جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Graph Neural Networks

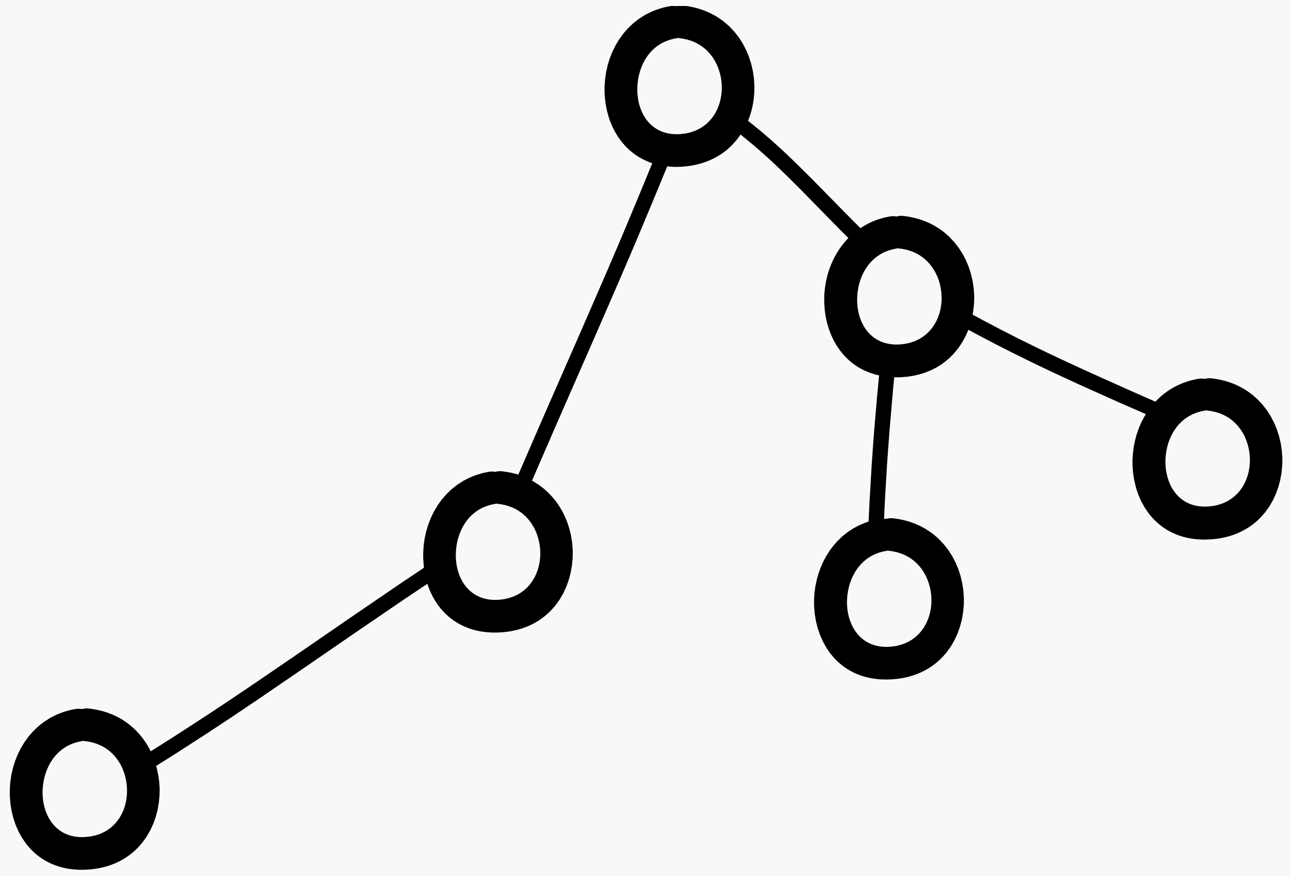
Daniel Alsadiq

21 July 2025

أكاديمية كاوست
KAUST ACADEMY

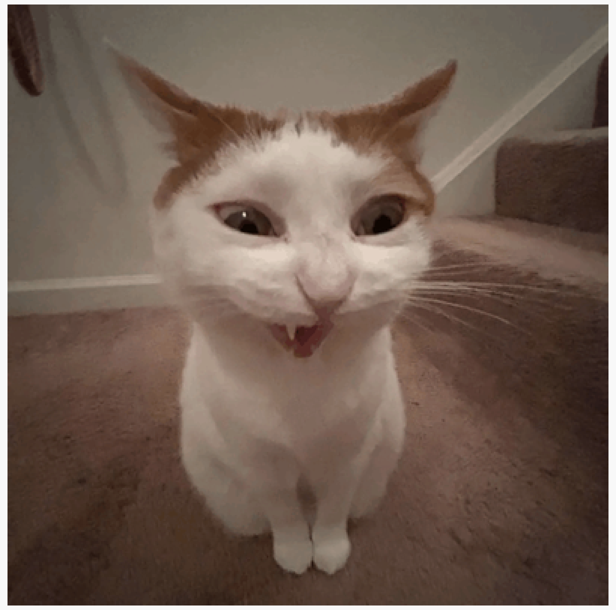
What are Graphs?

Graphs are a general language for describing and analyzing entities with relations/interactions



Why?

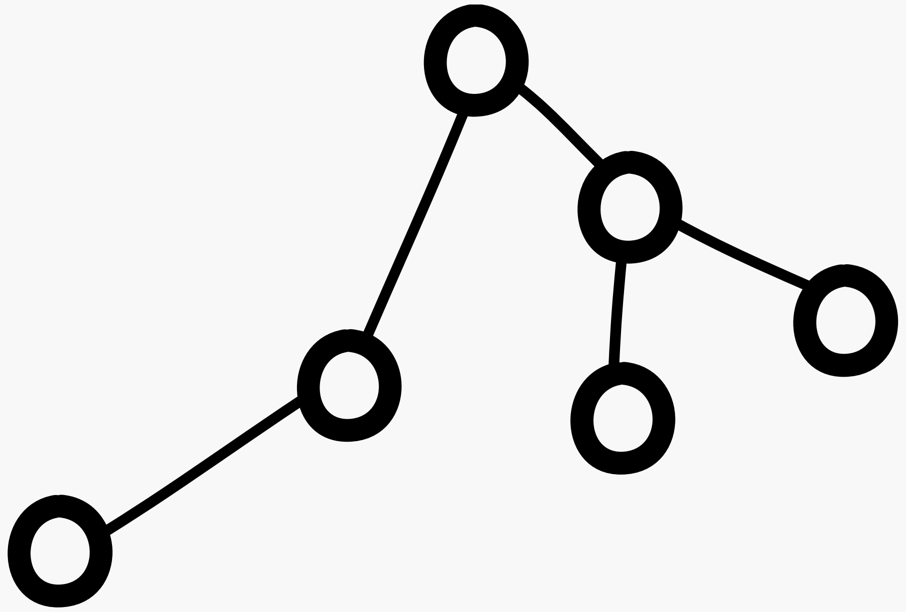
Real-world is messy — graphs offer natural representations



CNN

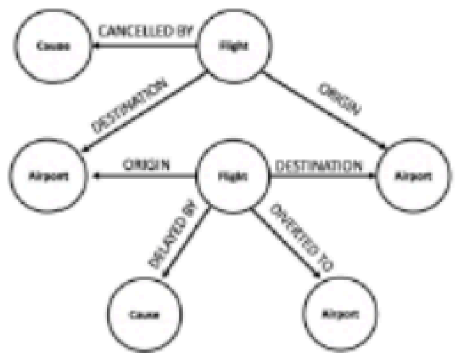
In one so bereft of light

RNN



???

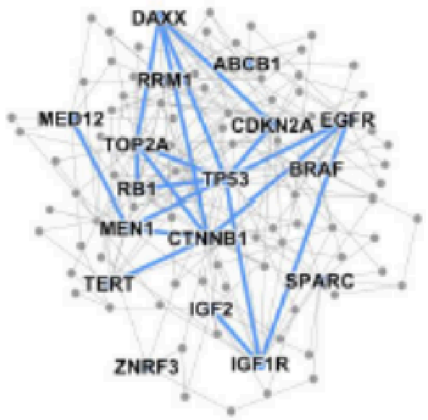
Examples



Event Graphs

Image credit: SalientNetworks

Computer Networks



Disease Pathways

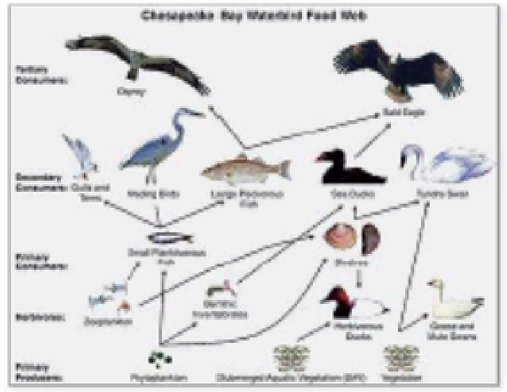


Image credit: Wikipedia

Food Webs

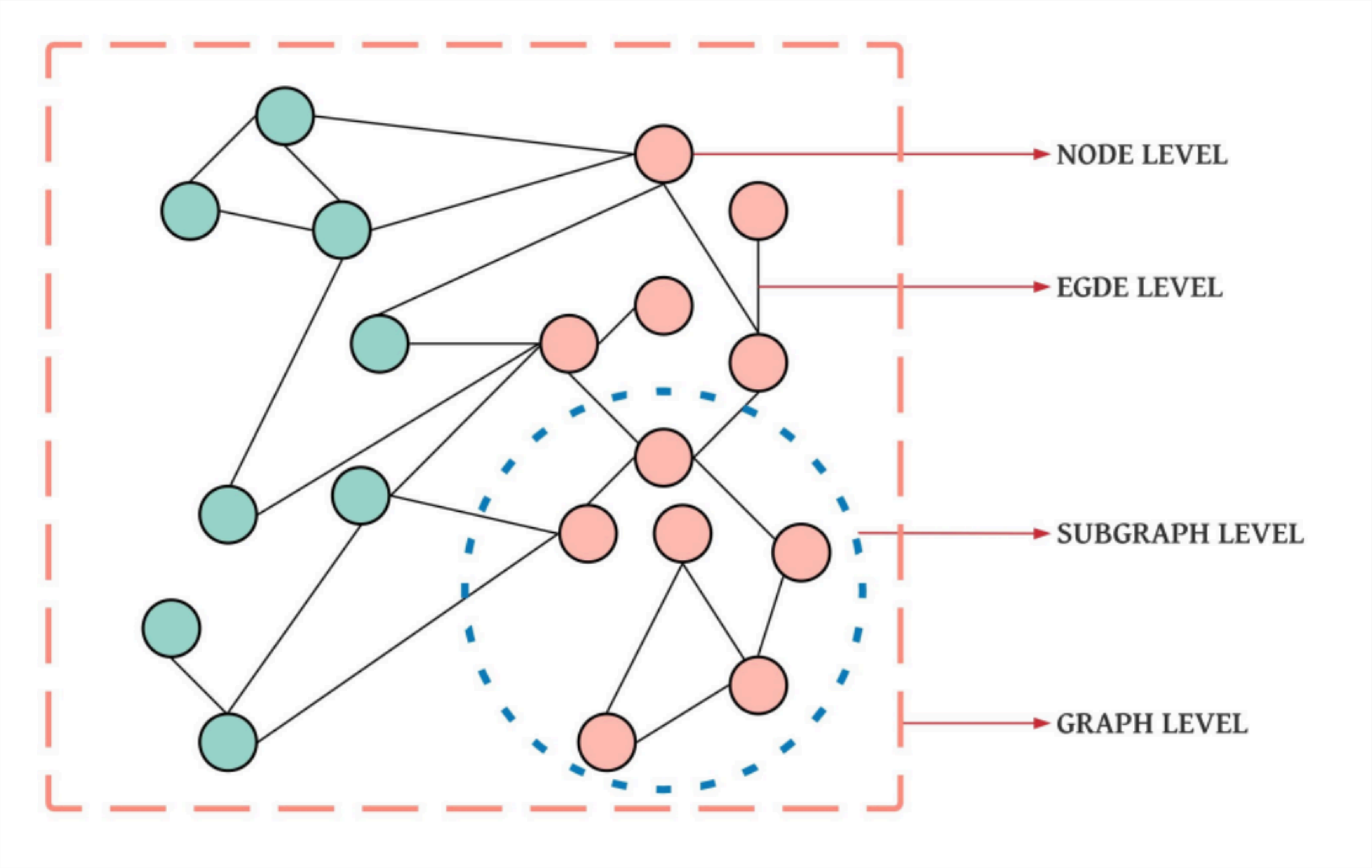
Image credit: Pinterest

Particle Networks

Image credit: visitlondon.com

Underground Networks

What Can We Do With Graphs?




AlphaFold 3
Google

Formal Definition

A graph is a triplet $G = (V, E, W)$, which includes:

Vertices (or nodes): a set of n labels representing the entities in the graph.

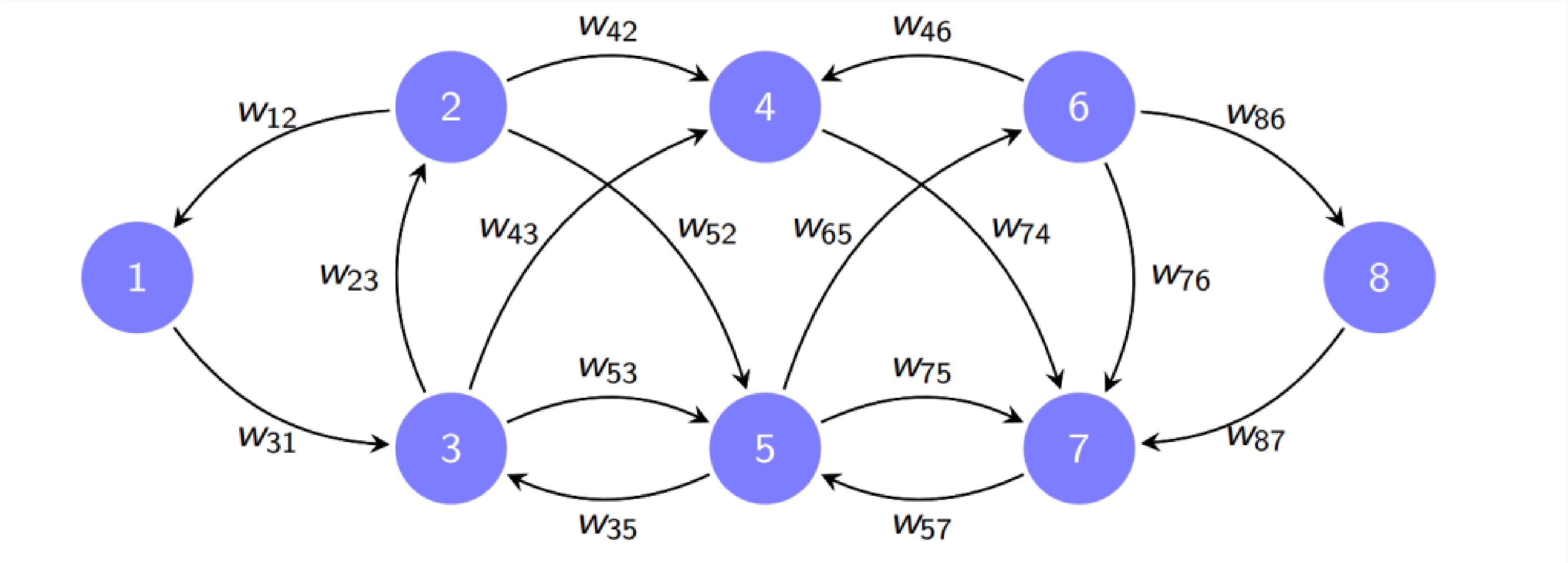
$$V = \{1, 2, \dots, n\}$$

Edges: ordered pairs (i, j), meaning that node i can be influenced by node j.

$$(i, j) \in E \Rightarrow \text{"node } i \text{ can be influenced by node } j\text{"}$$

Weights: $w_{ij} \in \mathbb{R}$, representing the strength of the influence of j on i.

$$w_{ij} \in \mathbb{R} \Rightarrow \text{"strength of the influence of } j \text{ on } i\text{"}$$



Adjacency

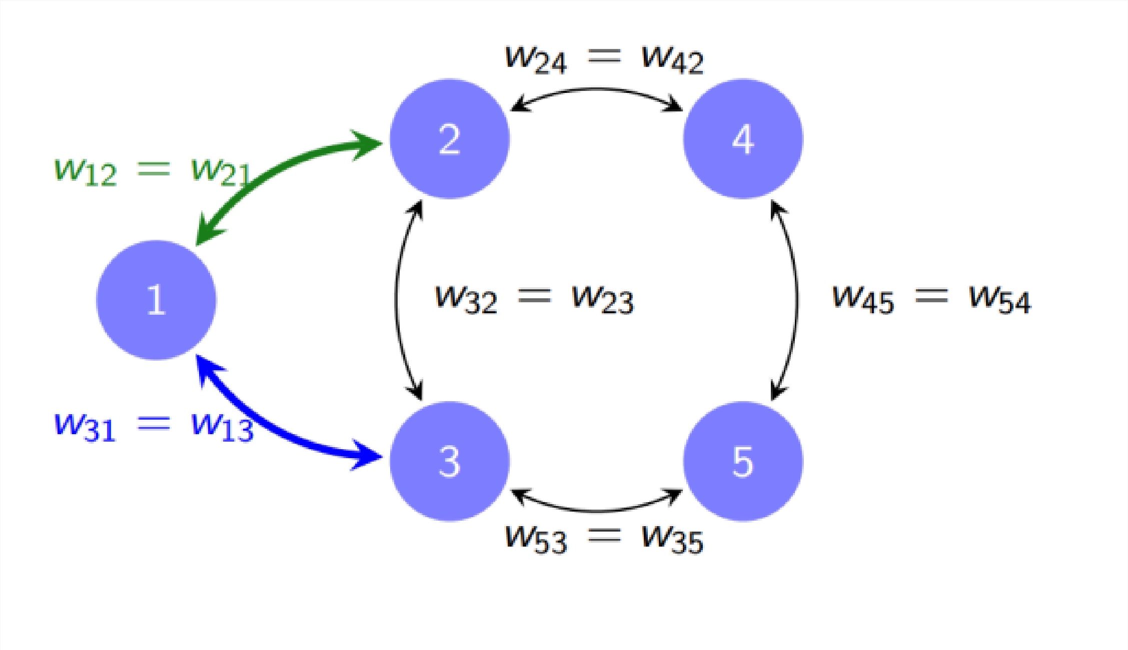
The adjacency matrix of a graph $G = (V, E, W)$ is the sparse matrix A with nonzero entries:

$$A_{ij} = w_{ij}, \quad \forall (i, j) \in E$$

If the graph is symmetric, the adjacency matrix is also symmetric:

$$A = A^T$$

Example of an adjacency matrix:



$$A = \begin{bmatrix} 0 & w_{12} & w_{13} & 0 & 0 \\ w_{21} & 0 & w_{23} & w_{24} & 0 \\ w_{31} & w_{32} & 0 & 0 & w_{35} \\ 0 & w_{42} & 0 & 0 & w_{45} \\ 0 & 0 & w_{53} & w_{54} & 0 \end{bmatrix}$$

Neighbourhood

The neighborhood of node i is the set of nodes that influence it:

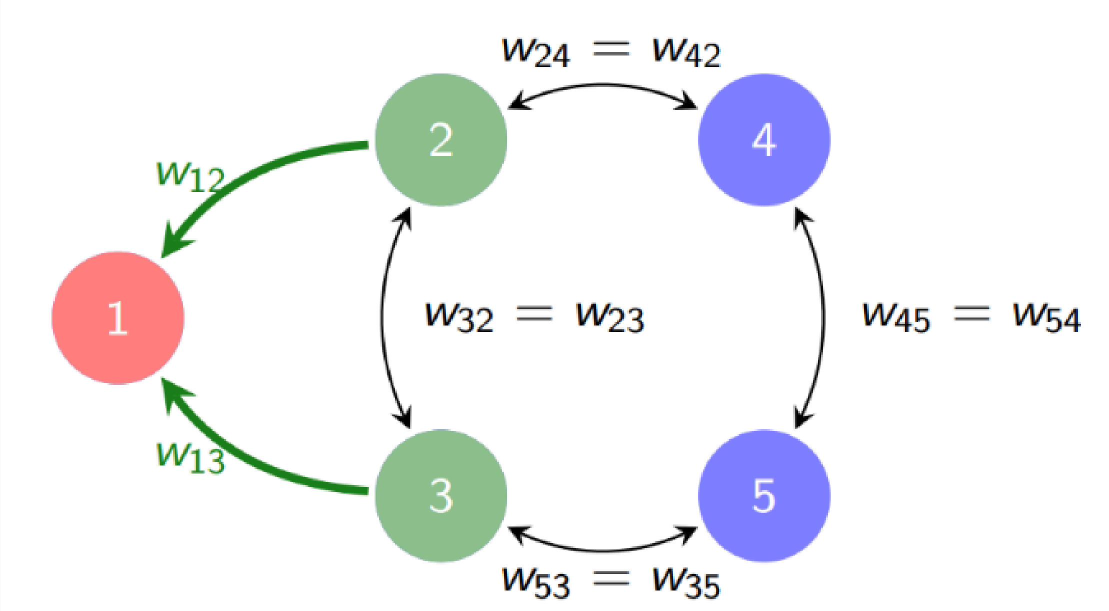
$$\Rightarrow \quad n(i) := \{j \mid (i,j) \in E\}$$

Example: Node 1 neighborhood $\Rightarrow n(1) = \{2, 3\}$

The degree d_i of node i is the sum of the weights of its incident edges:

$$\Rightarrow \quad d_i = \sum_{j \in n(i)} w_{ij} = \sum_{j:(i,j)\in E} w_{ij}$$

Example: Node 1 degree $\Rightarrow d_1 = w_{12} + w_{13}$



Degree Matrix

The degree matrix is a diagonal matrix D with degrees as diagonal entries:

$$\Rightarrow \quad D_{ii} = d_i$$

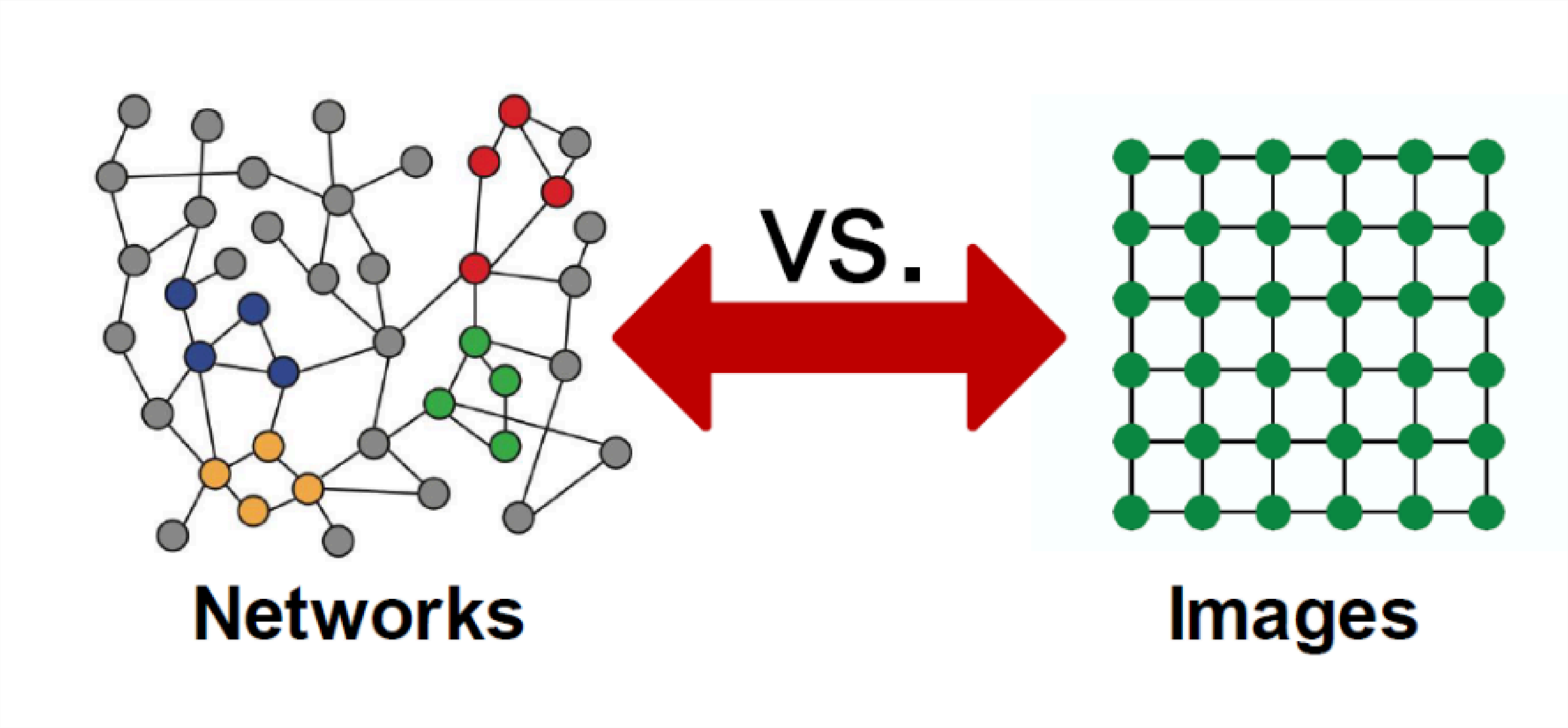
$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Laplacian Matrix

The Laplacian matrix of a graph with adjacency matrix A is defined as:

$$\Rightarrow \quad L = D - A = \mathrm{diag}(A\mathbf{1}) - A$$

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$


VS.
Networks
Images

Graph Shift Operator

The Graph Shift Operator S is a stand-in for any matrix representation of the graph.

Examples:

Adjacency Matrix: $S = A$

Laplacian Matrix: $S = L$

Normalized Adjacency Matrix: $S = \hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$

Normalized Laplacian Matrix: $S = \bar{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$

The specific choice of S matters in practice, but most results and analysis hold for any choice of S.

Graph Signal Diffusion

Multiplication by the graph shift operator implements diffusion of the signal over the graph.

Define diffused signal: $y = Sx$

Components: $y_i = \sum_{j \in \mathcal{N}(i)} w_{ij} x_j$

Stronger weights contribute more to the diffusion output.

This codifies a local operation where components are mixed with components of neighboring nodes.

Graph Signal Diffusion

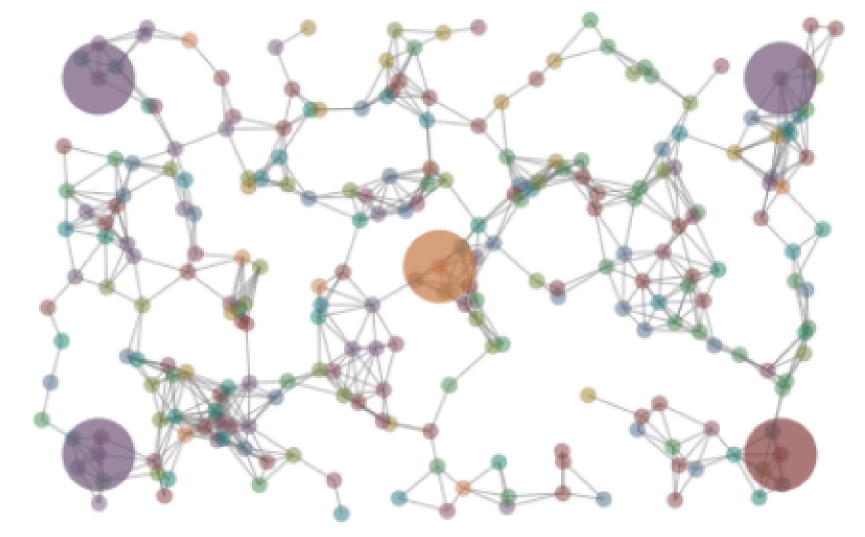
$$x^{(k+1)} = Sx^{(k)} \quad \text{with} \quad x^{(0)} = x$$

This can also be unrolled and written as a power sequence:

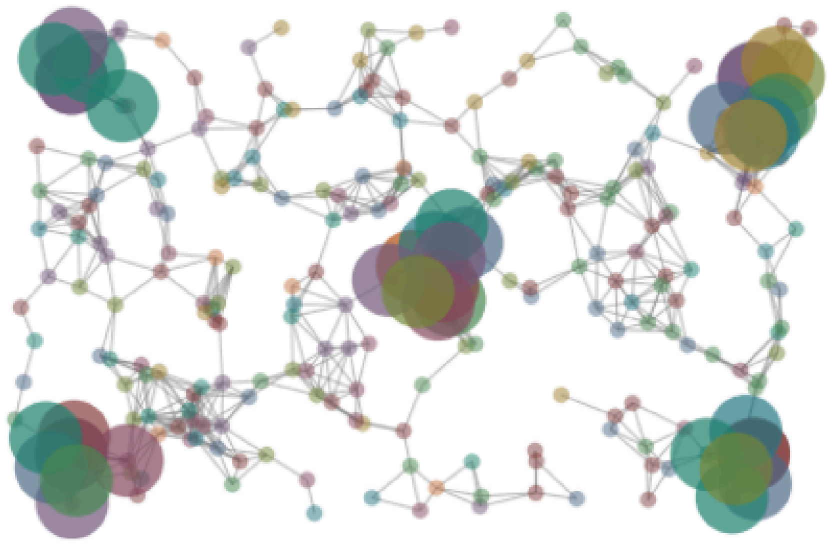
$$x^{(k)} = S^k x$$

The k^{th} element of the diffusion sequence $x^{(k)}$ diffuses information to k-hop neighborhoods.

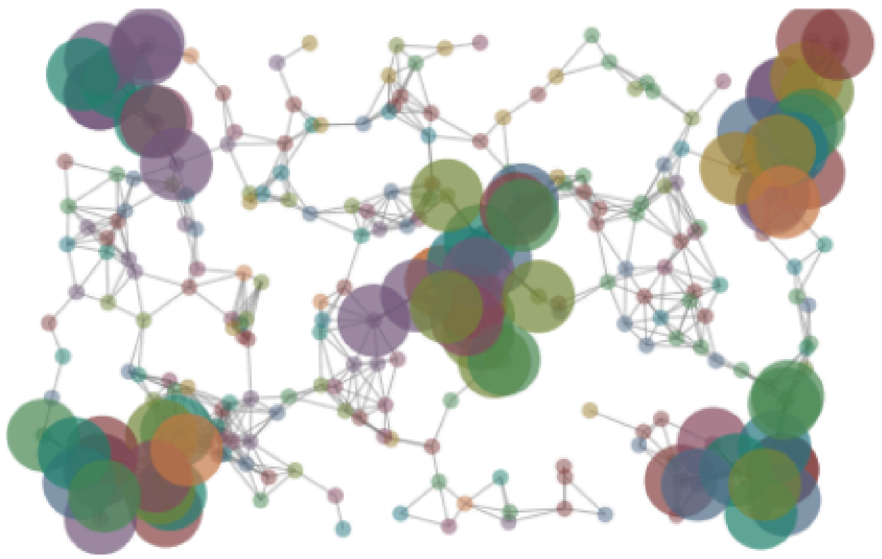
Graph Signal Diffusion



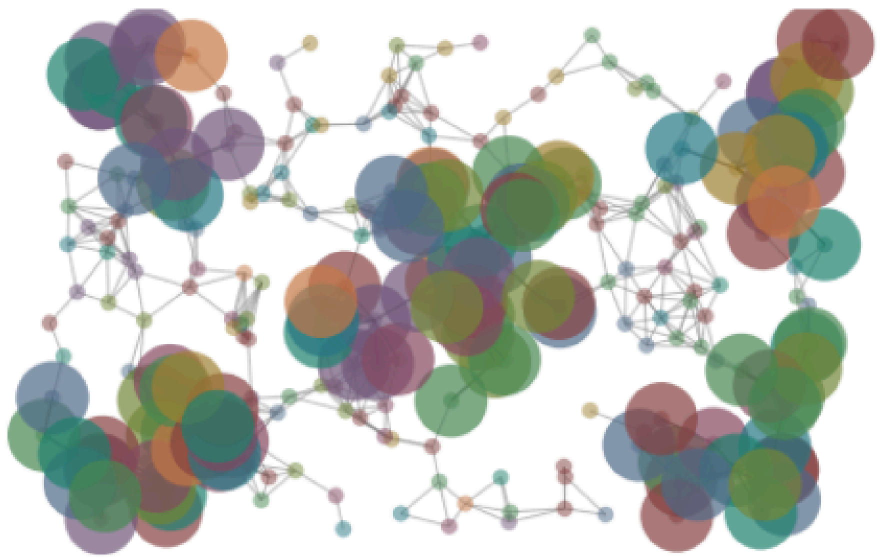
$$\mathbf{x}^{(0)} = \mathbf{x} \quad = \mathbf{S}^0\mathbf{x}$$



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



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



$$\mathbf{x}^{(3)} = \mathbf{S}\mathbf{x}^{(2)} \quad = \mathbf{S}^3\mathbf{x}$$


LEARNING WHERE?
imgflip.com

Graph Filters

Given graph shift operator S and coefficients $h_k,$ a graph filter is a polynomial (series) on S.

$$H(S) = \sum_{k=0}^{\infty} h_k S^k$$

The result of applying the filter $H(S)$ to the signal x is the signal

$$y = H(S)x = \sum_{k=0}^{\infty} h_k S^k x$$

We say that $y = h \star_S x$ is the graph convolution of the filter

$h = \{h_k\}_{k=0}^{\infty}$ with the signal x.

Graph Convolution

Graph convolutions aggregate information growing from local to global neighborhoods.

Consider a signal x supported on a graph with shift operator $S,$ along with filter $h = \{h_k\}_{k=0}^{K-1}.$

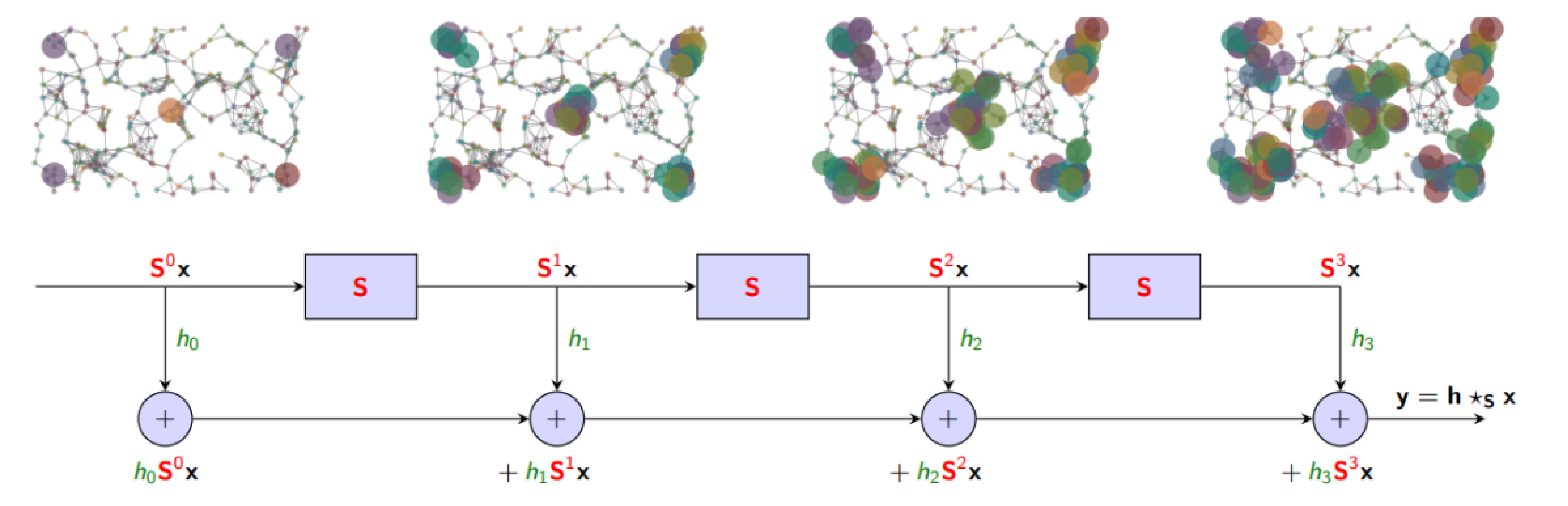
Graph convolution output is then given by:

$$y = h \star_S x = h_0 S^0 x + h_1 S^1 x + h_2 S^2 x + h_3 S^3 x + \cdots = \sum_{k=0}^{K-1} h_k S^k x$$

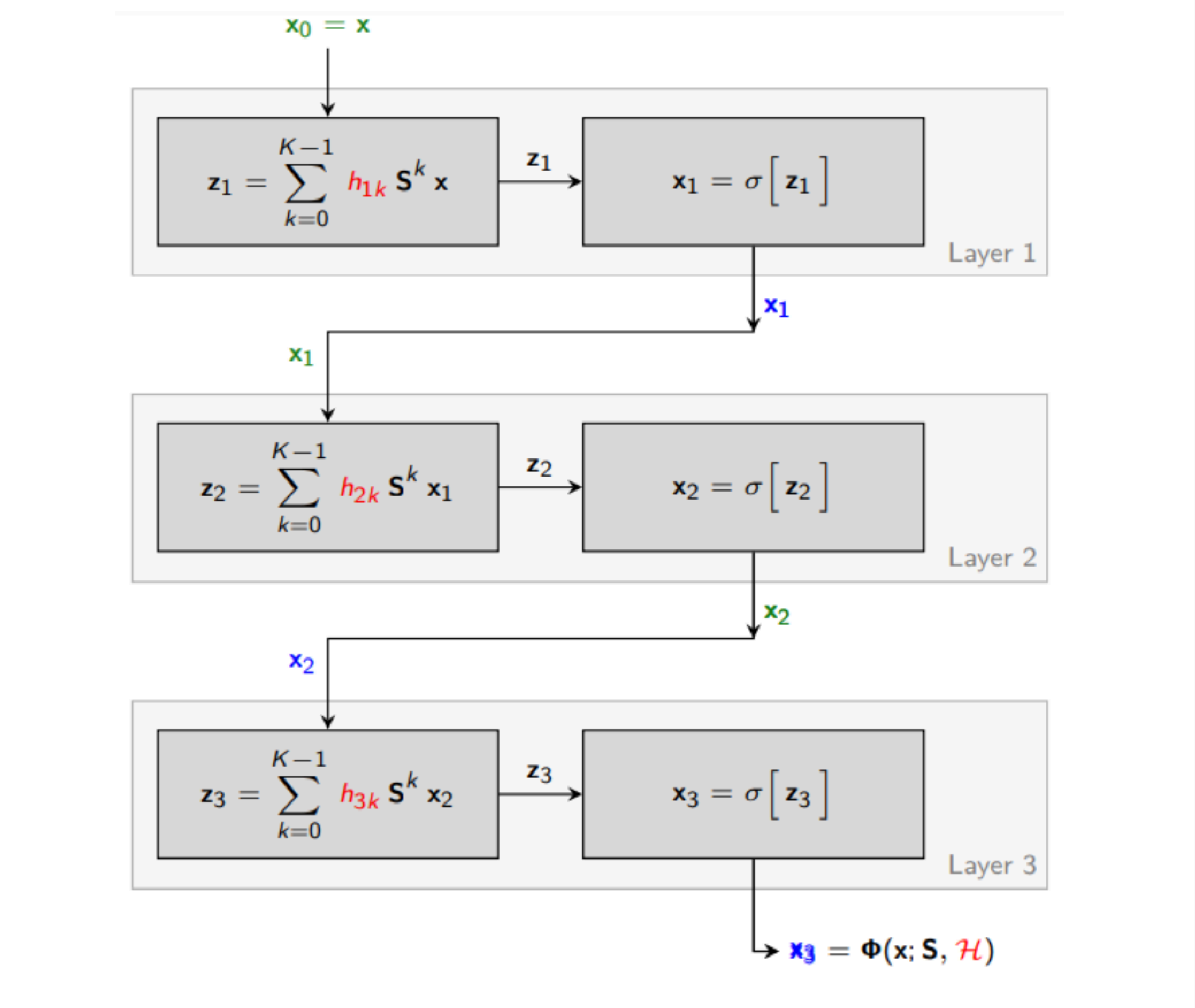
The same filter $h = \{h_k\}_{k=0}^{K-1}$ can be executed on multiple graphs $\Rightarrow$ the filter is transferable.

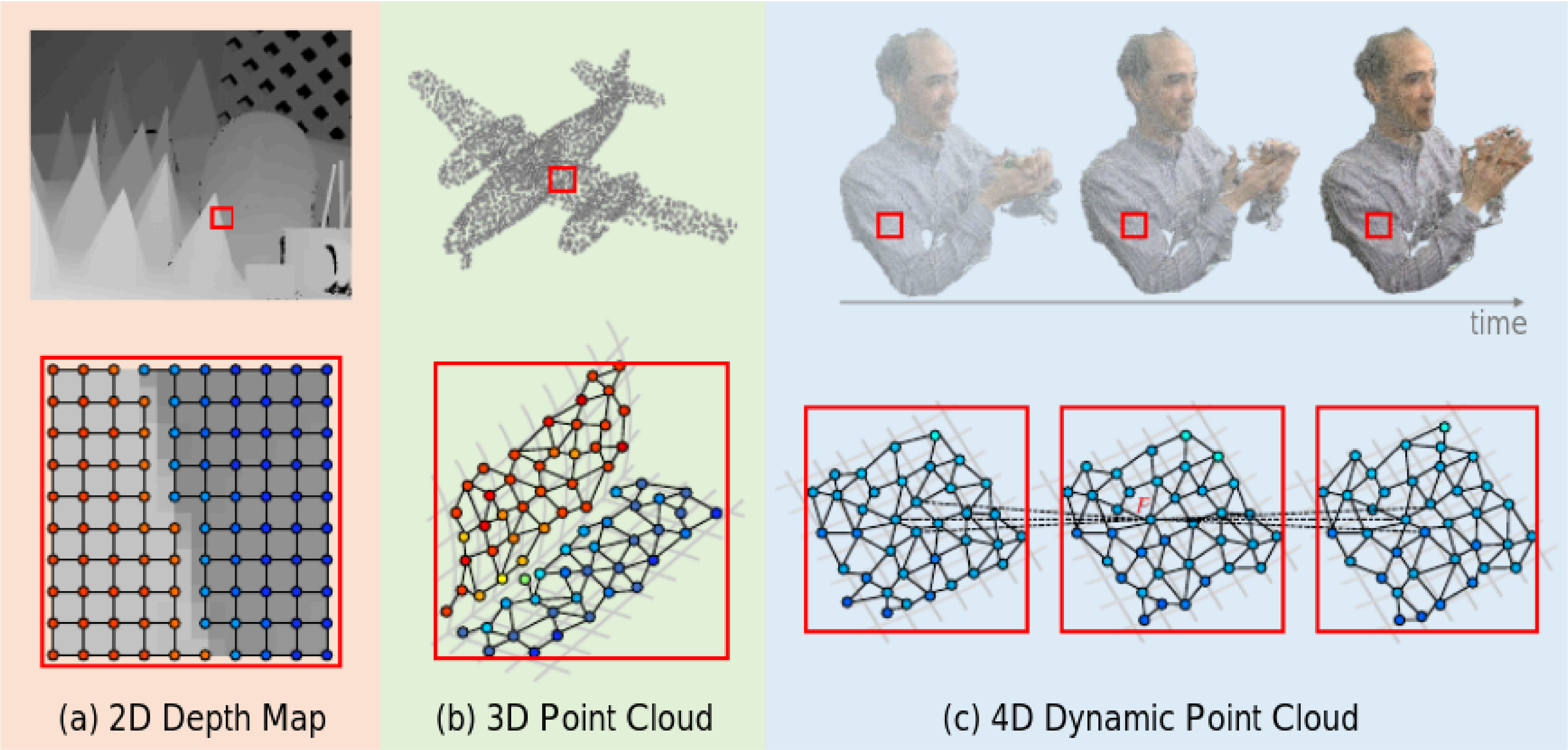
Output depends on the filter coefficients $h,$ the graph shift operator $S,$ and the signal $x.$

Graph Convolution



Graph Neural Network




time
(a) 2D Depth Map
(b) 3D Point Cloud
(c) 4D Dynamic Point Cloud

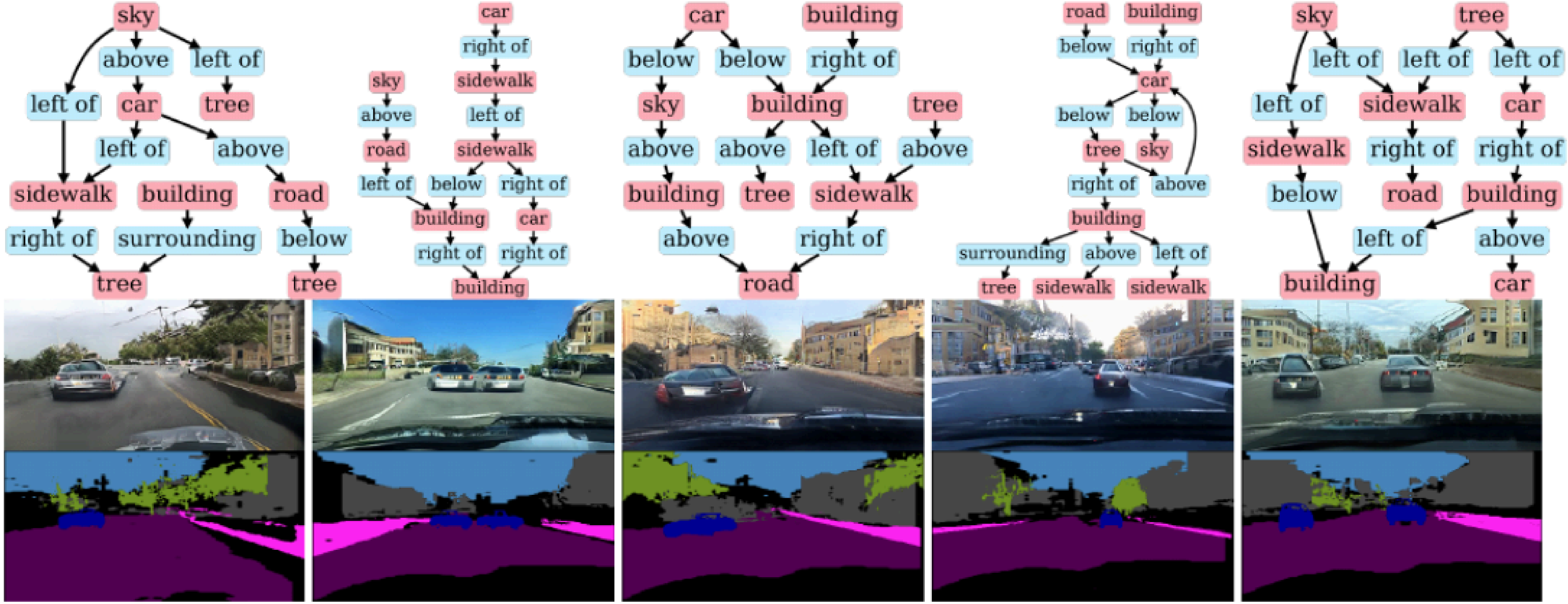


Fig. 4: Examples of synthetic scene graphs with corresponding BDD generated traffic scenes and semantic maps.

Different ideas

- GRNN: Combines graph structure with recurrent units for dynamic or sequential data.
- GRCN: Adds residual connections to enable deeper, stable GNN training.
- GAE & VGAE: Unsupervised embeddings for link prediction and graph generation.
- Graph Generative Models: Generate graphs (e.g., GraphVAE, GraphGAN, GCPN).
- Attention Variants: Multi-head and hierarchical attention for adaptive neighbor weighting.

Summary

- Images can be represented as grids in space. We generalize convolutional filters to graphs.
- Graphs can be represented in the form of matrices, providing graph shift operators and enabling operations over them.
- A graph signal is a vector in which each component xi is associated with node i.
- Graph signal, graph shift operator, and graph filter make up the core ingredients of a Graph Neural Network (GNN).
- Graph signal is the input. The graph shift operator is a parameter. We can also treat it as input if we want to consider different graphs.
- Graph filters are trainable parameters.
- A GCNN is composed of multiple graph perceptrons (graph filter + activation function).

Thank you