Graph Signal Processing summary

N. Kilis, Prof. Ioannis Pitas,
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 3.4.3

Artificial Intelligence & Information Analysis Lab

Outline

- Linear 1D convolution
- Cyclic 1D convolution
- Graph Basics
- Graph Matrix Representations
- Graph Fourier-like Basis
- Graph Signals
- Graph Signal Diffusion
- Spatial Graph Convolution
- Generalizing Convolutions to Graphs
- Spectral Graph Convolution
- Filtering
  - Spatial domain
  - Spectral domain
- Spatial – Spectral connection
- Graph Signal Sampling
- Graph Signals and Stationarity

Linear 1D convolution

The one-dimensional (linear) convolution of:

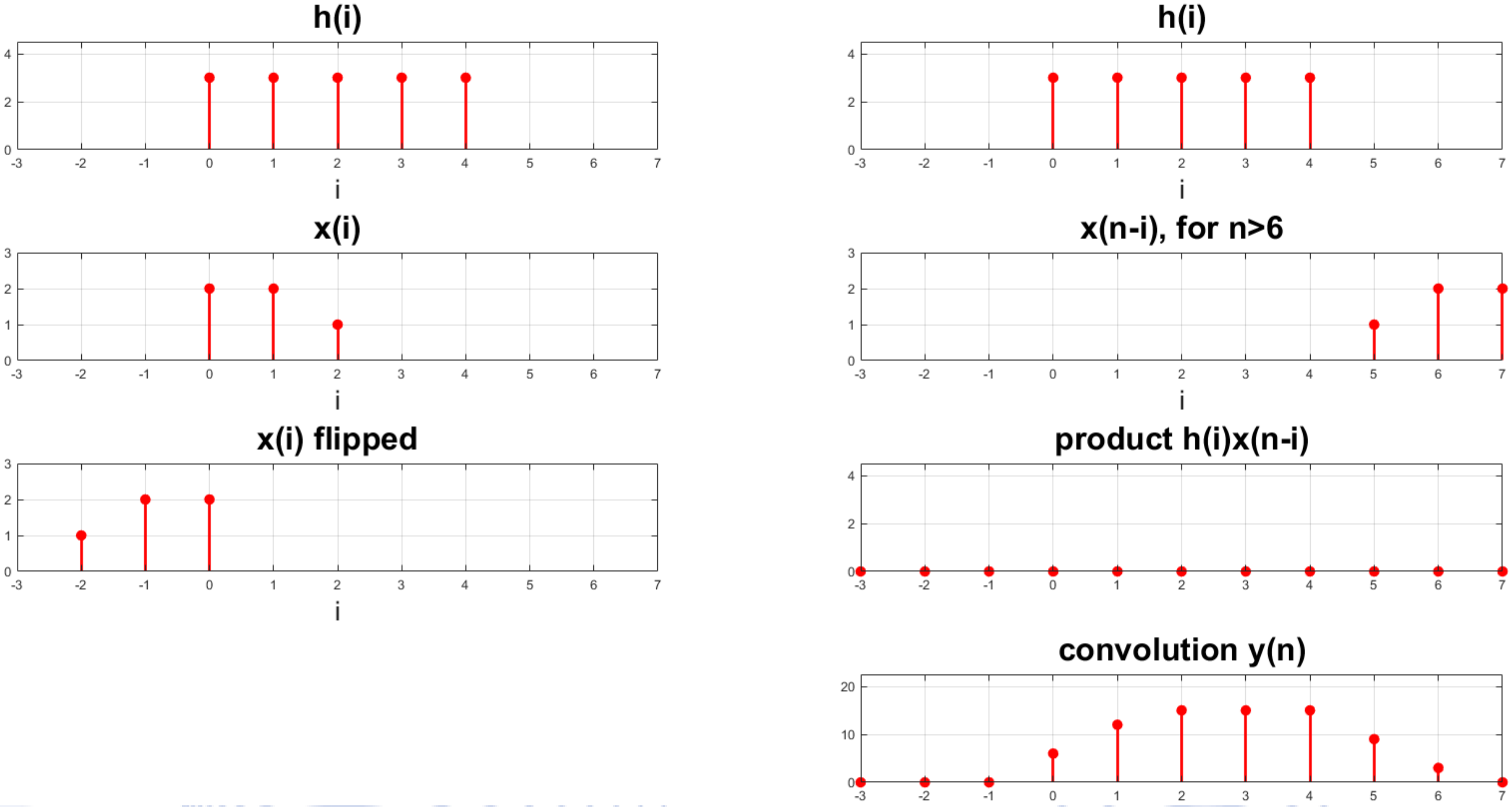
- an input signal x of length L and
- a convolution kernel h (filter mask, finite impulse response) of length M is defined as:

$$y(n) = h(n) * x(n) \triangleq \sum_{i=0}^{M-1} h(i)x(n-i).$$

- For a convolution kernel centered around 0 and $M = 2v + 1$, convolution takes the form:

$$y(n) = h(n) * x(n) = \sum_{i=-v}^{v} h(i)x(n-i).$$

Linear 1D convolution



Cyclic 1D convolution

- One-dimensional cyclic convolution of length N :

$$y(k) = x(k) \circledast h(k) = \sum_{i=0}^{N-1} h(i)x\big(((k-i)_N)\big),$$

$$(k)_N = k \mod N.$$

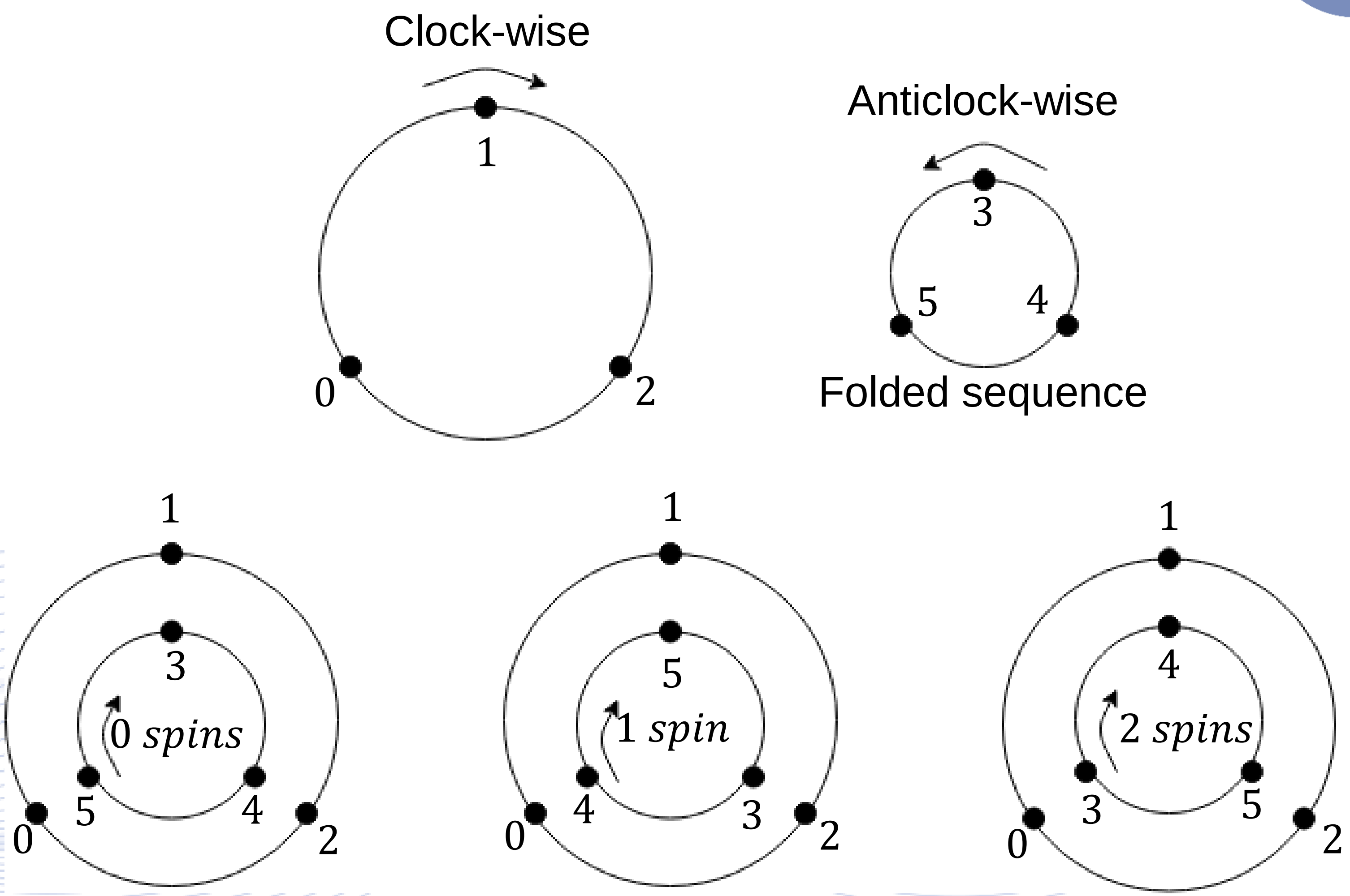
- It is of no use in modeling linear systems.
- Important use: Embedding linear convolution in a **fast** cyclic convolution $y(n) = x(n) \circledast h(n)$ of length $N \geq L + M - 1$ and then performing a cyclic convolution of length N.

Cyclic 1D convolution

$$y(0) = 1 \times 3 + 2 \times 4 + 0 \times 5$$

$$y(1) = 1 \times 5 + 2 \times 3 + 0 \times 4$$

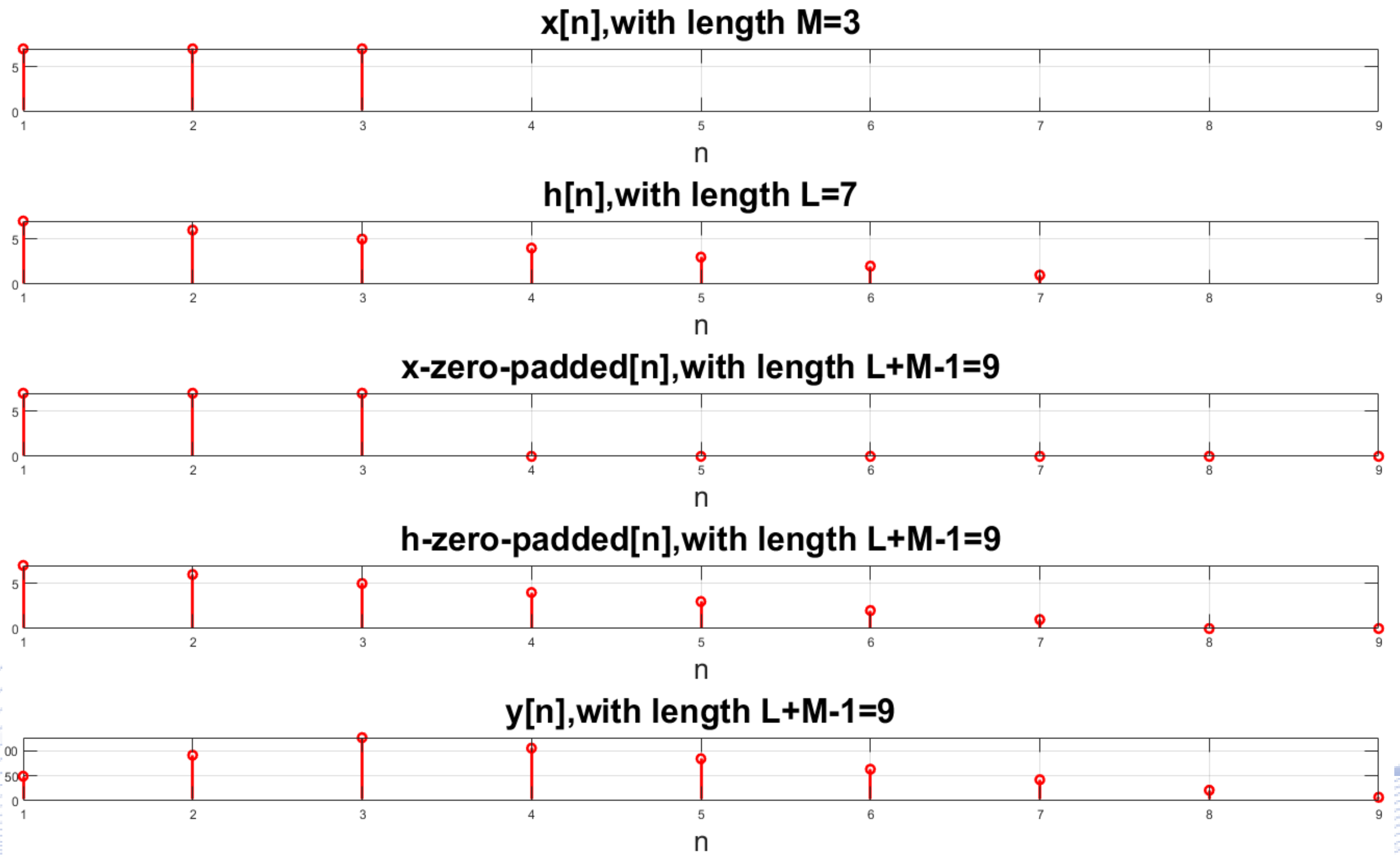
$$y(2) = 1 \times 4 + 2 \times 5 + 0 \times 3$$



Cyclic convolution of $x(n) = \{1, 2, 0\}$ and $h(n) = \{3, 5, 4\}$.

Artificial Intelligence & Information Analysis Lab

Cyclic 1D convolution

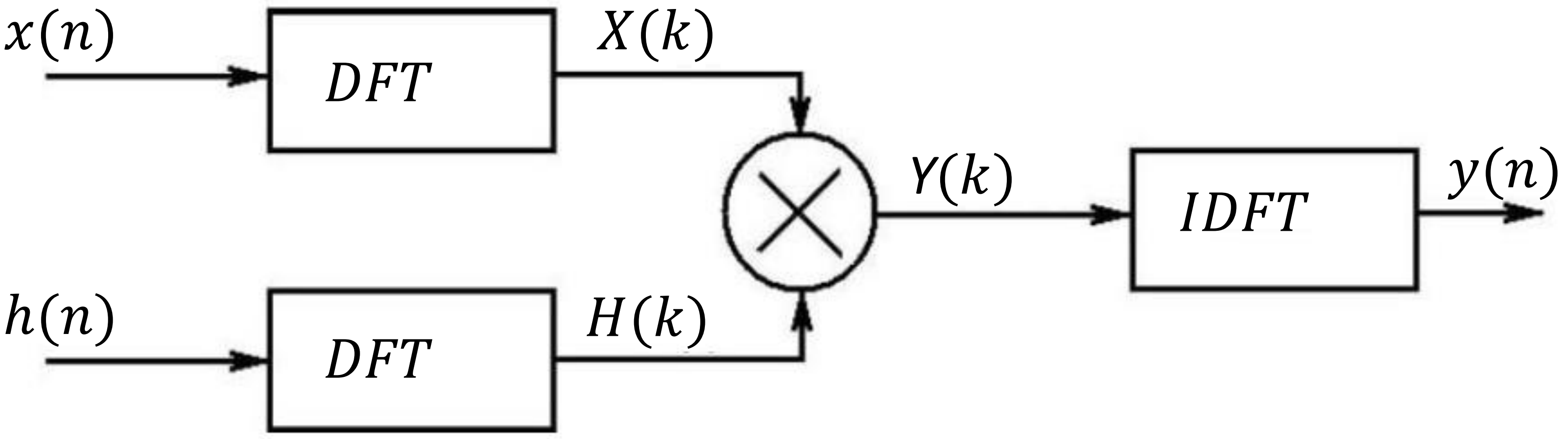


Zero-padding.

Cyclic 1D convolution

- Cyclic convolution calculation using 1D ***Discrete Fourier Transform*** (***DFT***):

$$\mathbf{y} = IDFT\big(DFT(\mathbf{x}) \otimes DFT(\mathbf{h})\big).$$

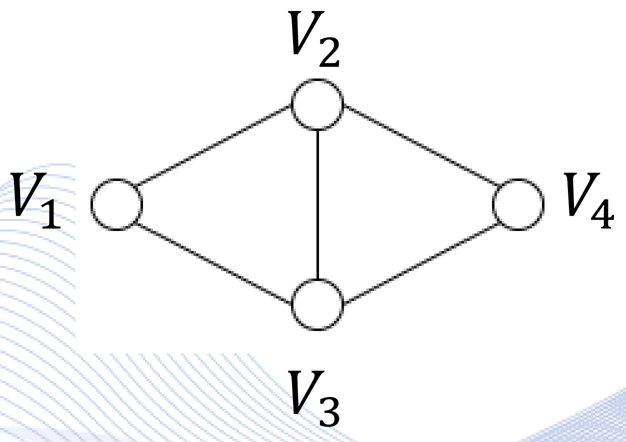


- Fast calculation of DFT, IDFT through ***FFT algorithm***.

Graph Basics

Graph definition: $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$

- $\mathcal{V}$: set of nodes,
- $\mathcal{E}$: set of edges,
- $\mathcal{W}$: set of edge weights.
- N: number of nodes
- E: number of edges



Graph types:

- Directed / Undirected or Symmetric,
- Weighted / Unweighted.

Artificial Intelligence & Information Analysis Lab

Graph Matrix Representations

Graph-Shift Operator (***GSO***):

$$\boldsymbol{S} \in \mathbb{R}^{N \times N}, \qquad S_{ij} \neq 0 \text{ if } i = j \text{ and/or } (i,j) \in \mathcal{E}.$$

- It enables matrix representations of graphs.
- It captures the local graph structure.
- If the graph is symmetric, $\boldsymbol{S}$ is also symmetric.

Graph Signals

- ***Vertex signal***:

$$x_i: \mathcal{V} \rightarrow \mathbb{R}.$$

- ***Vectorial vertex signal***:

$$\mathbf{x}_i: \mathcal{V} \rightarrow \mathbb{R}^n.$$

- ***Graph signals*** (single-valued vertex signals) can be described by a vector:

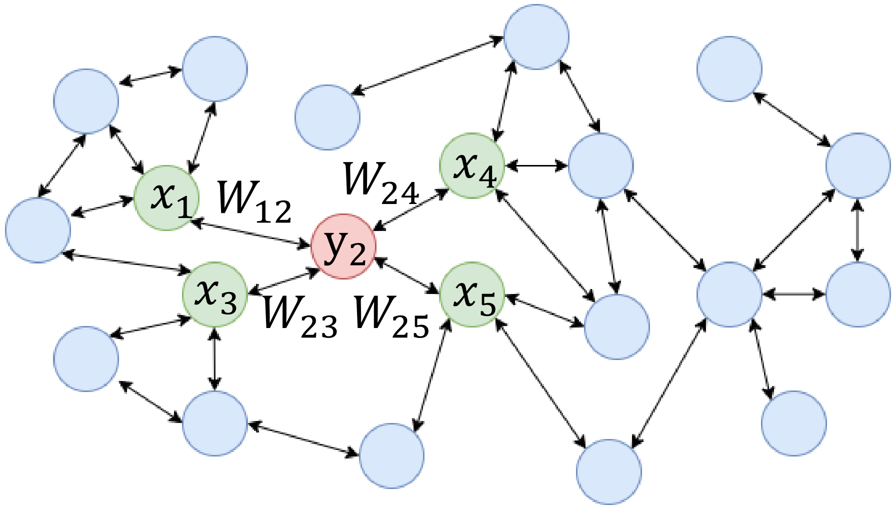
$$\mathbf{x} = [x_1, x_2, \dots, x_N]^T \in \mathbb{R}^N,$$

residing on the vertex set $\mathcal{V}$ of graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$.

Graph Signal Diffusion

- ***Diffusion*** of a Graph Signal: $\mathbf{y} = \boldsymbol{S}\mathbf{x}$.
  - Component i of $\mathbf{y}$ is affected by the set of nodes $j \in \mathcal{N}_i$ for $\boldsymbol{S} = \mathbf{W}$:

$$y_i = \sum_{j \in \mathcal{N}_i} W_{ij} x_j \,.$$



  - Stronger weights contribute more the diffusion.
  - Local operation where components are mixed with components of neighboring nodes.

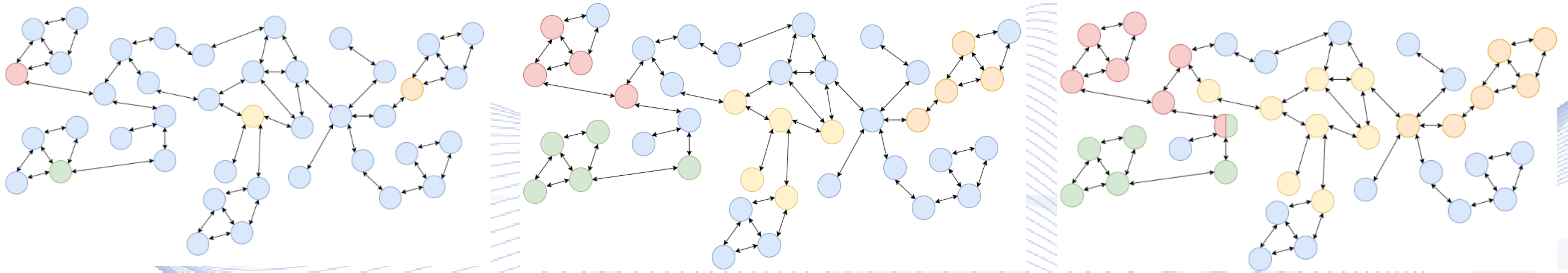
Graph Signal Diffusion

- ***Diffusion sequence*** → Recursive application of Diffusion:

$$\mathbf{x}^{(k+1)} = \mathbf{S}\mathbf{x}^{(k)},$$
$$\mathbf{x}^{(0)} = \mathbf{x}.$$

- We can also write the diffusion sequence as the power sequence:

$$\mathbf{x}^{(k)} = \mathbf{S}^{(k)}\mathbf{x}$$



$\mathbf{x}^{(0)} = \mathbf{x} = \mathbf{S}^{(0)}\mathbf{x}$ $\quad$ $\mathbf{x}^{(1)} = \mathbf{S}\mathbf{x}^{(0)} = \mathbf{S}^{(1)}\mathbf{x}$ $\quad$ $\mathbf{x}^{(2)} = \mathbf{S}\mathbf{x}^{(1)} = \mathbf{S}^{(2)}\,\mathbf{x}$

- Always implement the recursive version. Power version only for analysis.

Two ways to define Convolution

Spatial / Vertex domain:

- A Graph is a set of nodes connected by edges.
- We define how to aggregate the information of one node through its neighbors.
- ***Spatial Graph Convolution***.

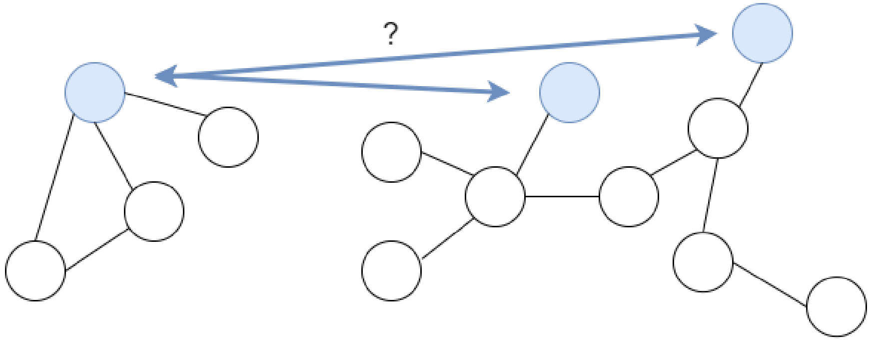
Spectral domain:

- A Graph is a discrete manifold [GEOM].
- Discretize manifold and do Spectral Convolution using the Laplacian matrix.
- ***Spectral Graph Convolution***.

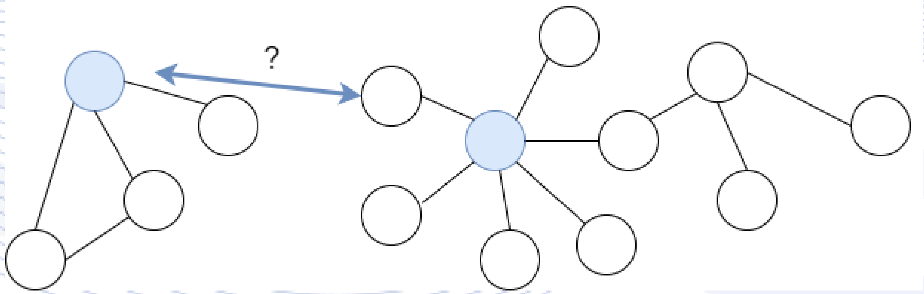
Spatial Graph Convolution

Limitations:

- ***Lack of node ordering***:
  - Can not match the template features with the data features.
  - The nodes do not have a well-defined position, but only an arbitrary index.



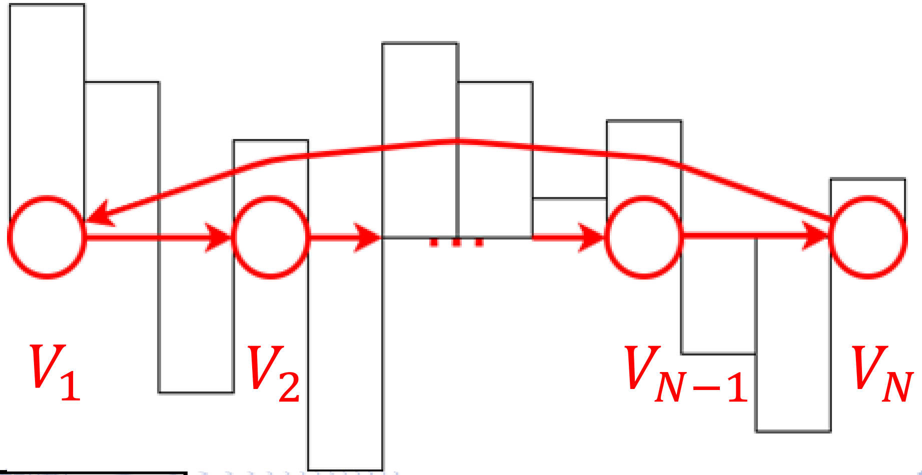
- ***Heterogeneous neighborhoods***:
  - Can not deal with nodes that have a different number of neighbors.



1-D Spatial Graph Convolution

- ***Graph-Shift matrix*** = ***Adjacency matrix*** of a graph:

$$\mathbf{S} = \mathbf{A} = \begin{array}{c c} \begin{array}{cccccc} 1 & 2 & 3 & \dots & \dots & N \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 & \dots & \dots & 1 \\ 1 & 0 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & \dots & 1 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ \dots \\ \dots \\ N \end{array} \end{array}$$



$$Time\ series \rightarrow \boxed{Shift\ operator/Adjacency\ matrix} \rightarrow Graph\ representation$$

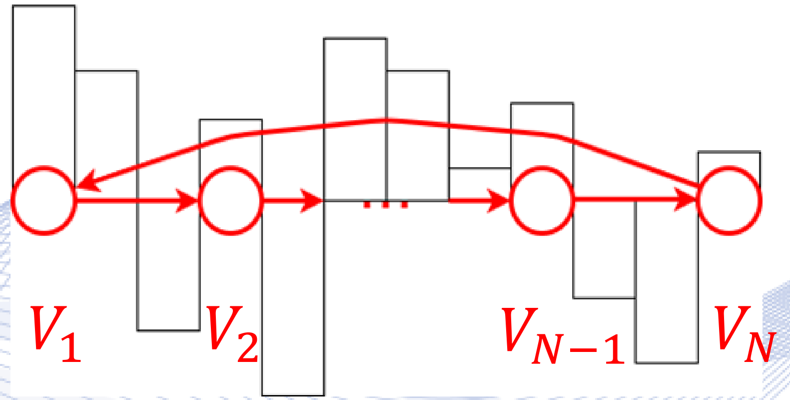
Graph:

- Nodes = time samples,
- (Directed) Edges = successive nature of time samples.

1-D Spatial Graph Convolution

- The ***DFT*** and the traditional frequency grid is obtained by the adjacency matrix of the ***circulant graph***:

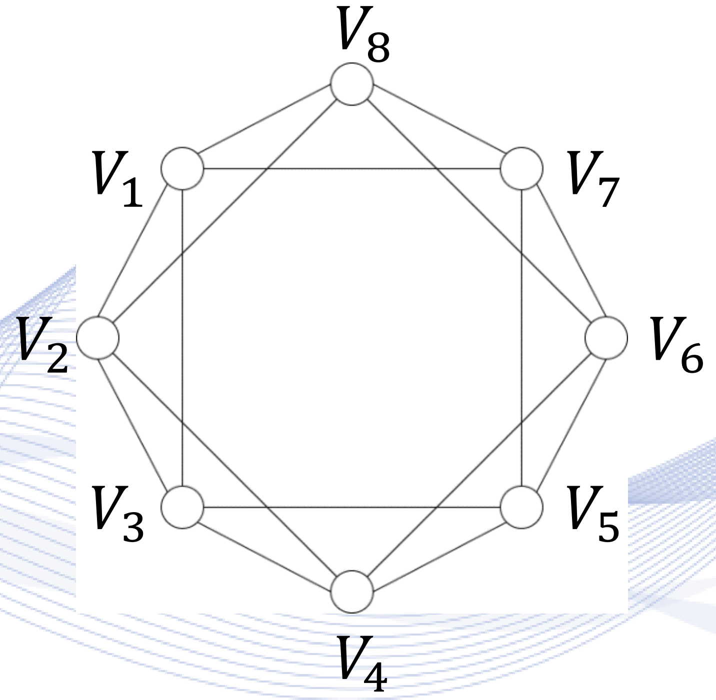
$$
\mathbf{S} = \mathbf{A} = \begin{array}{c@{}c}
\begin{array}{cccccc} 1 & 2 & 3 & \dots & \dots & N \end{array} & \\
\begin{bmatrix}
0 & 0 & 0 & \dots & \dots & 1 \\
1 & 0 & 0 & \dots & \dots & 0 \\
0 & 1 & 0 & \dots & \dots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \dots & 1 & 0 & 0 \\
0 & 0 & \dots & \dots & 1 & 0
\end{bmatrix} &
\begin{array}{c} 1 \\ 2 \\ 3 \\ \dots \\ \dots \\ N \end{array}
\end{array}
$$



1-D Spatial Graph Convolution

- Any circulant graph (directed or not) in principle leads to the DFT as the matrix that diagonalizes the Graph-Shift operator:

$$\mathbf{S} = \mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$



1-D Spatial Graph Convolution

Graph-Shift:

- local operation,
- we can replace nodes' signal values → weighted linear combination at neighbors $\mathcal{N}_i$:

$$x_{i,out} = \sum_{j \in \mathcal{N}_i} S_{ij} x_{j,in}, \qquad i = 1, \dots, N.$$

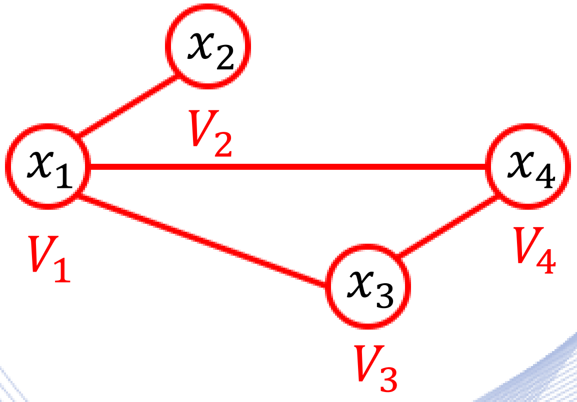
Graph Signal Processing (***GSP***) framework:

1. Signal shift operator
2. Graph theory ***adjacency matrix***

1-D Spatial Graph Convolution

- For an undirected Graph:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

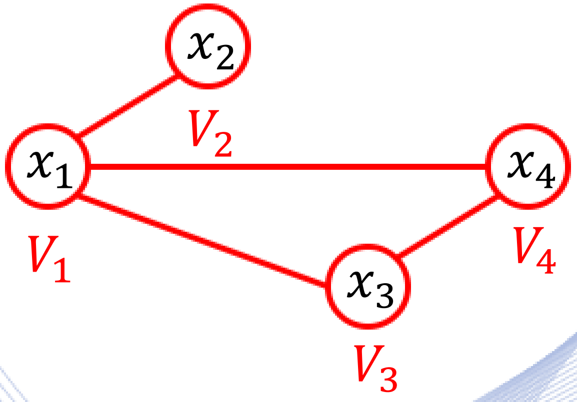


$$\mathbf{S} = \mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

1-D Spatial Graph Convolution

- Graph Shifted Signal:

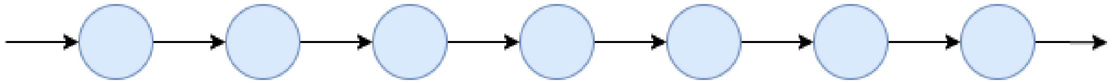
$$\mathbf{x}' = \mathbf{S}\mathbf{x} = \begin{bmatrix} x_2 + x_3 + x_4 \\ x_1 \\ x_1 + x_4 \\ x_1 + x_3 \end{bmatrix}$$



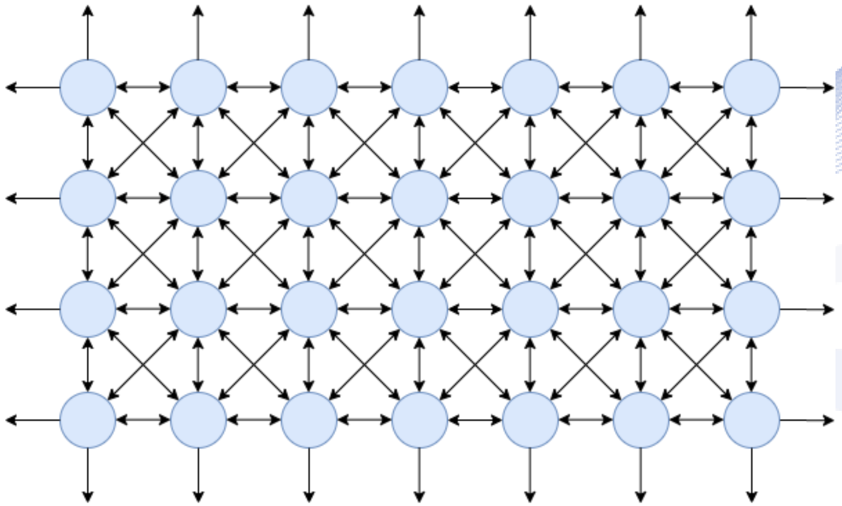
Generalizing Convolutions to Graphs

- Convolutions are operations on graphs.
- First two trivial observations of natural Graphs:

1. Discrete time:
   - Graphs that support time signals.
   - Directed Line Graph.

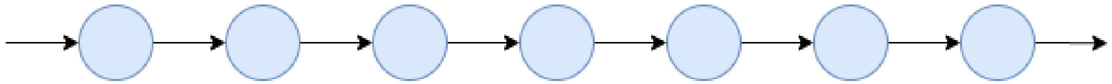


2. Discrete space:
   - Graphs that support space signals.
   - Grid Graph.



Generalizing Convolutions to Graphs

- Filter output:

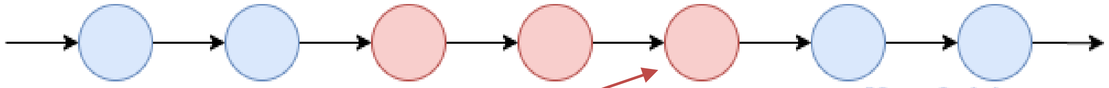


$$\mathbf{x}_{out} = w_0\mathbf{S}^0\mathbf{x}_{in} + w_1\mathbf{S}^1\mathbf{x}_{in} + w_2\mathbf{S}^2\mathbf{x}_{in} + \cdots = \sum_{k=0}^{K-1} w_k\mathbf{S}^k\mathbf{x}_{in} = \mathbf{H}(\mathbf{S})\,\mathbf{x}_{in},$$

$$\mathbf{H}(\mathbf{S}) \triangleq \sum_{k=0}^{K-1} w_k\mathbf{S}^k.$$

Generalizing Convolutions to Graphs

- Filter output:



$$\mathbf{x}_{out} = w_0\mathbf{S}^0\mathbf{x}_{in} + w_1\mathbf{S}^1\mathbf{x}_{in} + w_2\mathbf{S}^2\mathbf{x}_{in} + \cdots = \sum_{k=0}^{K-1} w_k\mathbf{S}^k\mathbf{x}_{in} = \mathbf{H}(\mathbf{S})\,\mathbf{x}_{in},$$

$$\mathbf{H}(\mathbf{S}) \triangleq \sum_{k=0}^{K-1} w_k\mathbf{S}^k.$$

Generalizing Convolutions to Graphs

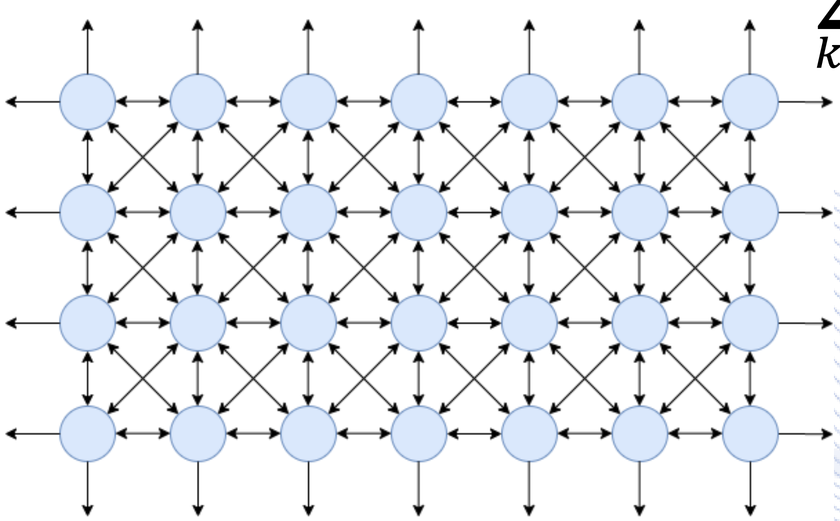
- The same holds for Grid Graphs.
- Discrete space:
  - Implementation of a convolutional filter with coefficients w_k and order K.
  - Linear combination of diffuse versions of the input signal.
  - Filter output:

$$\mathbf{x}_{out} = w_0\mathbf{S}^0\mathbf{x}_{in} + w_1\mathbf{S}^1\mathbf{x}_{in} + w_2\mathbf{S}^2\mathbf{x}_{in} + \cdots = \sum_{k=0}^{K-1} w_k\mathbf{S}^k\mathbf{x}_{in}$$

Generalizing Convolutions to Graphs

- Filter output:

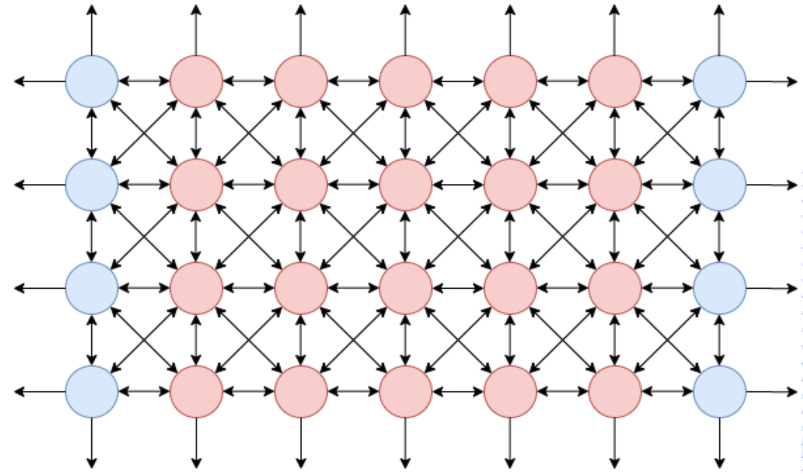
$$\mathbf{x}_{out} = w_0\mathbf{S}^0\mathbf{x}_{in} + w_1\mathbf{S}^1\mathbf{x}_{in} + w_2\mathbf{S}^2\mathbf{x}_{in} + \cdots = \sum_{k=0}^{K-1} w_k\mathbf{S}^k\mathbf{x}_{in}$$



Generalizing Convolutions to Graphs

- Filter output:

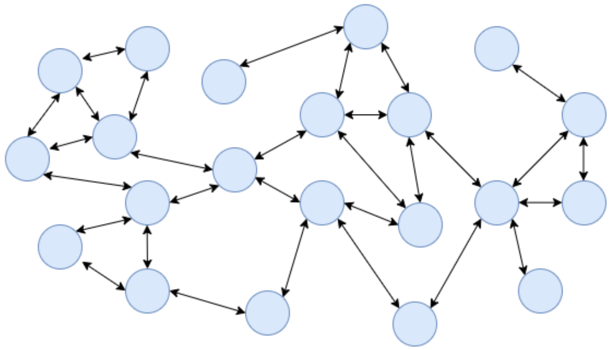
$$\mathbf{x}_{out} = w_0\mathbf{S}^0\mathbf{x}_{in} + w_1\mathbf{S}^1\mathbf{x}_{in} + w_2\mathbf{S}^2\mathbf{x}_{in} + \cdots = \sum_{k=0}^{K-1} w_k\mathbf{S}^k\mathbf{x}_{in}$$



Generalizing Convolutions to Graphs

- By analogy, the same holds for any Graph.
- Arbitrary Graph:
  - Implementation of a convolutional filter with coefficients w_k and order K.
  - Linear combination of diffuse versions of the input signal $\mathbf{x}_{in}$ scaled by w_k.
  - Filter output:

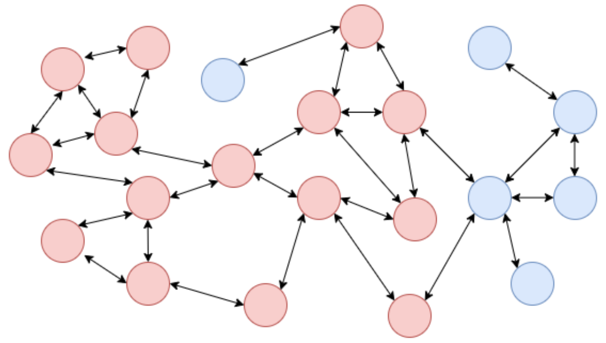
$$\mathbf{x}_{out} = w_0\mathbf{S}^0\mathbf{x}_{in} + w_1\mathbf{S}^1\mathbf{x}_{in} + w_2\mathbf{S}^2\mathbf{x}_{in} + \cdots = \sum_{k=0}^{K-1} w_k\mathbf{S}^k\mathbf{x}_{in}$$



Generalizing Convolutions to Graphs

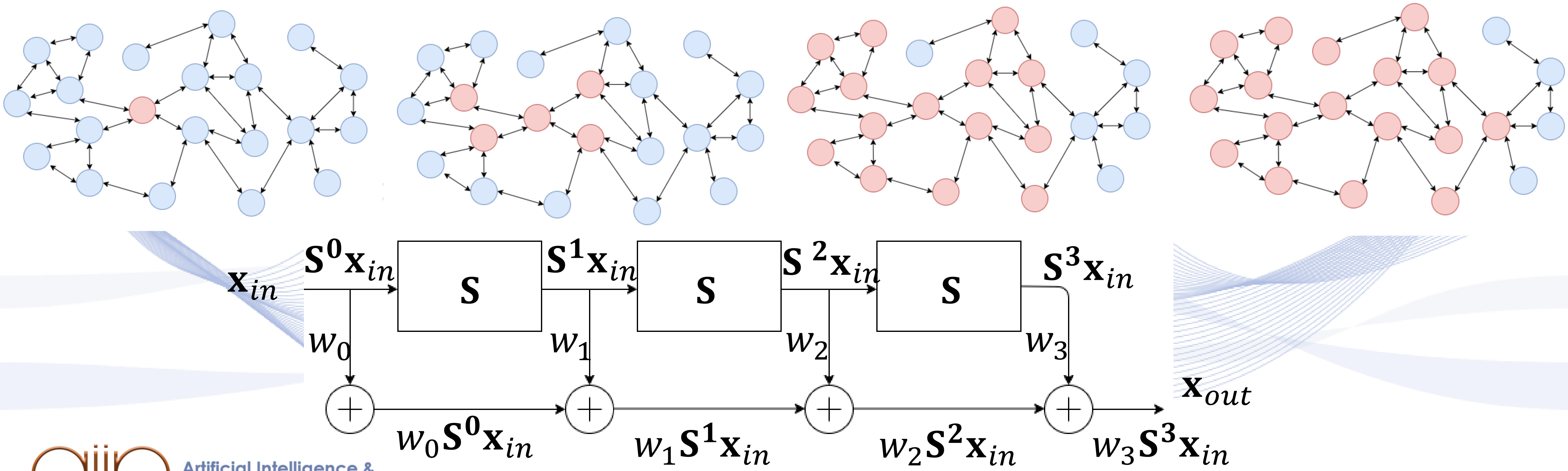
- By analogy, the same holds for any Graph.
- Arbitrary Graph:
  - Implementation of a convolutional filter with coefficients w_k and order K.
  - Linear combination of diffuse versions of the input signal $\mathbf{x}_{in}$ scaled by w_k.
  - Filter output:

$$\mathbf{x}_{out} = w_0\mathbf{S}^0\mathbf{x}_{in} + w_1\mathbf{S}^1\mathbf{x}_{in} + w_2\mathbf{S}^2\mathbf{x}_{in} + \cdots = \sum_{k=0}^{K-1} w_k\mathbf{S}^k\mathbf{x}_{in}$$



Generalizing Convolutions to Graphs

- Graph Convolutional filters perform linear processing of graph signals.



Spectral Graph Convolution

We perform Spectral Graph Convolution in 4 steps:

1. Graph Laplacian
2. Graph Fourier Functions
3. Graph Fourier Transform
4. Convolution Theorem

Graph Laplacian

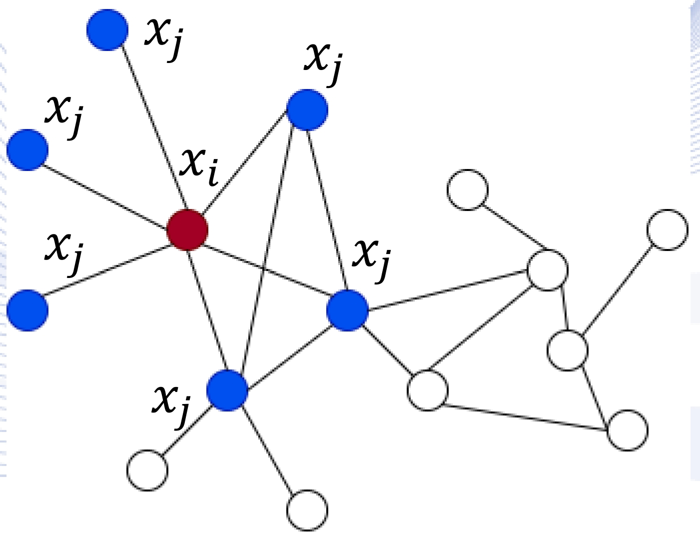
- Normalized Graph Laplacian is the main operator in Spectral Graph Theory:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$$

$$\mathbf{D} = \mathrm{diag}(\sum_{j \neq i} A_{ij})$$

- Physical interpretation:
  - ***Smoothness*** of a Graph Signal:
    - difference between x_i and node information in $\mathcal{N}_i$.
  - Smooth signal ⇒ small Graph Laplacian.

$$(\mathbf{Lx})_i = x_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} A_{ij} x_j \, .$$



Graph Fourier Functions

- Eigen-decomposition of Graph Laplacian:

$$\mathbf{L} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U}.$$

- The Laplacian eigenvectors are the ***Graph Fourier Functions***:

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N] \in \mathbb{R}^{N \times N}.$$

- ***Graph Fourier Functions***:
  - form an orthonormal basis:

$$\mathbf{U}^T \mathbf{U} = \mathbf{I},$$

  - are related to graph geometry (e.g., communities, hubs, etc.).

Graph Fourier Functions

- The Laplacian eigenvalues are known as the ***Graph spectrum***:

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \dots, \lambda_N) = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_N \end{bmatrix}$$

- For the normalized Graph Laplacian: $0 \leq \lambda_1 \leq \cdots \leq \lambda_N = \lambda_{max} \leq 2$.

Graph Fourier Transform

- ***Graph Fourier Transform (GFT)***:

$$\hat{\mathbf{x}} = \mathcal{F}\{\mathbf{x}\} \triangleq \mathbf{U}^T\mathbf{x}.$$

- Analysis of $\mathbf{x}$, with ***Graph Fourier Functions***, through the ***Graph Fourier Series***:

$$\mathbf{x} = \sum_{i=1}^{N} (\mathbf{x}^T\mathbf{u}_i)\mathbf{u}_i = \mathbf{U}\hat{\mathbf{x}}.$$

Convolution theorem

- The ***Graph Frequency Response*** is a polynomial of graph frequencies:

$$\hat{h}(\lambda) \triangleq \sum_{k=0}^{K-1} w_k \, \lambda^{\kappa}$$

$$\hat{h}_i = \hat{h}(\lambda_i), \qquad i = 1, \dots, N.$$

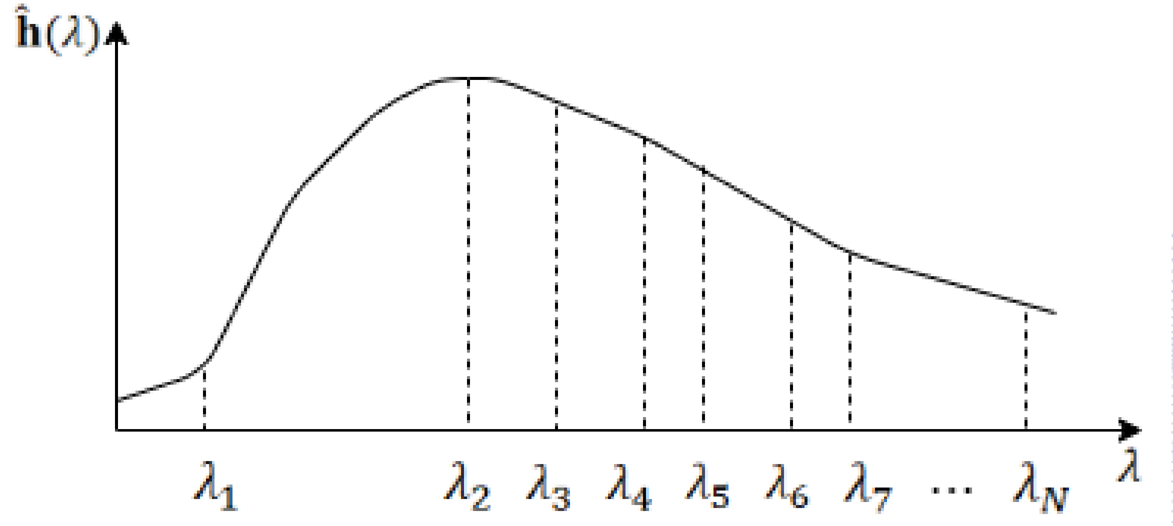
Artificial Intelligence &
Information Analysis Lab

Convolution theorem

- The spectral graph filter is given by:

$$\hat{\mathbf{h}} = \mathbf{U}^T\mathbf{h} = \left[\hat{h}(\lambda_1), \dots, \hat{h}(\lambda_N)\right]^T.$$

- Where $\lambda_1,\dots, \lambda_N$ are the eigenvalues.



Filtering – Spatial domain

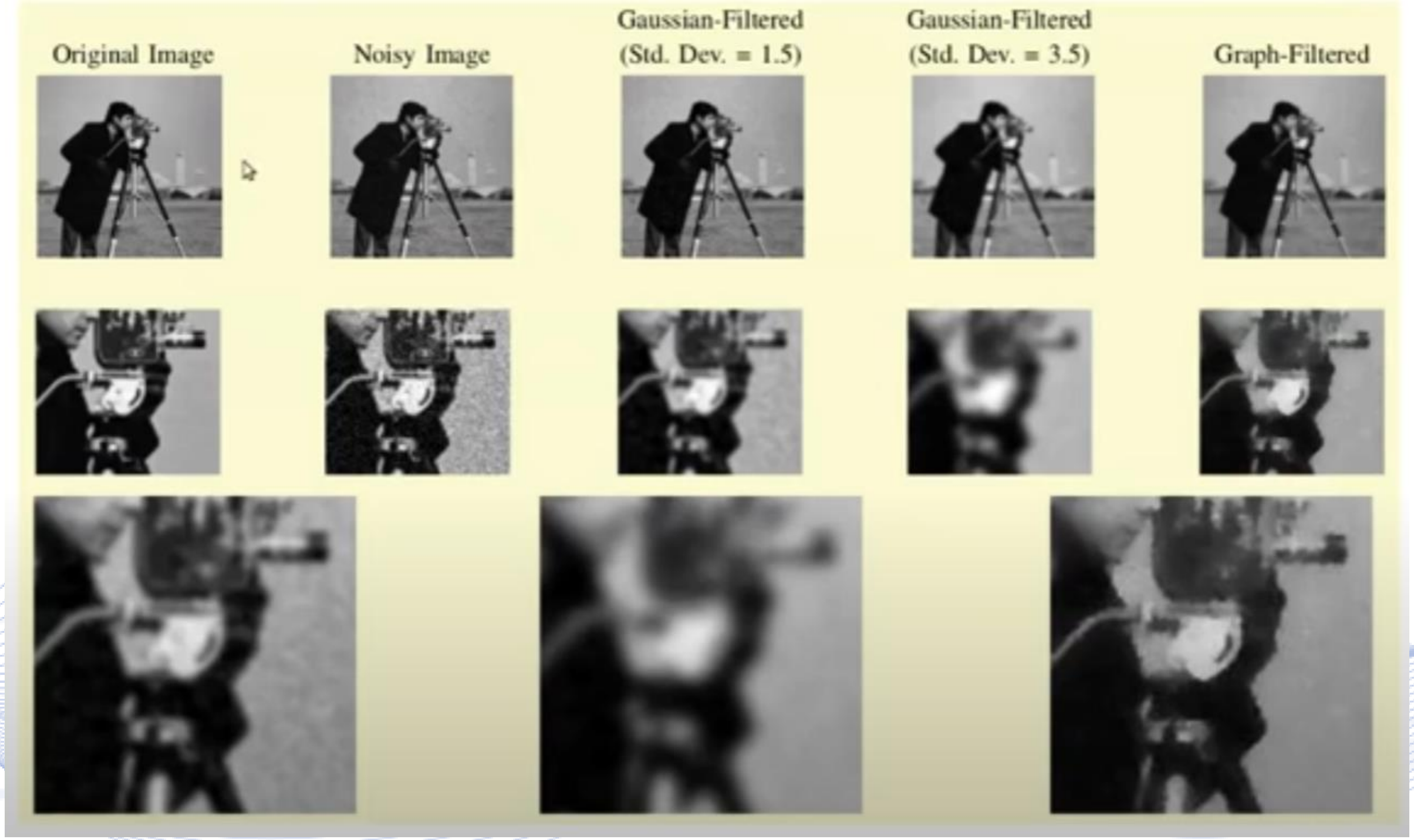
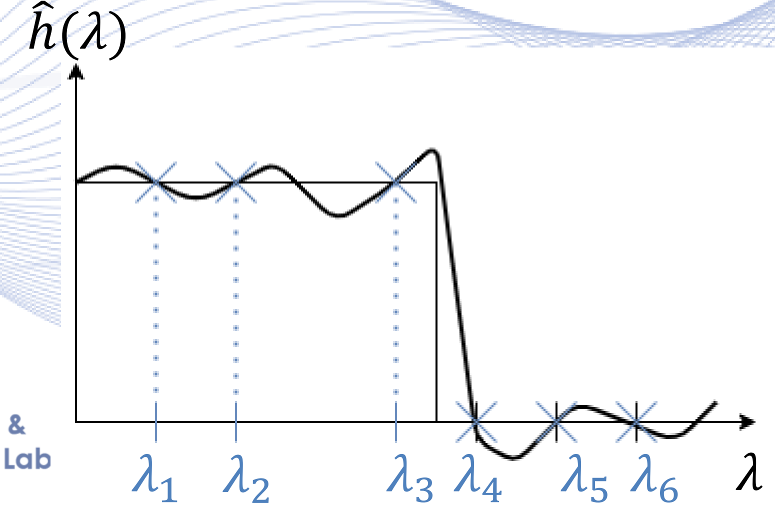
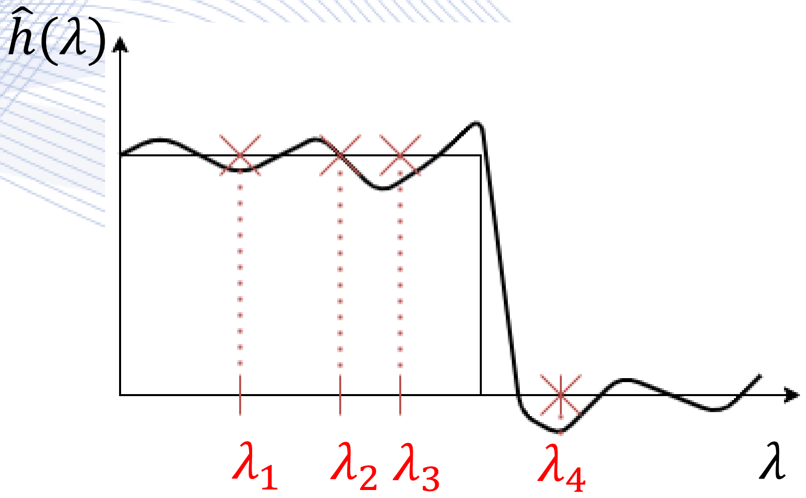


Image source [SHU2013].

Filtering – Spectral domain

- $\mathbf{H}(\boldsymbol{S})$ can be designed independently of the graph.
  - For every graph same characteristics, i.e. low-pass, over all possible graph frequencies $(\lambda_1, \dots, \lambda_N)$.
  - No perfect rectangular shape, but a polynomial curve is possible.
  - With the same filter coefficients:
    - magnify $\lambda_1, \lambda_2, \lambda_3$ and attenuate $\lambda_4, \lambda_5, \lambda_6$ for one graph with 6 nodes and
    - magnify $\lambda_1, \lambda_2, \lambda_3$ and attenuate λ_4 for another graph with 4 nodes.





Artificial Intelligence & Information Analysis Lab

FIR Graph Filters

- The ***Graph Frequency Response*** is a polynomial of graph frequencies:

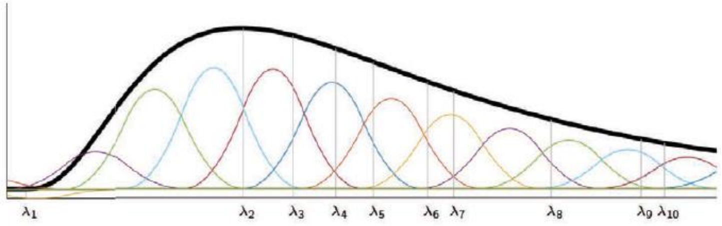
$$\hat{h}(\lambda) \triangleq \sum_{k=0}^{K-1} w_k \, \lambda^{\kappa}$$

$$\hat{h}_i = \hat{h}(\lambda_i), \qquad i = 1, \dots, N.$$

- Same polynomial that defines the Graph Filter (but on a scalar λ).
- Independent of the Graph (depends only on the filter coefficients).
- Role of Graph: determine the eigenvalues on which the response is instantiated.

Spatial – Spectral connection

- If smooth in Spectral domain:



- Then localized in Spatial domain:

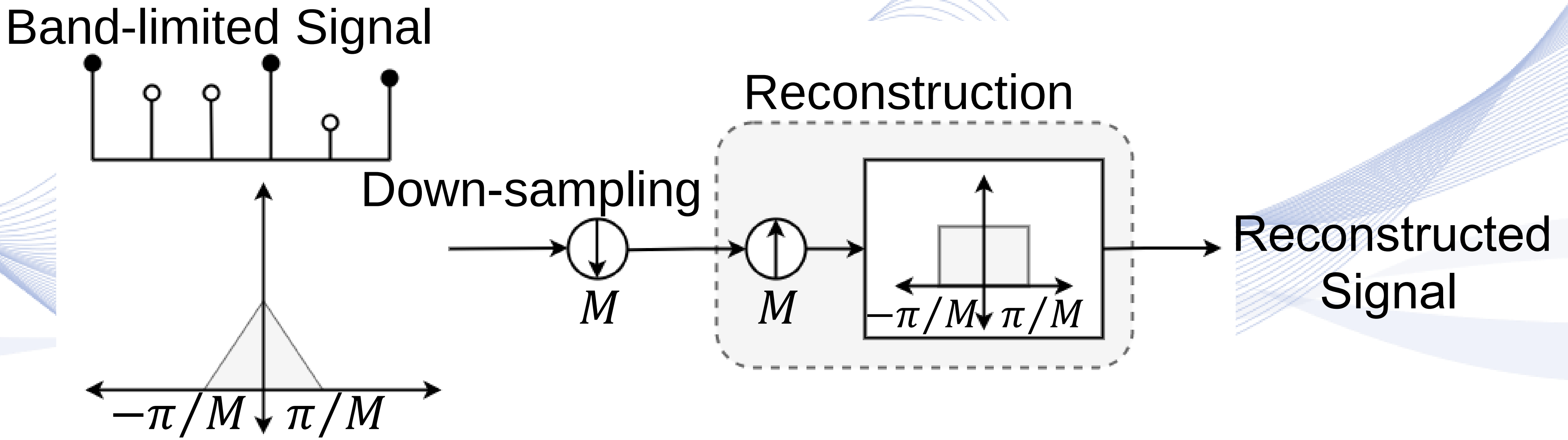
- Related publication [SHU2016].

Graph Signal Sampling

- Sample a signal at discrete points in time/space.
- Reconstruction: Ability to recover the original signal from the samples.
- Applications:
  - Sensor network (measure only a subset of sensors),
  - Social network (estimate interests from a subset of users).

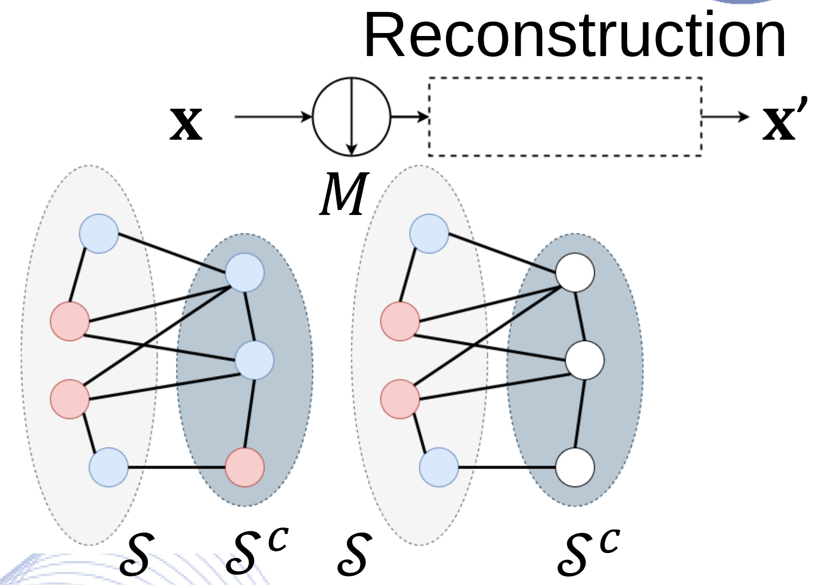
Graph Signal Sampling

- Traditional DSP:
  - How to sample? Regular sampling.
  - What properties enable recovery? Smooth signals, low frequency.
  - How to reconstruct? Low pass filtering.



Graph Signal Sampling

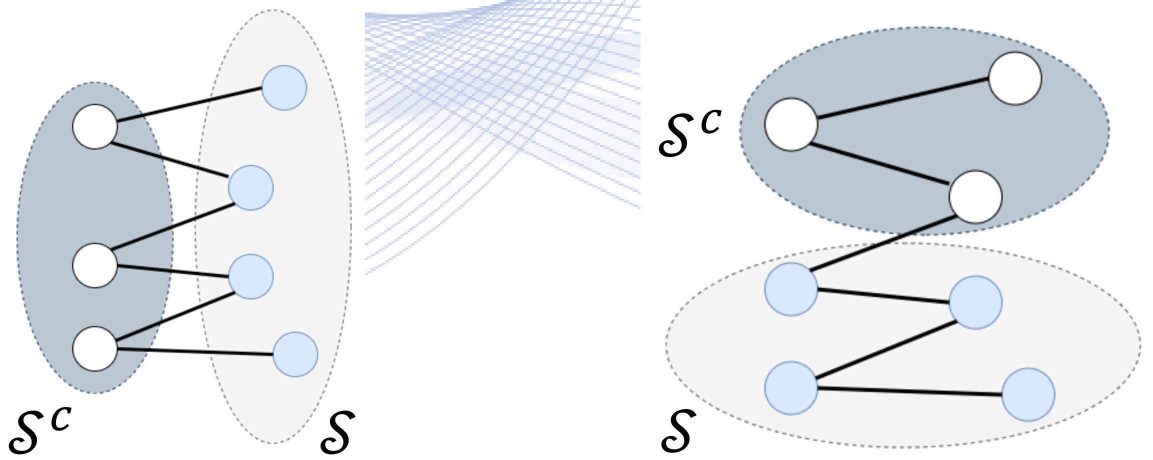
- In Graph Sampling:
  - Measure a subset ($\mathcal{S}$) of nodes,
  - Reconstruct the whole graph signal.



- How to sample? No obvious regular sampling (lack of node ordering).
- What properties enable recovery? Graph Frequency needed.
- How to reconstruct? Filtering needed.

Graph Signal Sampling

- Optimal set of labels to observe?
  - There are vertex and spectral domain solutions.
- Example method:
  - Minimize the distance from any node that you did not observe, to a node that you observed.
  - More cross-links ⇒ higher variation in lowest eigenvector of $\mathbf{L}\ (\mathcal{S}^c)$ ⇒ robust sampling.



Bibliography

[GEOM] http://geometricdeeplearning.com/ (Geometric Deep Learning on Graphs and Manifolds)

[SHU2013] D. I. Shuman, et al. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains." *IEEE signal processing magazine* 30.3 (2013): 83-98.

[SHU2016] D. I. Shuman, B. Ricaud, and P. Vandergheynst. "Vertex-frequency analysis on graphs." Applied and Computational Harmonic Analysis 40.2 (2016): 260-291.

[DEF2016] M. Defferrard, X. Bresson, and P. Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." arXiv preprint arXiv:1606.09375 (2016).

Bibliography

[PIT2016] I. Pitas (editor), “Graph analysis in social media”, CRC Press, 2016.

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.

[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).

[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.

[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.

[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr