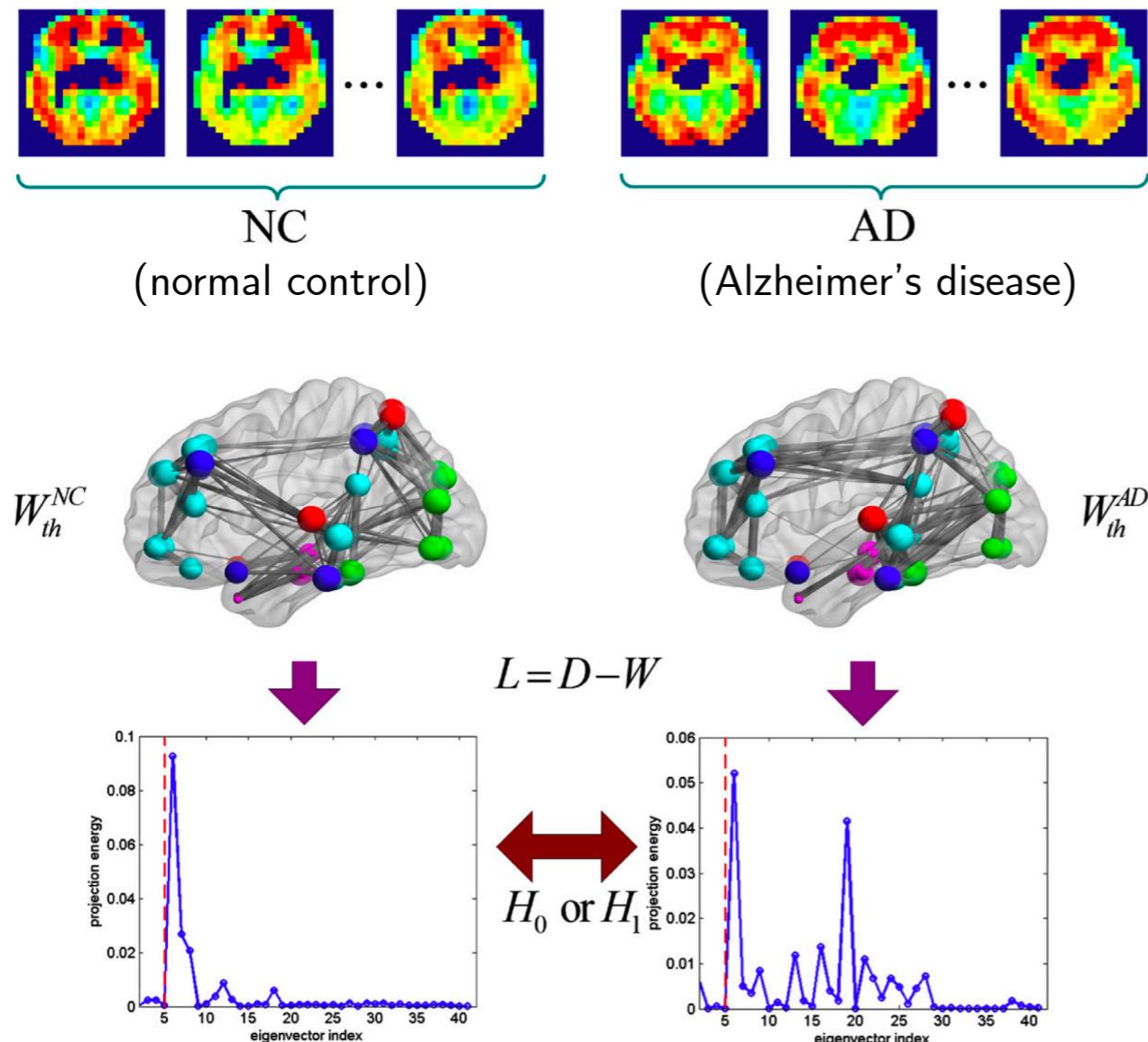
- record BOLD signals while responding to sequentially presented stimuli

Application I: Understanding brain functioning

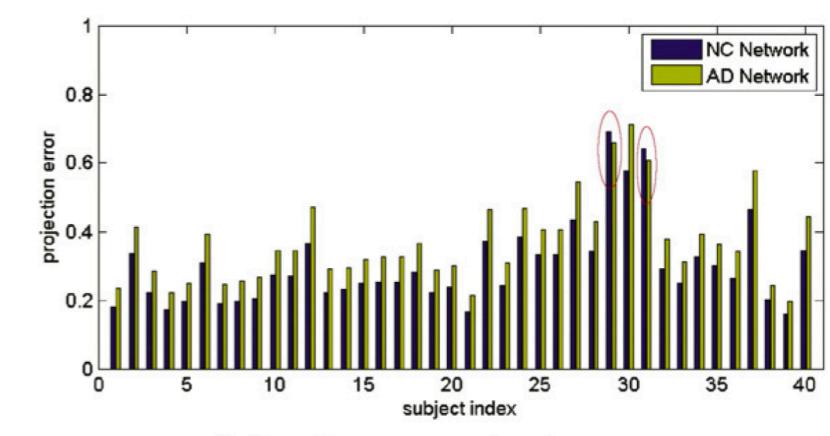
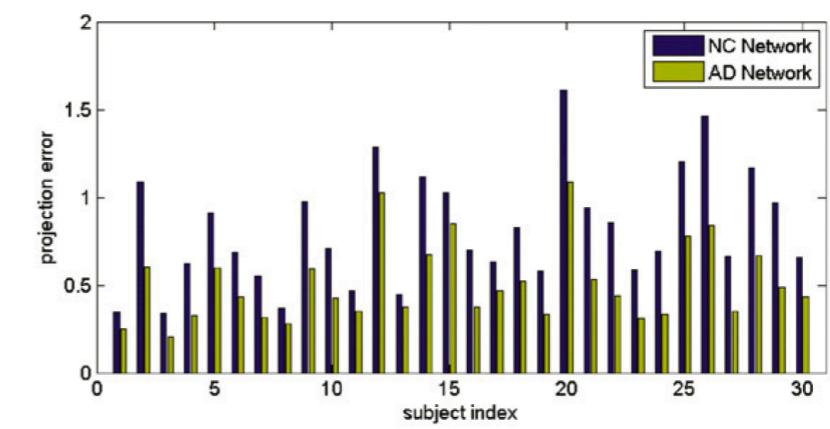
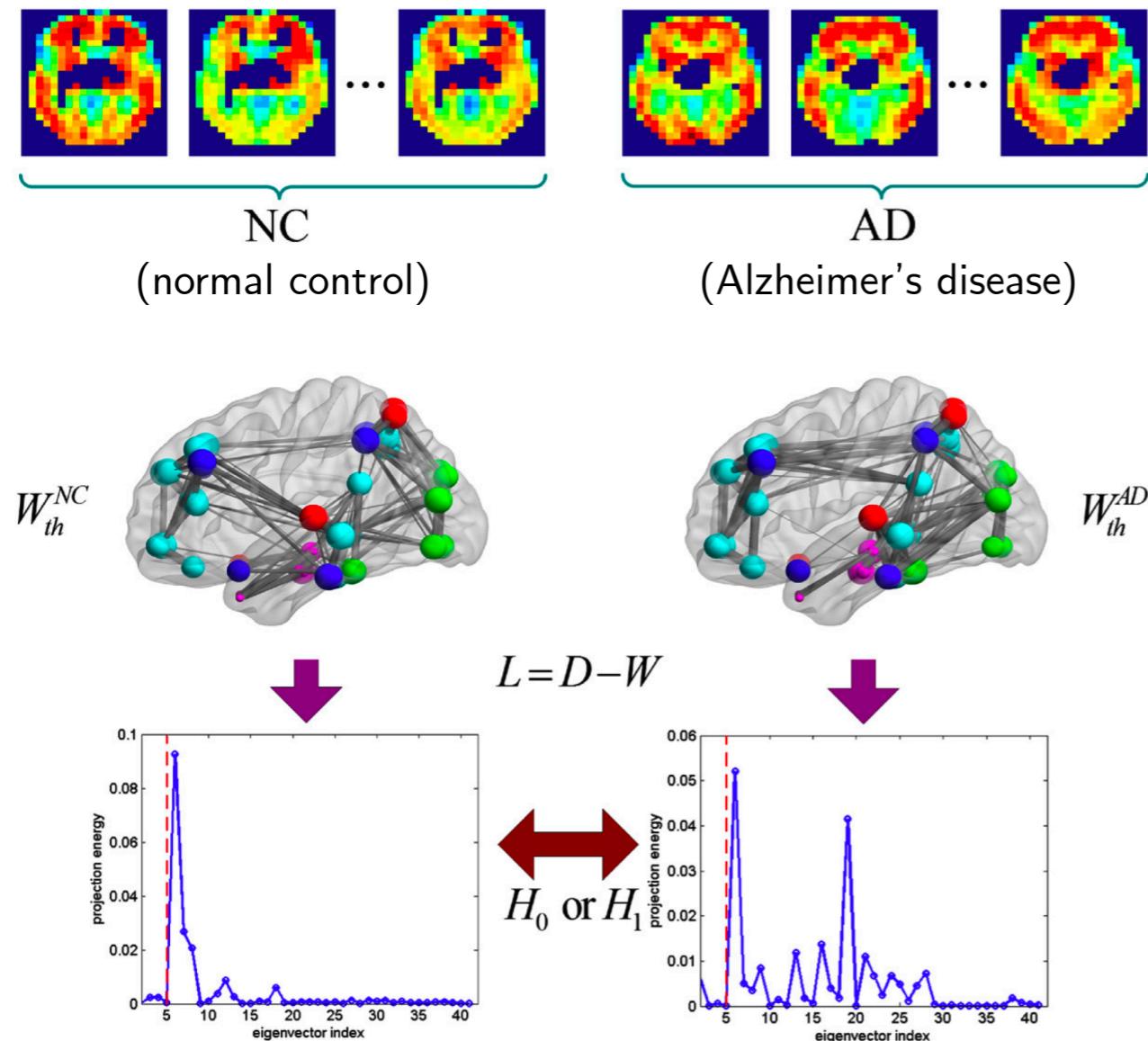


- record BOLD signals while responding to sequentially presented stimuli
- it favours learning to have
 - **smooth, spread** signals (low-freq.) when facing **unfamiliar** task
 - **varied, spiking** signals (high-freq.) when task becomes **familiar**

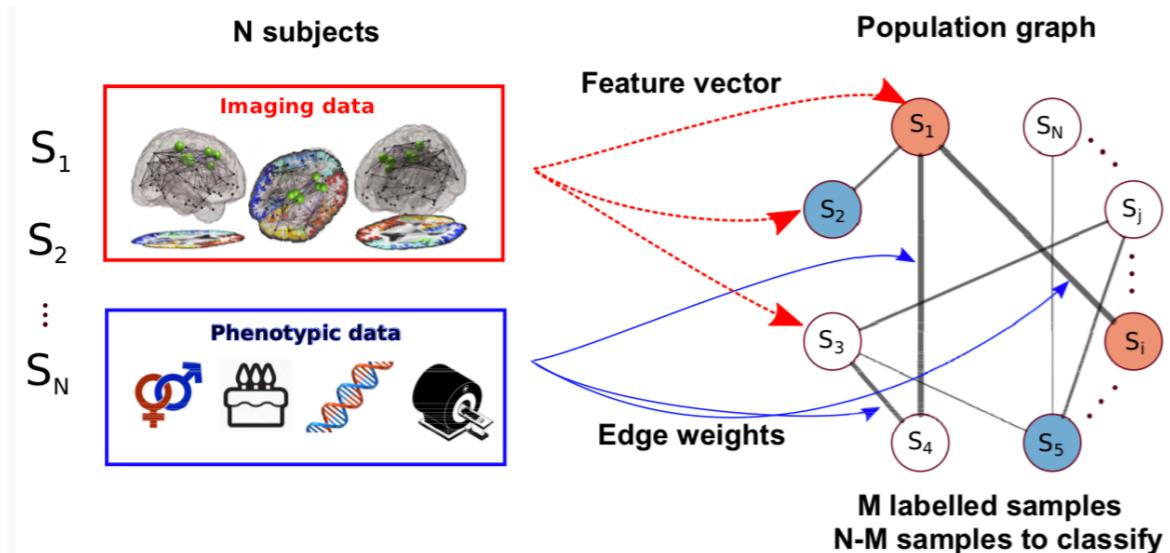
Application II: Disease classification



Application II: Disease classification



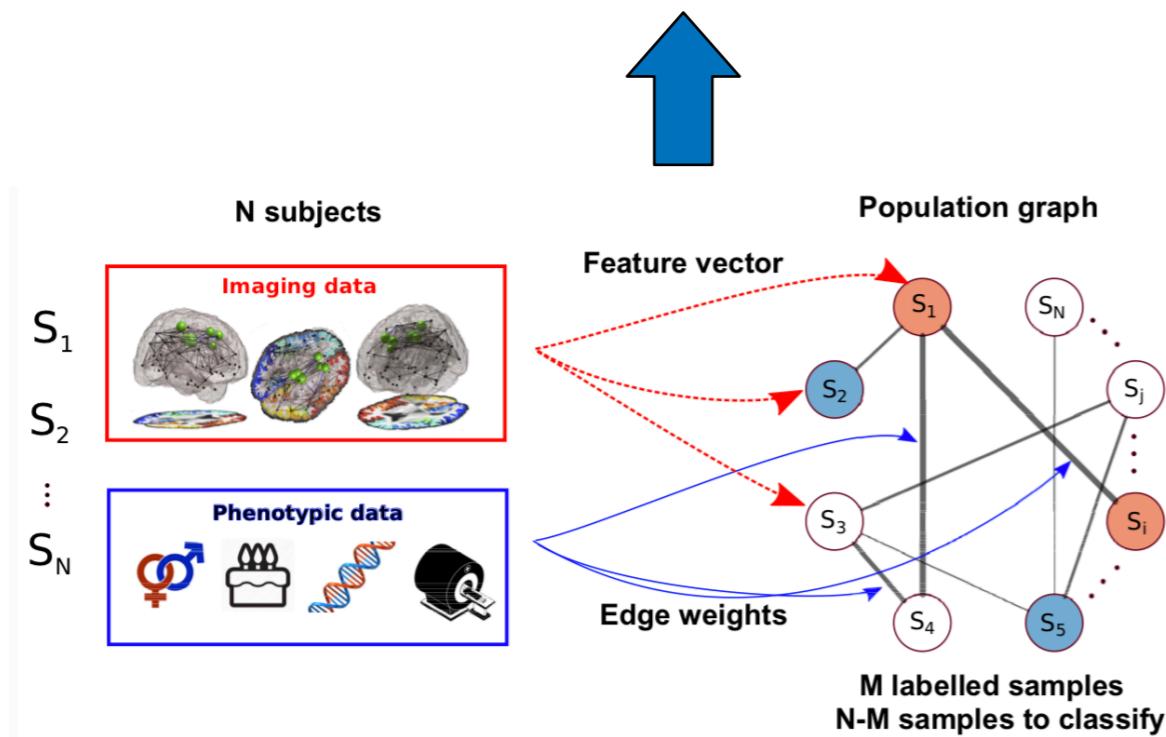
Application II: Disease classification



Application II: Disease classification

ADNI (structural MRI): volumes of brain structures

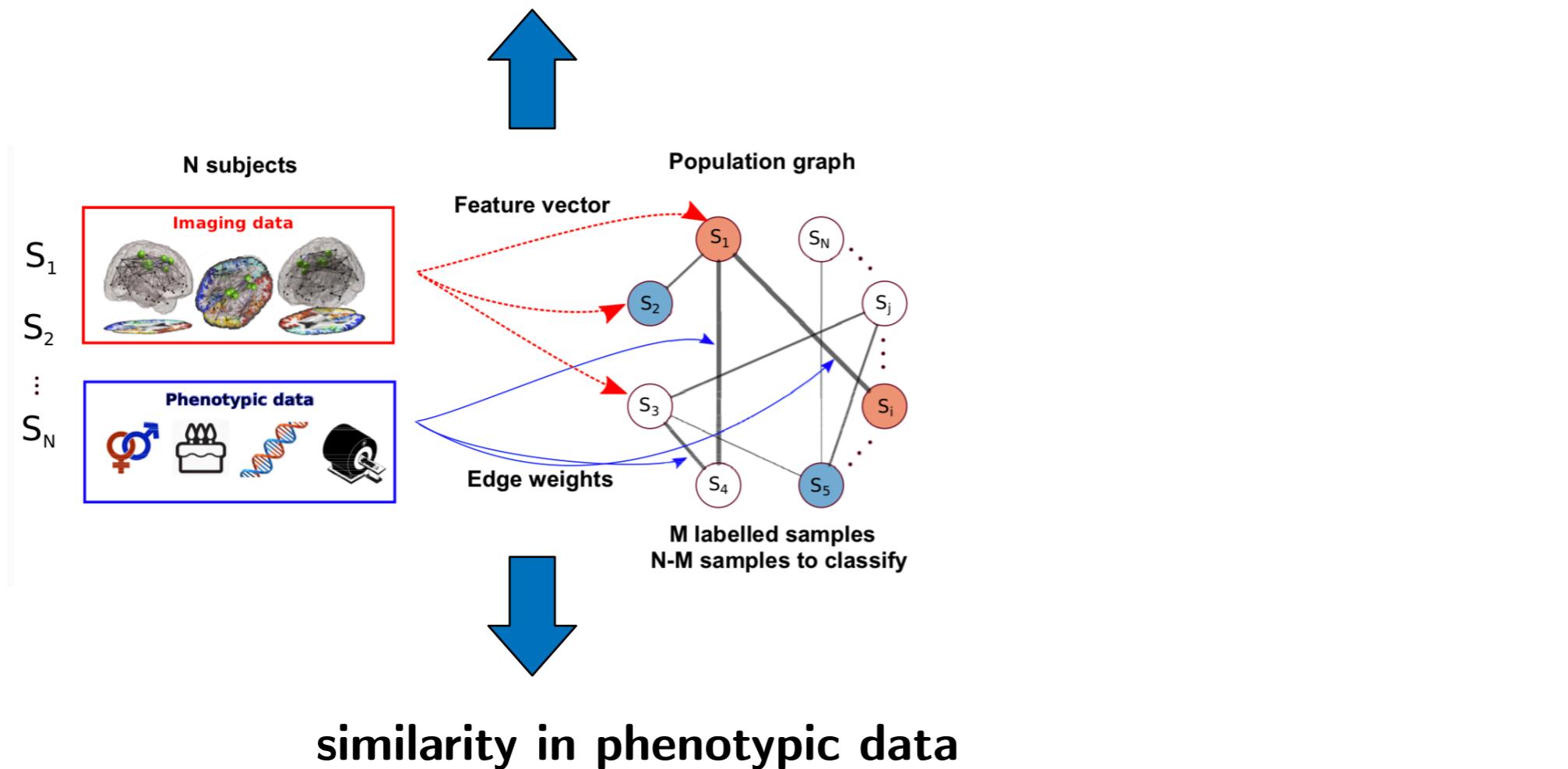
ABIDE (fMRI): off-diagonal of functional connectivity



Application II: Disease classification

ADNI (structural MRI): volumes of brain structures

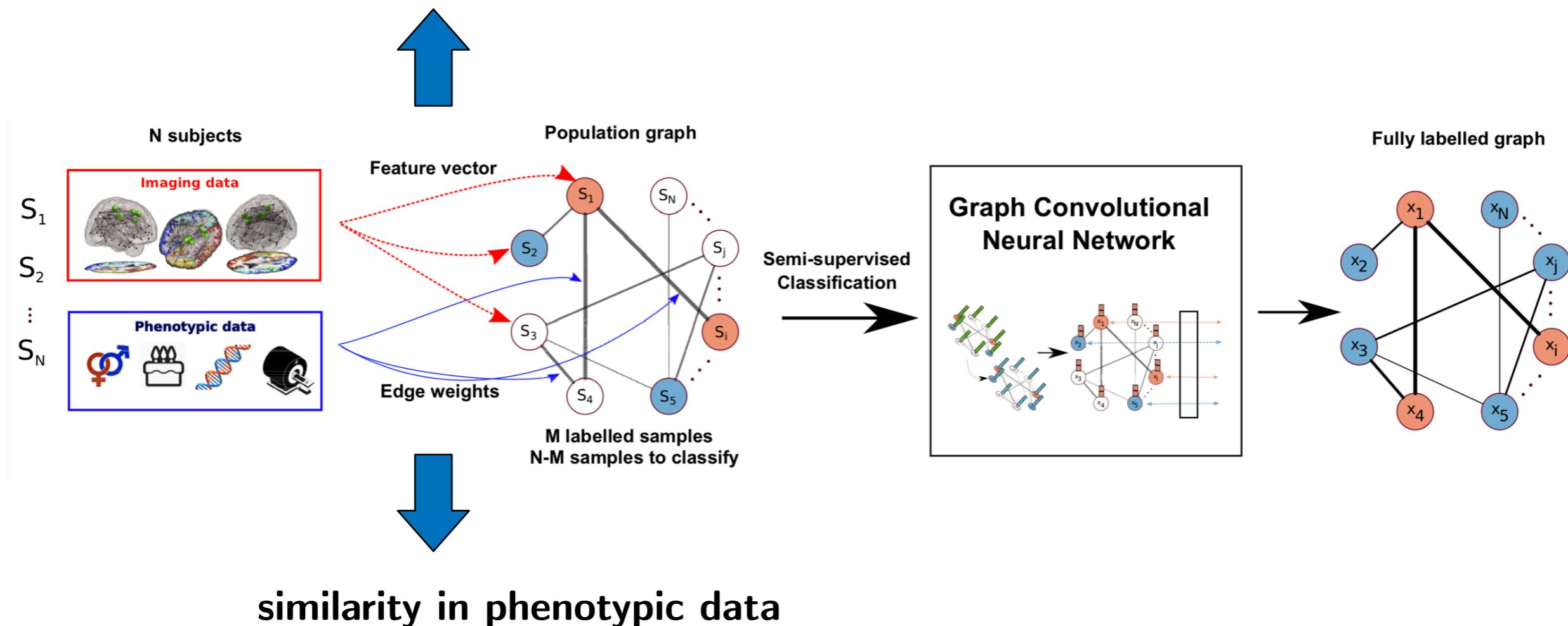
ABIDE (fMRI): off-diagonal of functional connectivity



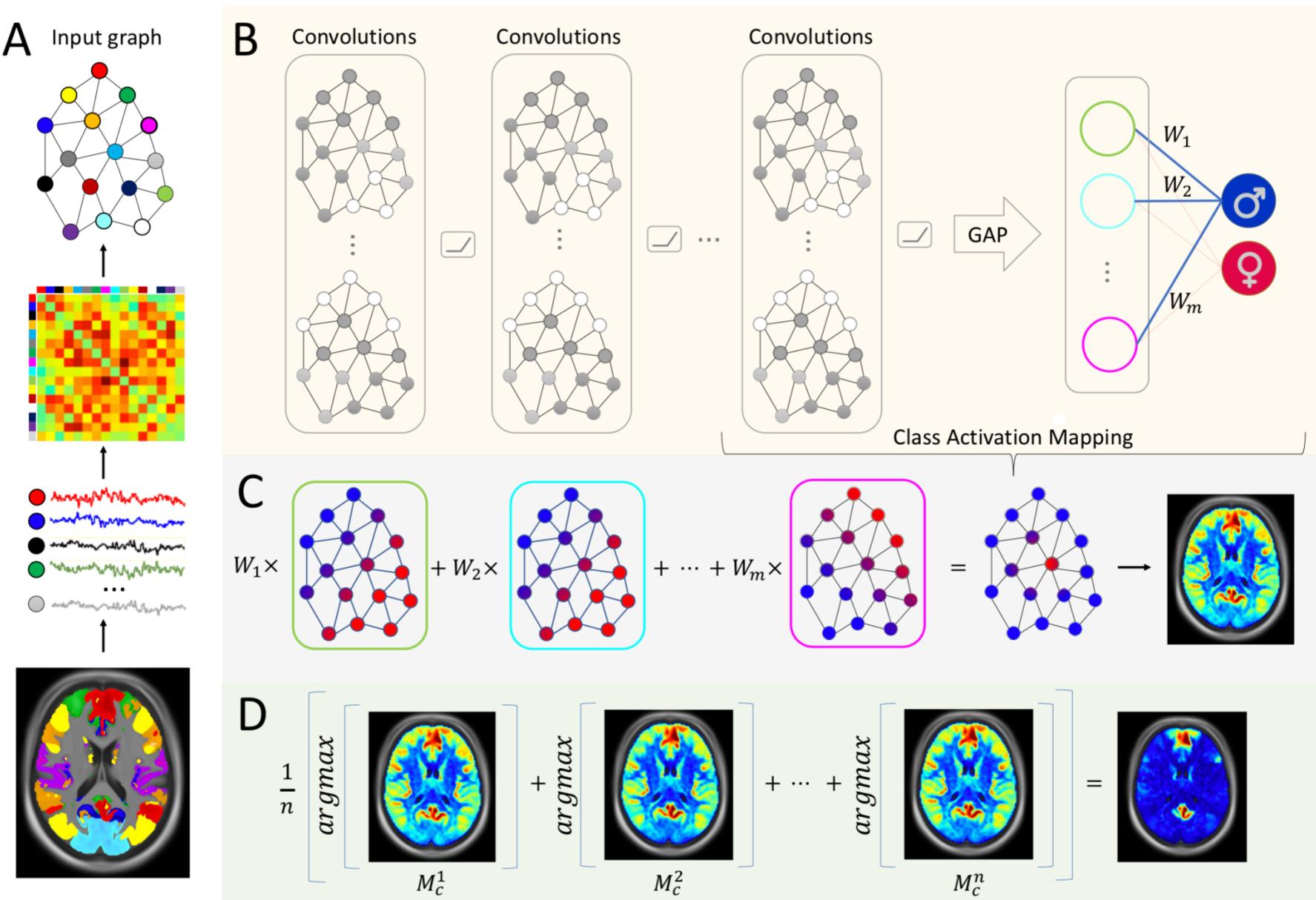
Application II: Disease classification

ADNI (structural MRI): volumes of brain structures

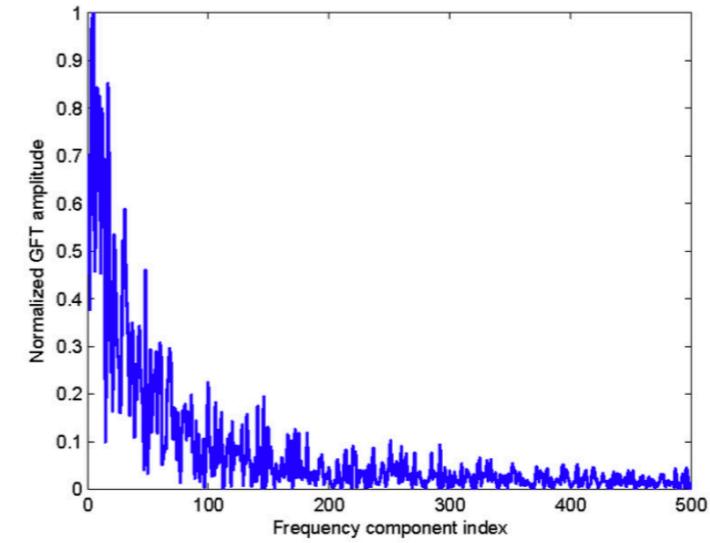
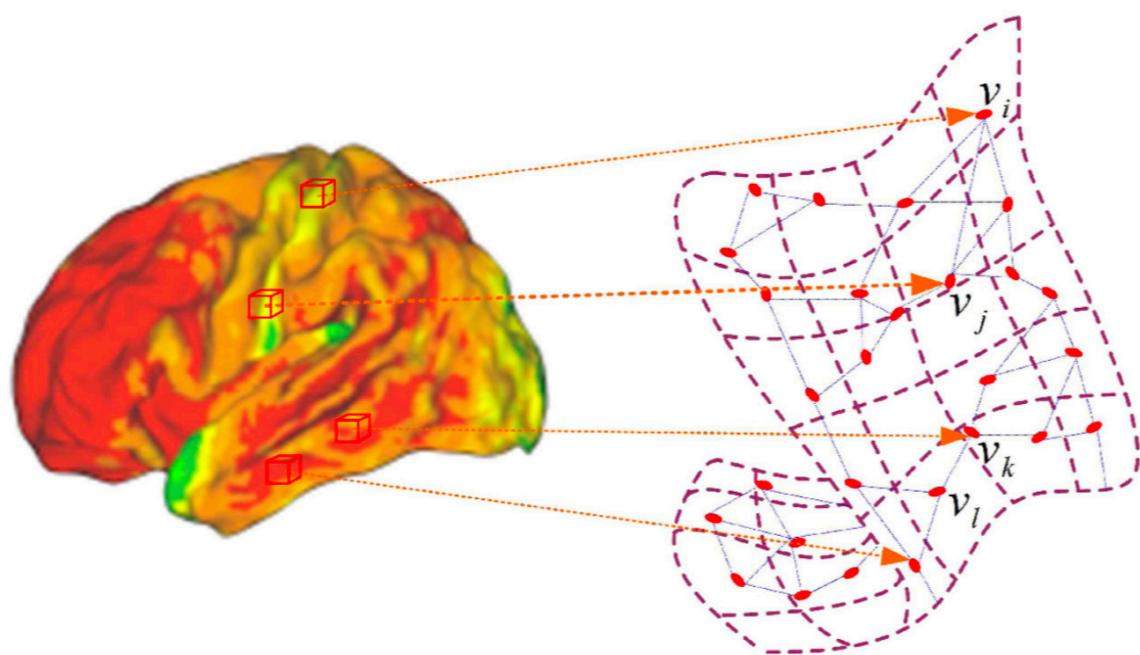
ABIDE (fMRI): off-diagonal of functional connectivity



Application III: Gender classification

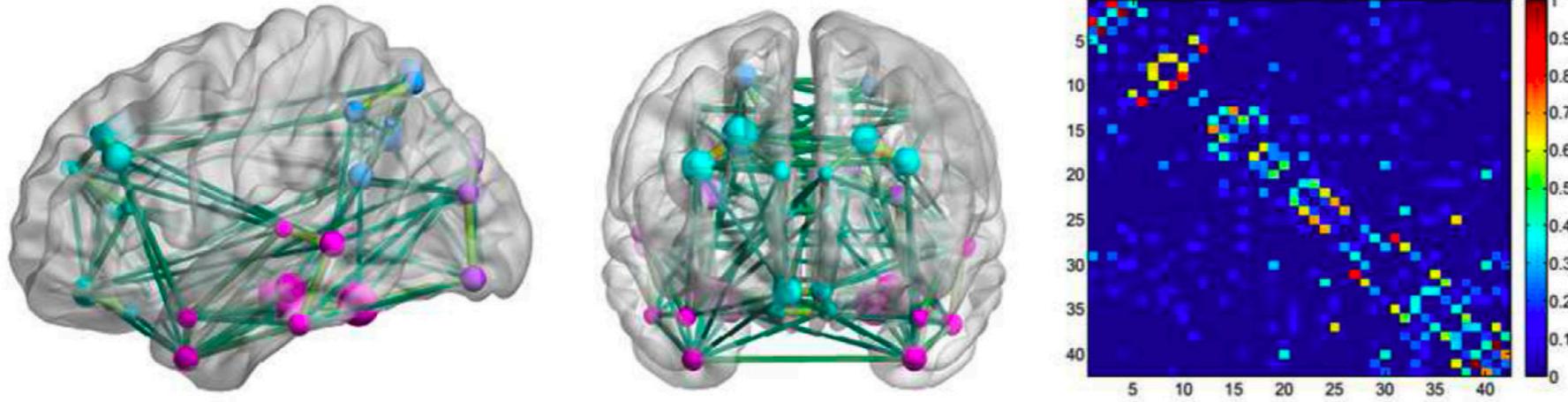


Application IV: Inferring brain connectivity

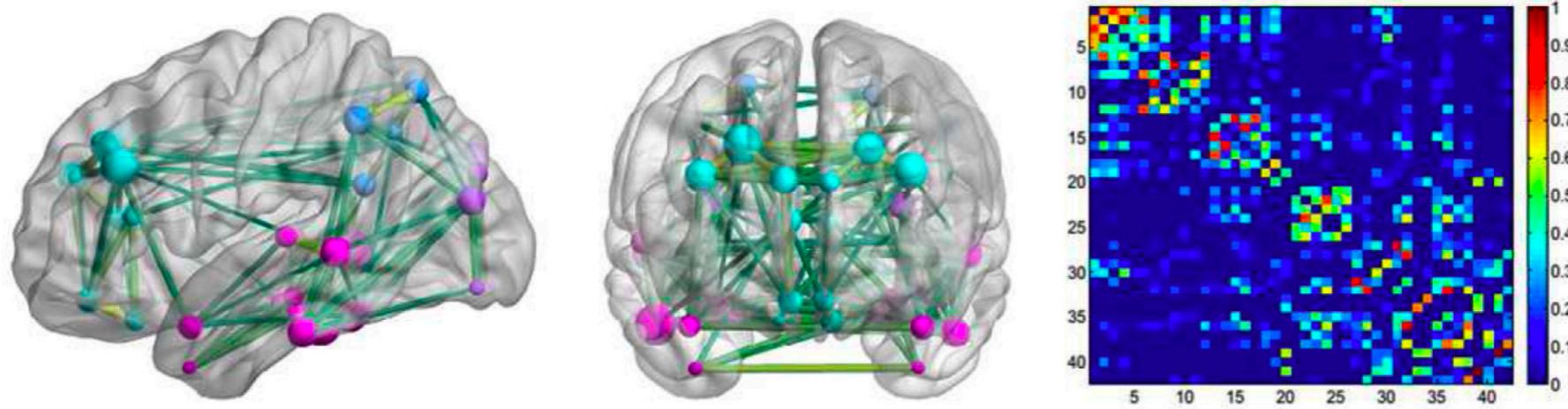


Application IV: Inferring brain connectivity

Alzheimer's disease



normal control



Hu et al., "A spectral graph regression model for learning brain connectivity of Alzheimer's disease", PLOS ONE, 2015.
Shen et al., "Nonlinear structural vector autoregressive models for inferring effective brain network connectivity", 2016.

Future of GSP

- Mathematical models for graph signals
 - global and local smoothness / regularity
 - underlying physical processes
- Graph construction
 - how to infer topologies given observed data?
- Fast implementation
 - fast graph Fourier transform
 - distributed processing
- Connection to / combination with other fields
 - statistical machine learning
 - deep learning on graphs and manifolds
- Key applications

Bronstein et al., "Geometric deep learning", IEEE SPM, 2017.

Wu et al., "A comprehensive survey on graph neural networks", arXiv, 2019.

Resources

- Three tutorial/overview papers:

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

The Emerging Field of Signal Processing on Graphs

Adaptation and Learning over Complex Networks

BETHANY COOK/NATIONAL INSTITUTE OF NEUROLOGICAL DISORDERS AND STROKE

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

In applications such as social, energy, transportation, sensor, and neuronal networks, high-dimensional data naturally reside on the vertices of weighted graphs. The emerging field of signal processing on graphs merges algebraic and spectral graph theoretic concepts with computational harmonic analysis to process such signals on graphs. In this tutorial overview, we outline the main challenges of the area, discuss different ways to define graph spectral domains, which are the analogs to the classical frequency domain, and highlight the importance of incorporating the irregular structures of graph data domains when processing signals on graphs. We then review methods to generalize fundamental operations such as filtering, translation, modulation, dilation, and down sampling to the graph setting and survey the localized, multiscale transforms that have

been proposed to efficiently extract information from high-dimensional data on graphs. We conclude with a brief discussion of open issues and possible extensions.

INTRODUCTION

Graphs are generic data representation forms that are useful for describing the geometric structures of data domains in numerous applications, including social, energy, transportation, sensor, and neuronal networks. The weight associated with each edge in the graph often represents the similarity between the two vertices it connects. The connectivities and edge weights are either dictated by the physics of the problem at hand or inferred from the data. For instance, the edge weight may be inversely proportional to the physical distance between nodes in the network. The data on the graphs can be visualized as a finite collection of samples, with one sample at each vertex in the graph. Collectively, we refer to these

Digital Object Identifier 10.1109/MSP.2012.2235503
Date of publication: 5 April 2013

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 ORTEGA¹, Fellow IEEE, PASCAL FROSSARD, Fellow IEEE, JELENA KOVACEVIC, Fellow IEEE, José M. F. MOURA², Fellow IEEE, and PIERRE VANDERHEYDT

ABSTRACT | Research in graph signal processing (GSP) aims to develop tools for processing data defined on irregular graph domains. In this paper, we first provide an overview of core ideas in GSP and their connection to conventional digital signal processing, along with a brief historical perspective to highlight how concepts recently developed in GSP build on top of prior research in other areas. We then summarize recent advances in developing basic GSP tools, including methods for sampling, filtering, or graph learning. Next, we review progress in several application areas using GSP, including processing and analysis of sensor network data, biological data, and applications to image processing and machine learning.

KEYWORDS | Graph signal processing (GSP); network science and graphs; sampling; signal processing

I. INTRODUCTION AND MOTIVATION

Data is all around us, and massive amounts of it. Almost every aspect of human life is now being recorded at all levels: from the marking and recording of processing inside the cells starting with the advent of fluorescent markers, to our personal data through health monitoring devices and apps, financial and banking data, our social networks, mobility and traffic patterns, marketing preferences, fads, and many more. The complexity of such networks [1] and interactions mean that the data now reside on irregular and complex structures that do not lend themselves to standard tools.

Graphs offer the ability to model such data and complex interactions among them. For example, users on Twitter can be modeled as nodes while their friend connections can be modeled as edges. This paper explores adding attributes to such nodes and modeling those as signals on a graph; for example, year of graduation in a social network, temperature in a given city on a given day in a weather network, etc. Doing so requires us to extend classical signal processing concepts and tools such as Fourier transform, filtering, and frequency response to data residing on graphs. It also leads us to tackle complex tasks such as sampling in a principled way. The field that gathers all these questions is a communication of graph signal processing (GSP) [2], [3].

While the precise definition of a “graph signal” will be given later in the paper, let us start by noting that a graph signal is a set of values residing on a set of nodes. These nodes are connected via (possibly weighted) edges. As in classical signal processing, such signals can stem from a variety of domains; while in classical signal processing, however, the underlying graph can set a fair amount about those signals through their structure. Different types of graphs model different types of networks that these nodes represent.

Typical data include Erdős-Rényi graphs, ring graphs, random geometric graphs, small-world graphs, power-law graphs, nearest-neighbor graphs, scale-free graphs, and many others. These model networks with random connections (Erdős-Rényi graphs), networks of brain neurons (small-world graphs), social networks (scale-free graphs), and others.

In a classical signal processing, graph signals can have properties such as smoothness, that need to be appropriately defined. They can also be represented via basic atoms and can have a spectral representation. In particular, the graph Fourier transform allows us to develop the intuition gathered in the classical setting and extend it to graphs; we can talk about the notions of frequency and bandlimitedness,

Manuscript received November 25, 2017; revised March 10, 2018; accepted March 23, 2018. Date of current version April 24, 2018. (Corresponding author: Antonio Ortega.)
1A. Ortega is with the University of Southern California, Los Angeles, CA 90089, USA.
2E-mail: amf@cs.cmu.edu
P. Frossard, and P. Vandergheynst are with EPFL, Lausanne, Switzerland 1015, Lausanne, Switzerland.
J. Kovacevic, and J. M. F. Moura are with Carnegie Mellon University, Pittsburgh, PA 15213, USA.
Digital Object Identifier 10.1109/JPROC.2018.2830246

0018-9219 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications/rights/index.html for more information.

- More available at: <http://web.media.mit.edu/~xdong/resource.html>