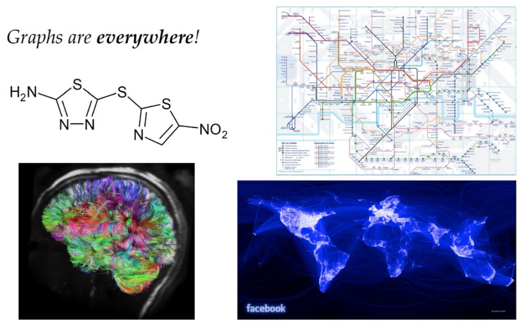
GDL Lecture 5: GRAPHS & SETS I

Levan Bokeria

GNN Reading group presentation, 2023-11-08

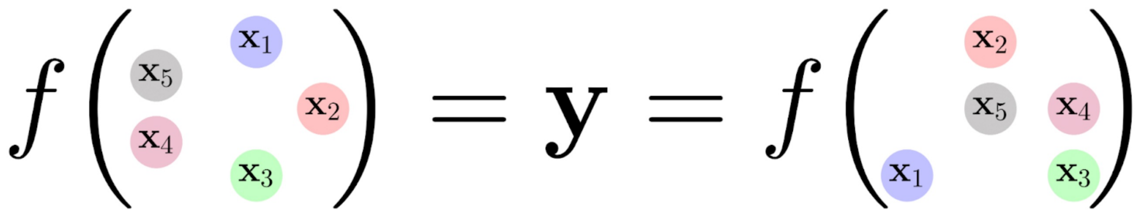
Slides adapted from the lecture slides of Petar Veličković

Part 1 overview:



no edges (i.e. $\Omega = \mathcal{V}$, the set of nodes)

$$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T \quad \begin{bmatrix} - & \mathbf{x}_1 & - \\ - & \mathbf{x}_2 & - \\ - & \mathbf{x}_3 & - \\ - & \mathbf{x}_4 & - \end{bmatrix}$$



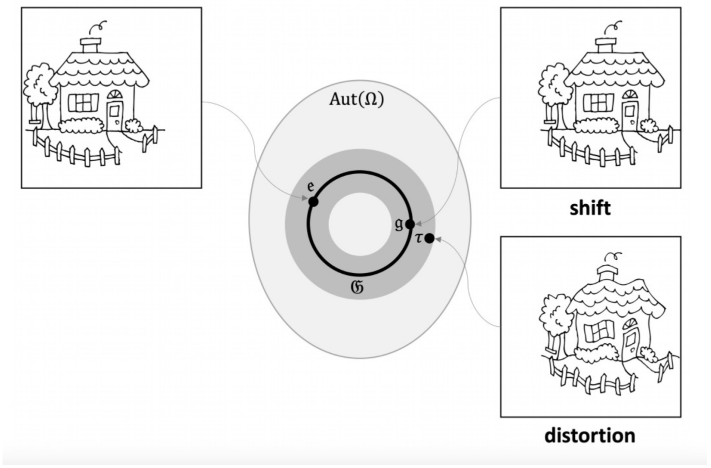
$$f(\mathbf{X}) = \phi\left(\sum_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$

$$\mathbf{P}_{(2,4,1,3)}\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} - & \mathbf{x}_1 & - \\ - & \mathbf{x}_2 & - \\ - & \mathbf{x}_3 & - \\ - & \mathbf{x}_4 & - \end{bmatrix} = \begin{bmatrix} - & \mathbf{x}_2 & - \\ - & \mathbf{x}_4 & - \\ - & \mathbf{x}_1 & - \\ - & \mathbf{x}_3 & - \end{bmatrix}$$

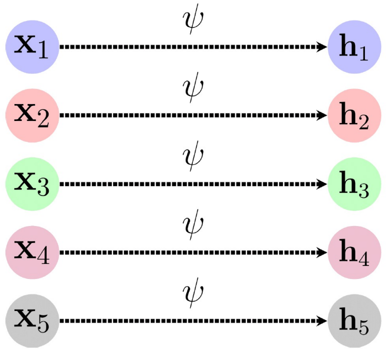
Permutation invariance:

$$f(\mathbf{PX}) = f(\mathbf{X})$$

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



$$\mathbf{h}_i = \psi(\mathbf{x}_i)$$



Permutation equivariance:

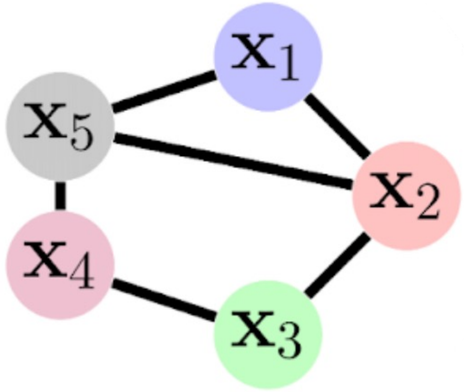
$$\mathbf{F}(\mathbf{PX}) = \mathbf{PF}(\mathbf{X})$$

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$

Part 2: graphs with edges

graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

$$a_{ij} = \begin{cases} 1, & (i,j) \in \mathcal{E} \\ 0, & (i,j) \notin \mathcal{E} \end{cases}$$

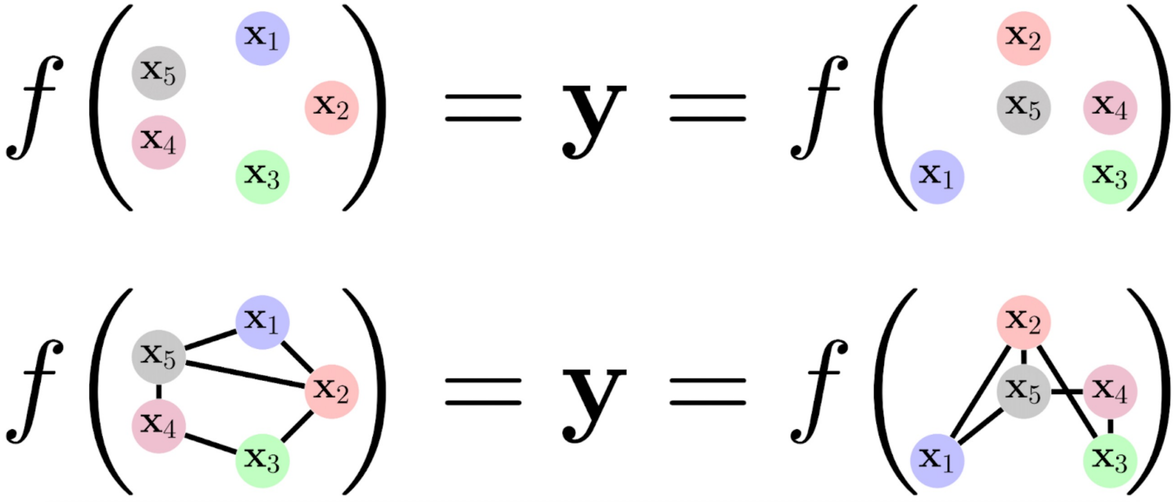


Adjacency matrix A

	x_1	x_2	x_3	x_4	x_5
x_1	-	1	0	0	1
x_2	1	-	1	0	1
x_3	0	1	-	1	0
x_4	0	0	1	-	1
x_5	1	1	0	1	-

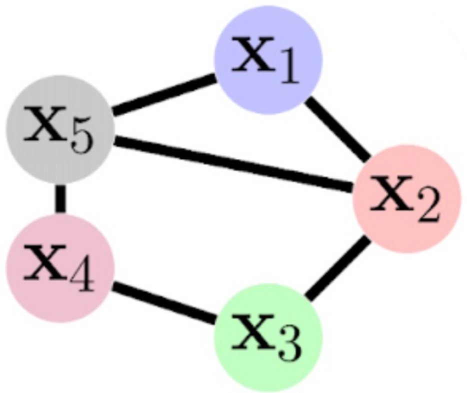
Part 2: graphs with edges

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

$$a_{ij} = \begin{cases} 1, & (i,j) \in \mathcal{E} \\ 0, & (i,j) \notin \mathcal{E} \end{cases}$$

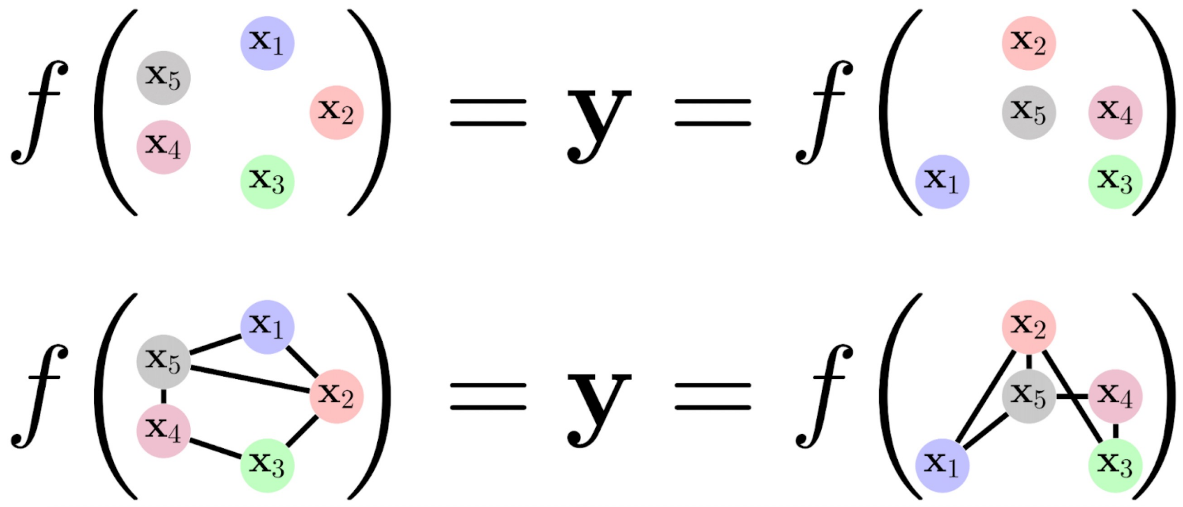


Adjacency matrix *A*

	x_1	x_2	x_3	x_4	x_5
x_1	-	1	0	0	1
x_2	1	-	1	0	1
x_3	0	1	-	1	0
x_4	0	0	1	-	1
x_5	1	1	0	1	-

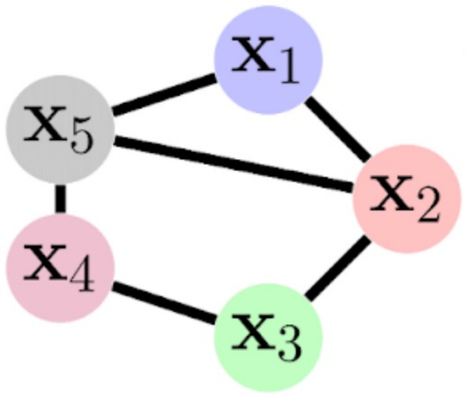
$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$

Part 2: graphs with edges



graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

$$a_{ij} = \begin{cases} 1, & (i,j) \in \mathcal{E} \\ 0, & (i,j) \notin \mathcal{E} \end{cases}$$



Adjacency matrix A

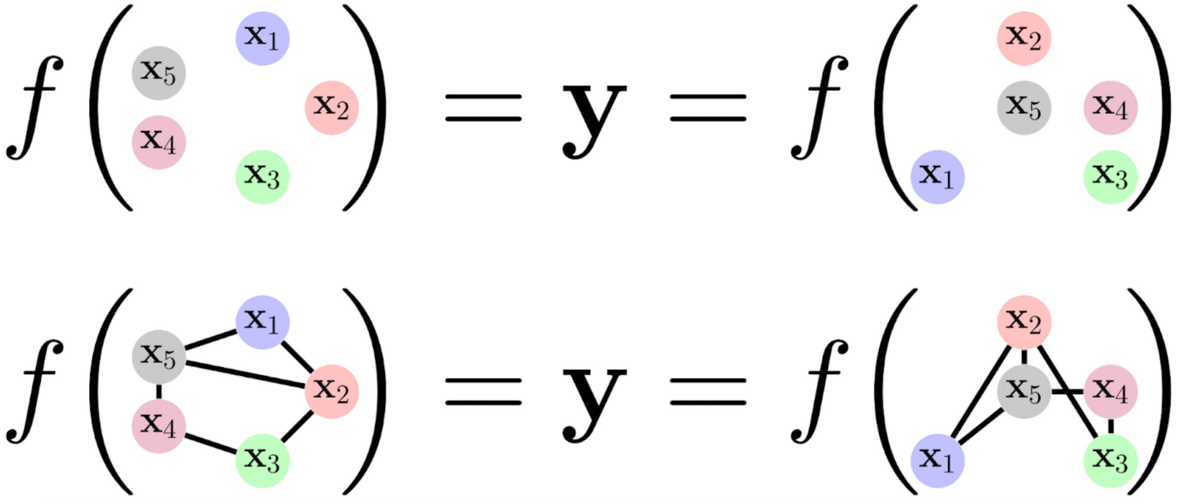
	x_1	x_2	x_3	x_4	x_5
x_1	-	1	0	0	1
x_2	1	-	1	0	1
x_3	0	1	-	1	0
x_4	0	0	1	-	1
x_5	1	1	0	1	-

Invariance: $f(\mathbf{PX}) = f(\mathbf{X})$

Equivariance: $\mathbf{F}(\mathbf{PX}) = \mathbf{PF}(\mathbf{X})$

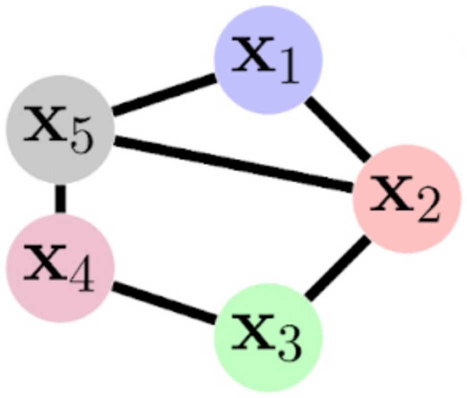
$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$

Part 2: graphs with edges



graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

$$a_{ij} = \begin{cases} 1, & (i,j) \in \mathcal{E} \\ 0, & (i,j) \notin \mathcal{E} \end{cases}$$



Adjacency matrix *A*

	x_1	x_2	x_3	x_4	x_5
x_1	-	1	0	0	1
x_2	1	-	1	0	1
x_3	0	1	-	1	0
x_4	0	0	1	-	1
x_5	1	1	0	1	-

Invariance: $f(\mathbf{PX}) = f(\mathbf{X})$

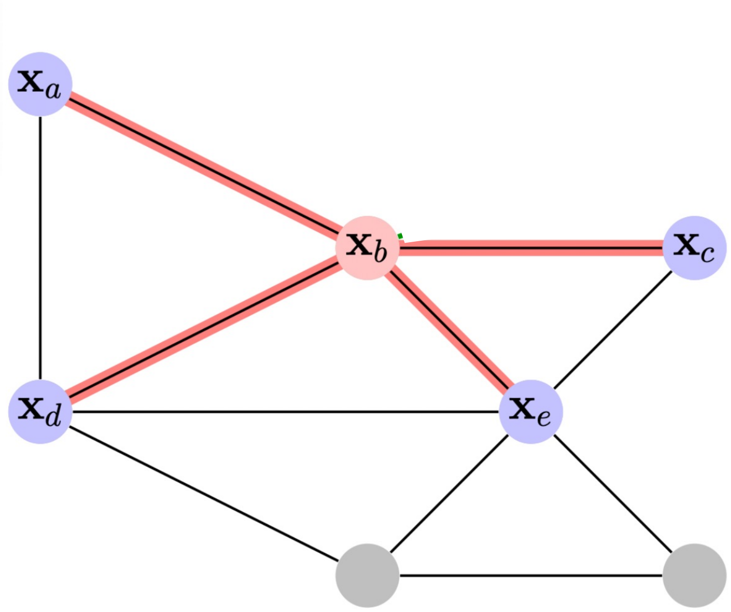
Equivariance: $\mathbf{F}(\mathbf{PX}) = \mathbf{PF}(\mathbf{X})$

Invariance: $f\left(\mathbf{PX}, \mathbf{PAP}^{\mathrm{T}}\right) = f(\mathbf{X}, \mathbf{A})$

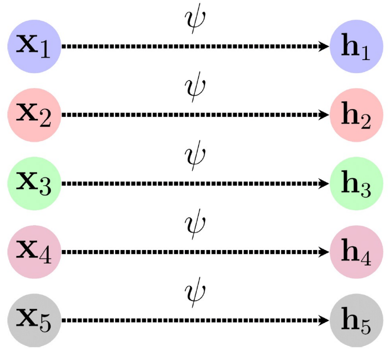
Equivariance: $\mathbf{F}\left(\mathbf{PX}, \mathbf{PAP}^{\mathrm{T}}\right) = \mathbf{PF}(\mathbf{X}, \mathbf{A})$

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$

Locality on graphs: Neighborhoods

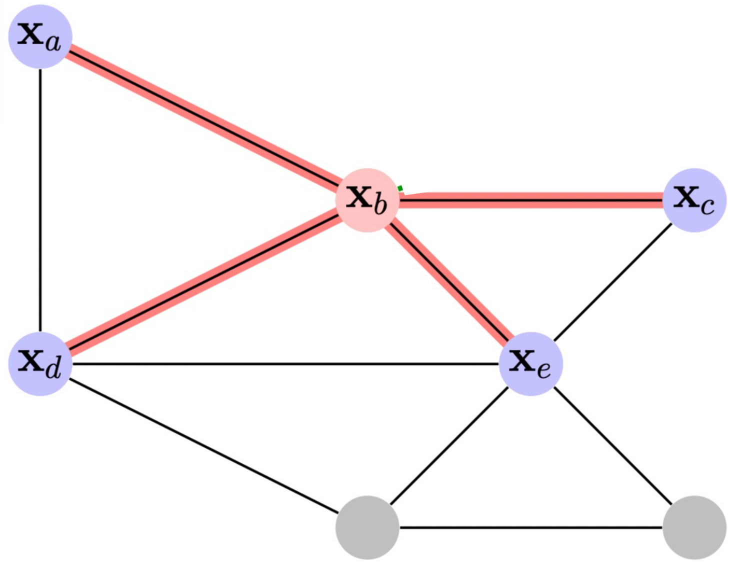


$$\mathbf{h}_i = \psi(\mathbf{x}_i)$$



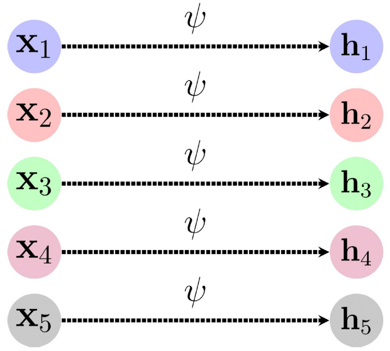
Locality on graphs: Neighborhoods

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



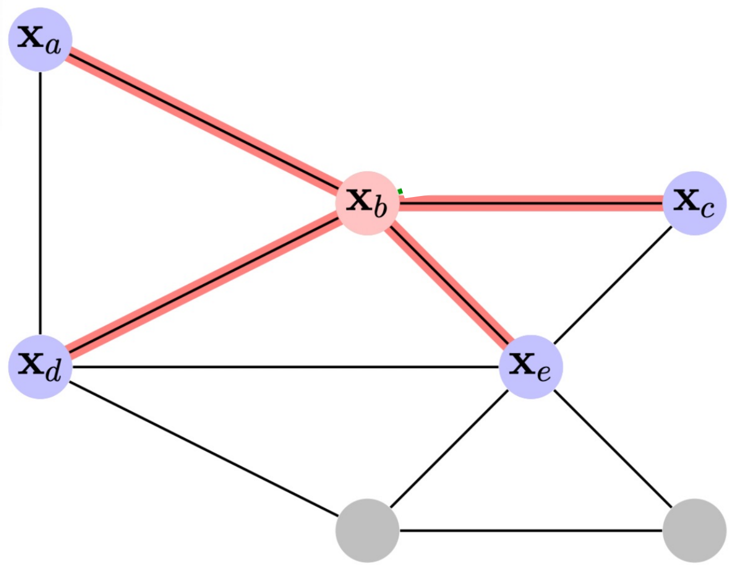
$$\mathbf{X}_{\mathcal{N}_b} = \{\{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d, \mathbf{x}_e\}\}$$

$$\mathbf{h}_i = \psi(\mathbf{x}_i)$$



Locality on graphs: Neighborhoods

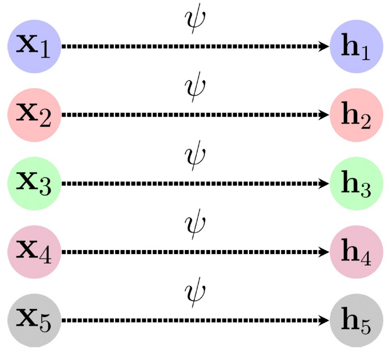
$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



$$\mathbf{X}_{\mathcal{N}_b} = \{\!\{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d, \mathbf{x}_e\}\!\}$$

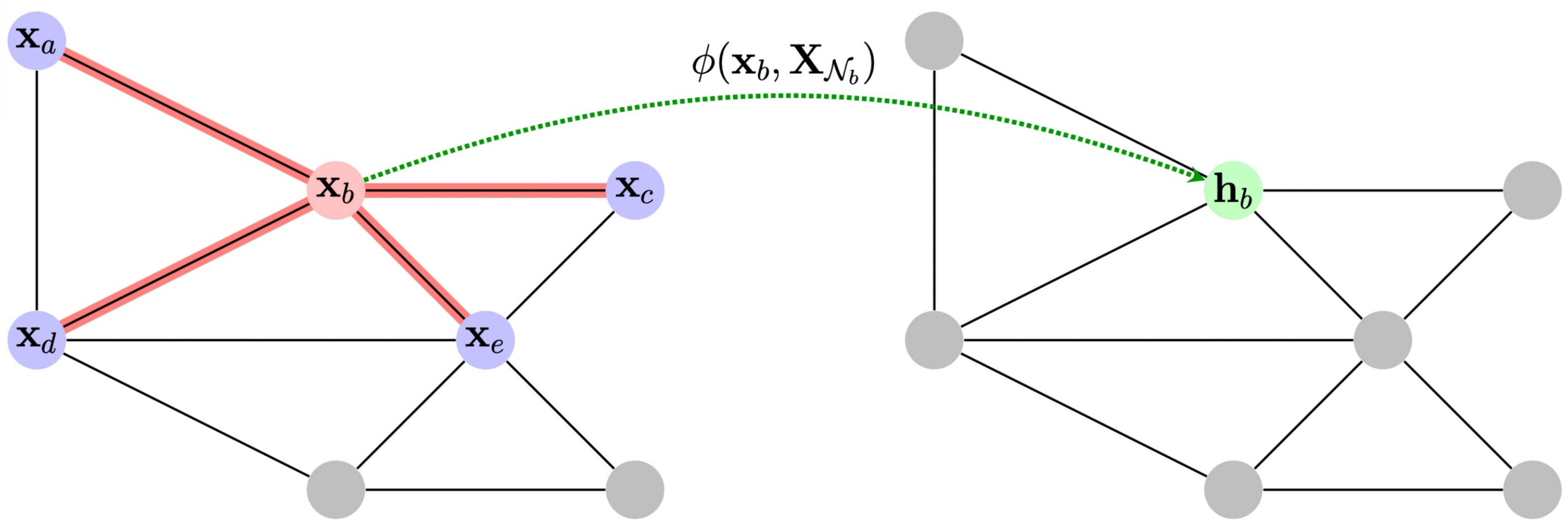
$$\mathbf{X}_{\mathcal{N}_i} = \{\!\{\, \mathbf{x}_j : j \in \mathcal{N}_i \,\}\!\}$$

$$\mathbf{h}_i = \psi(\mathbf{x}_i)$$



$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$

Locality on graphs: Neighborhoods

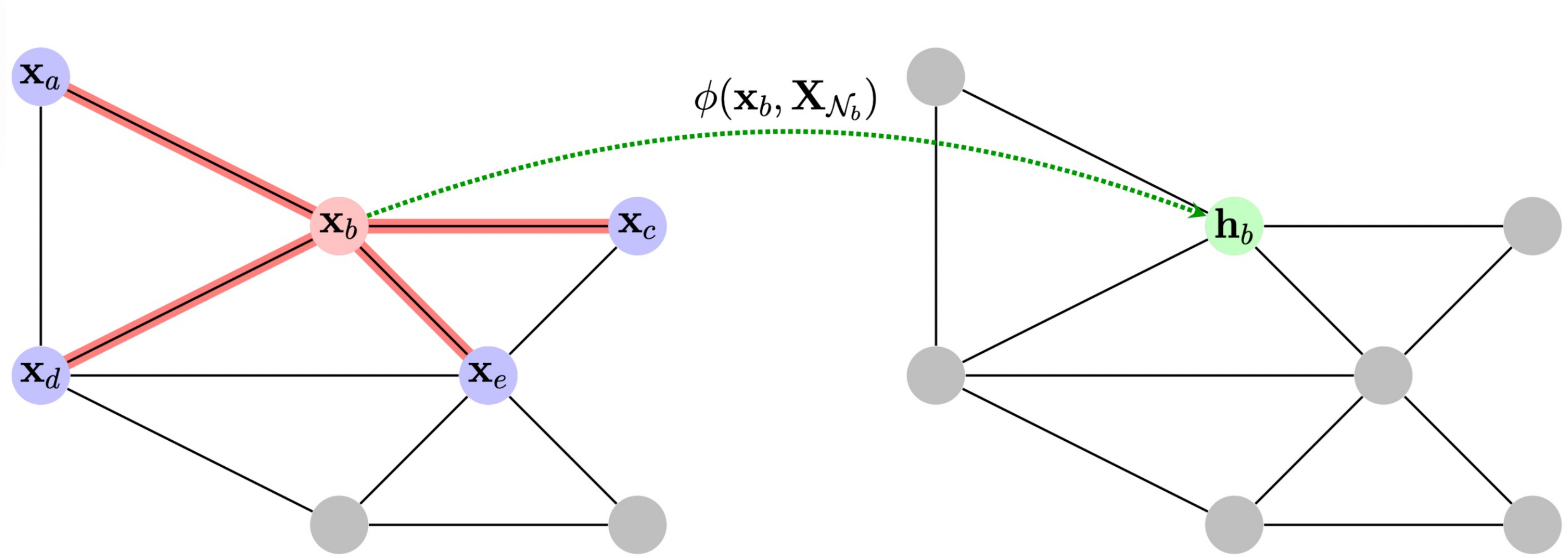


$$\mathbf{X}_{\mathcal{N}_b} = \{\!\{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d, \mathbf{x}_e\}\!\}$$

$$\mathbf{X}_{\mathcal{N}_i} = \{\{\, \mathbf{x}_j : j \in \mathcal{N}_i \,\}\}$$

Locality on graphs: Neighborhoods

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



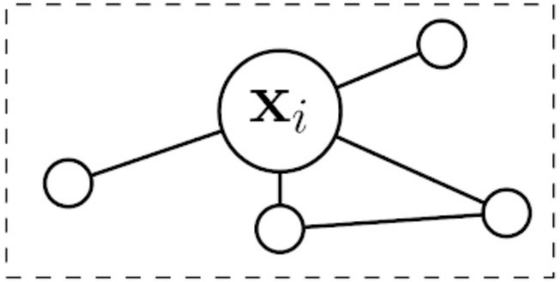
$$\mathbf{X}_{\mathcal{N}_b} = \{\{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d, \mathbf{x}_e\}\}$$

$$\mathbf{X}_{\mathcal{N}_i} = \{\{\, \mathbf{x}_j : j \in \mathcal{N}_i \,\}\}$$

$$\mathbf{F}(\mathbf{X}, \mathbf{A}) = \begin{bmatrix} - & \phi(\mathbf{x}_1, \mathbf{X}_{\mathcal{N}_1}) & - \\ - & \phi(\mathbf{x}_2, \mathbf{X}_{\mathcal{N}_2}) & - \\ & \vdots & \\ - & \phi(\mathbf{x}_n, \mathbf{X}_{\mathcal{N}_n}) & - \end{bmatrix}$$

General blueprint for learning on graphs

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$

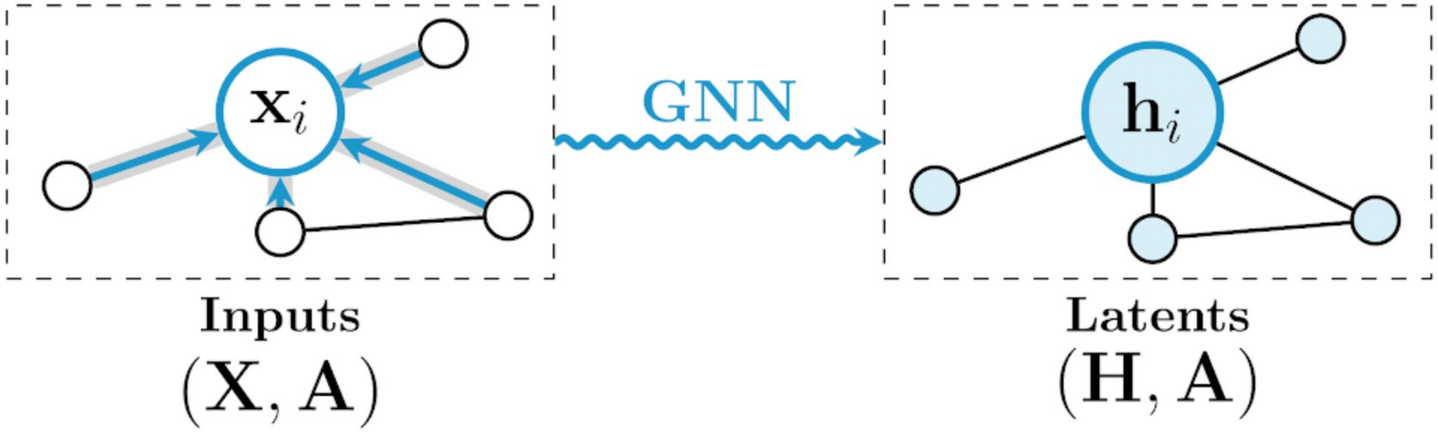


Inputs

$(\mathbf{X}, \mathbf{A})$

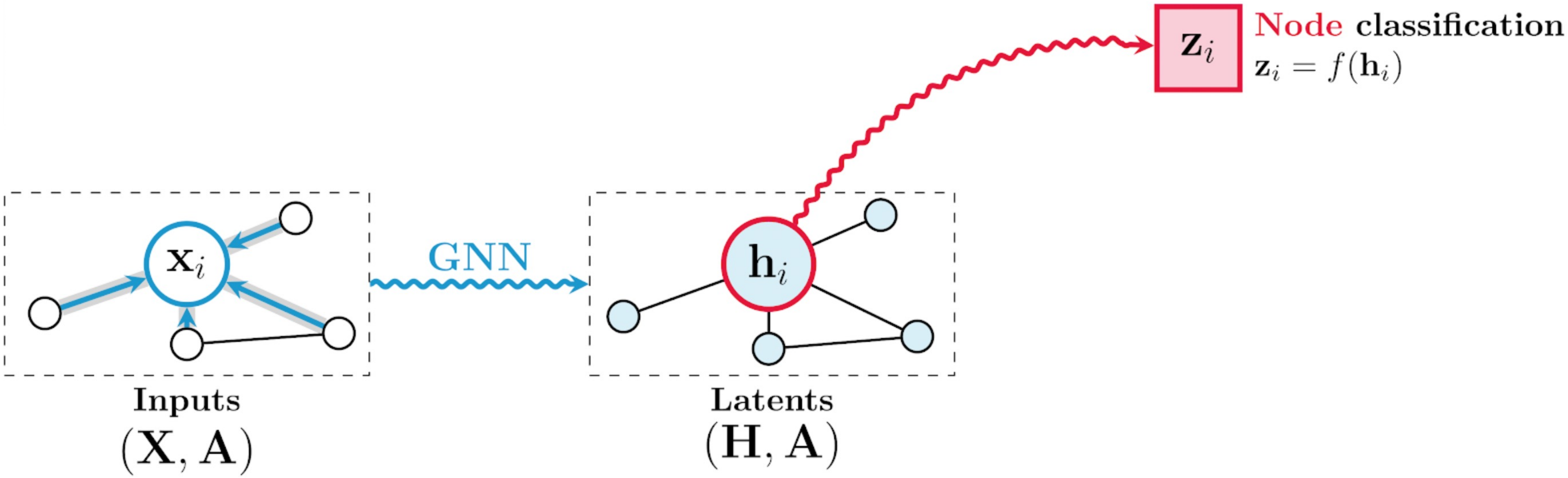
General blueprint for learning on graphs

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



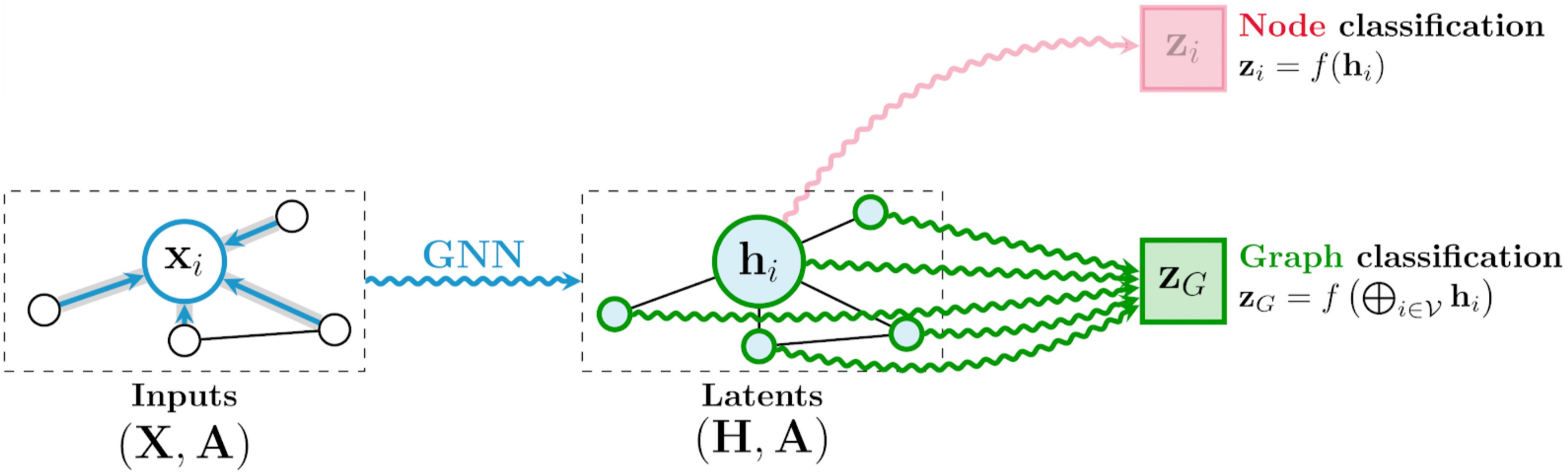
General blueprint for learning on graphs

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



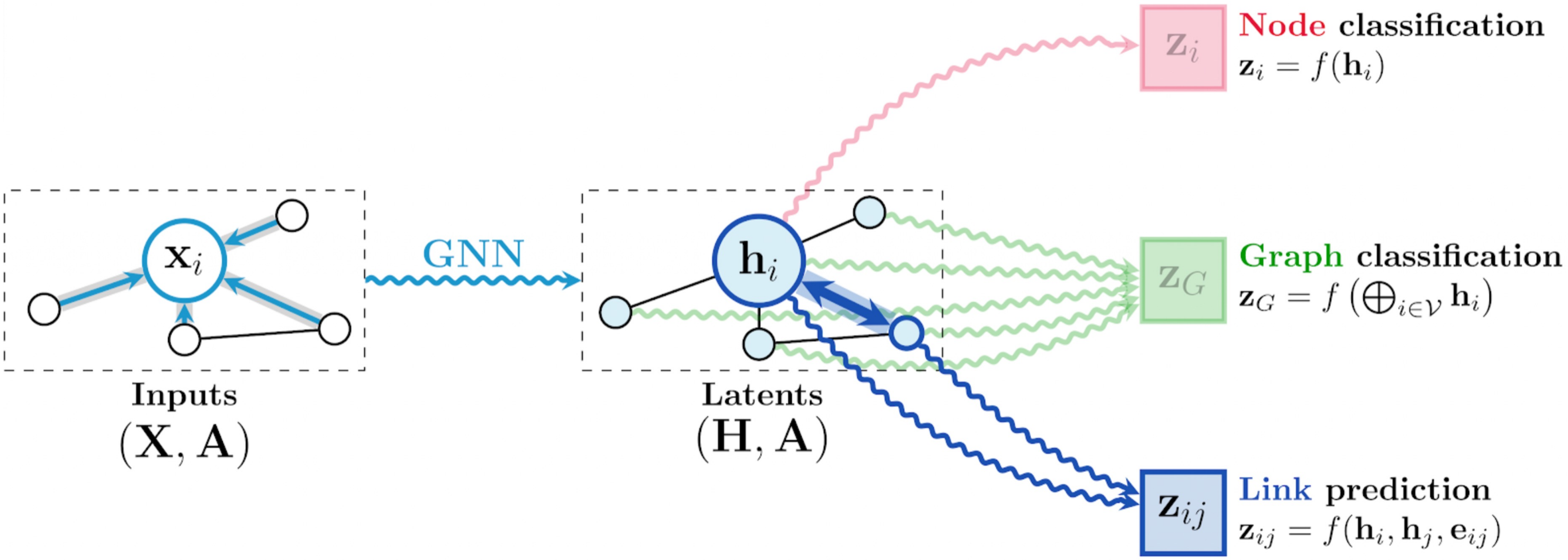
General blueprint for learning on graphs

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



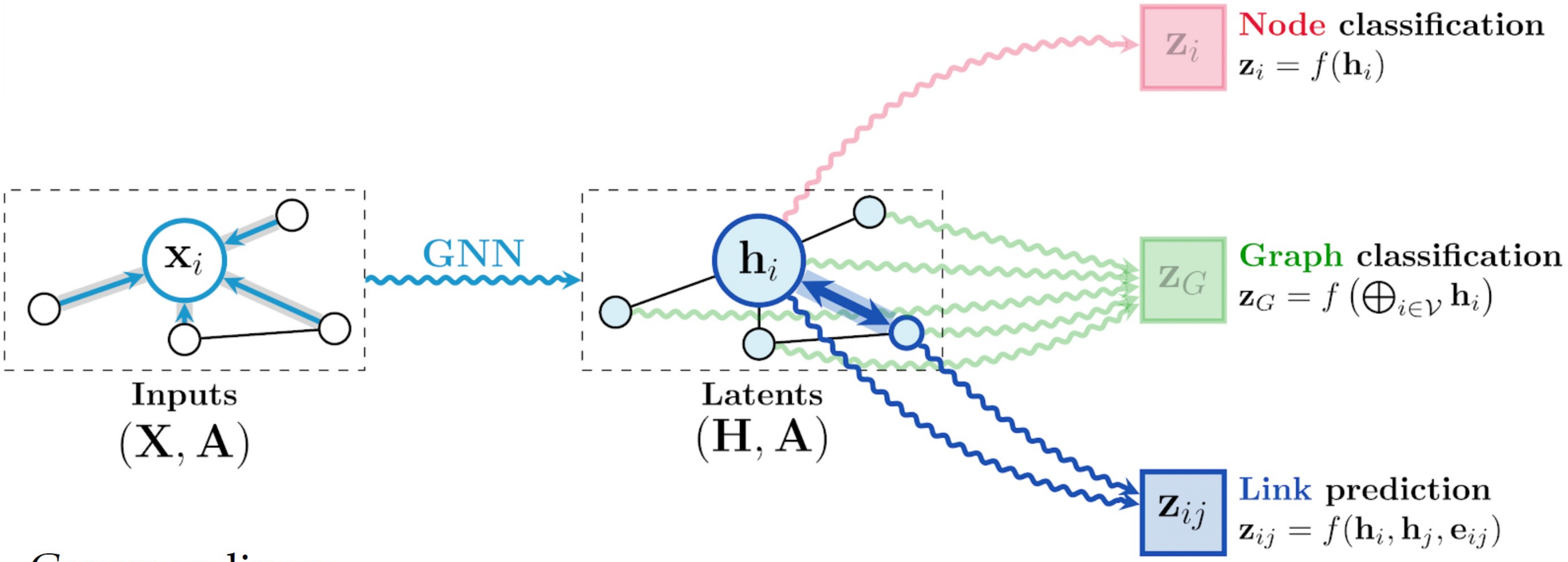
General blueprint for learning on graphs

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



General blueprint for learning on graphs

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



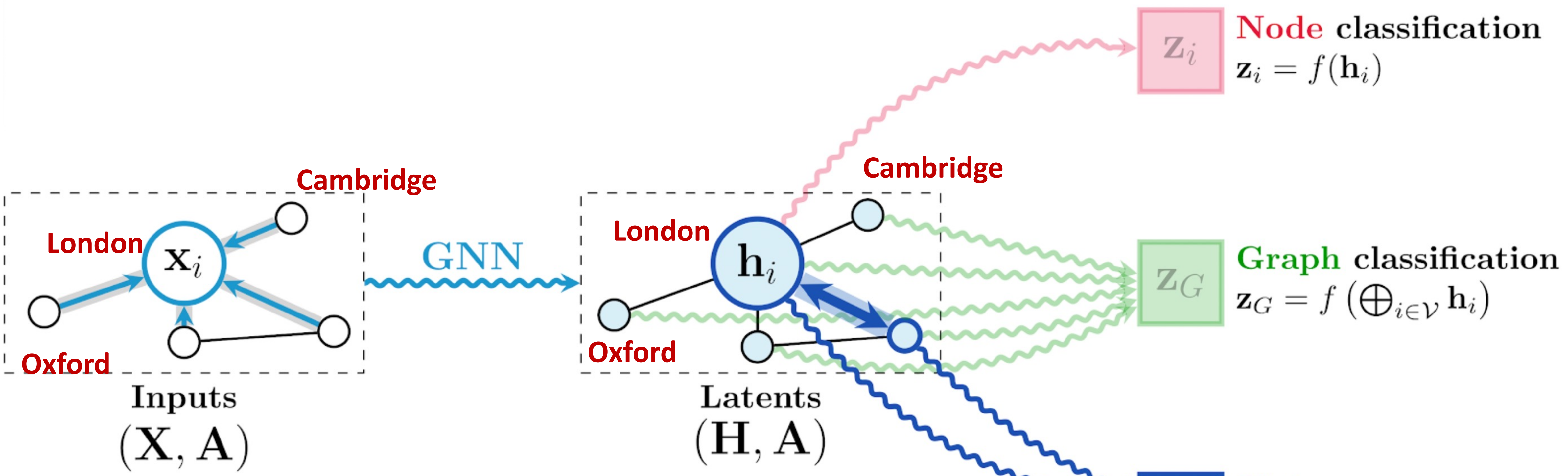
Common lingo:

F is a *"GNN layer"*

ϕ is *"diffusion"* / *"propagation"* / *"message passing"*

General blueprint for learning on graphs

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



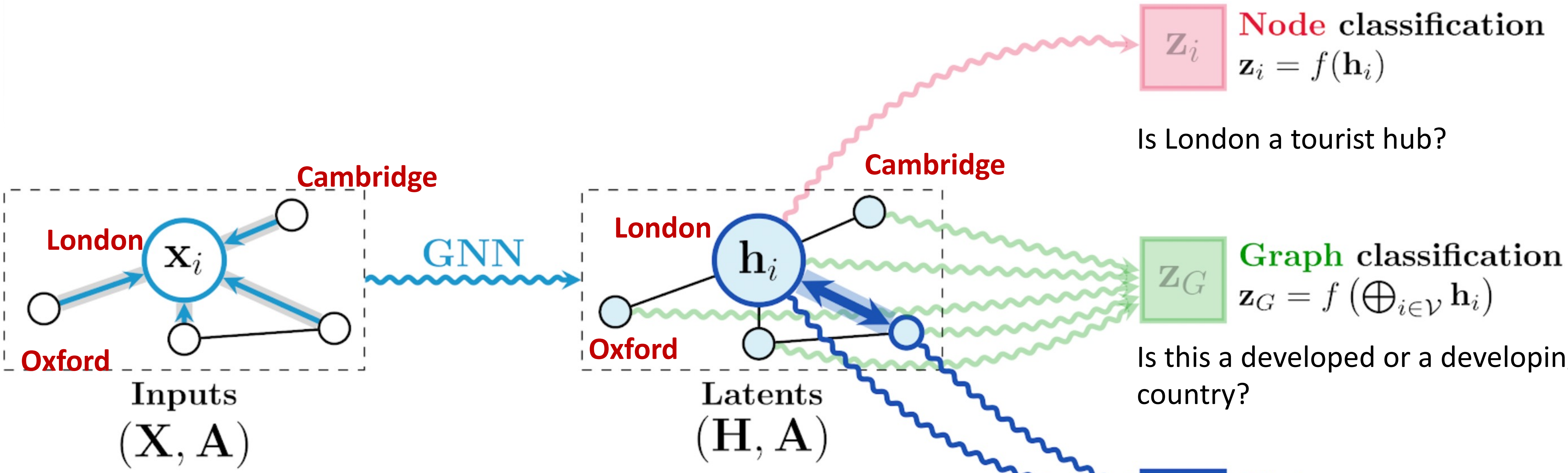
Common lingo:

F is a "*GNN layer*"

ϕ is "*diffusion*" / "*propagation*" / "*message passing*"

General blueprint for learning on graphs

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



Is London a tourist hub?

Is this a developed or a developing country?

Is there a direct train?

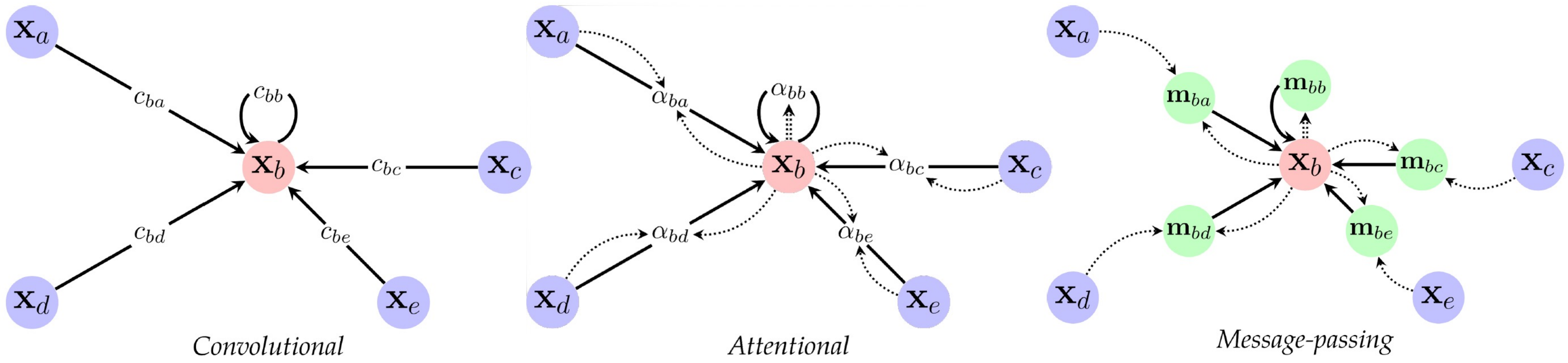
Common lingo:

F is a *"GNN layer"*

ϕ is *"diffusion"* / *"propagation"* / *"message passing"*

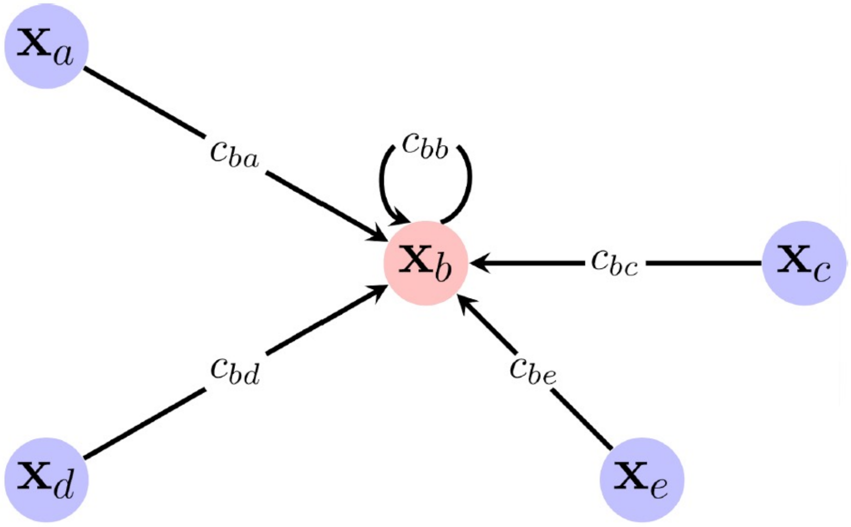
The three "flavours" of GNN layers

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



The three "flavours" of GNN layers

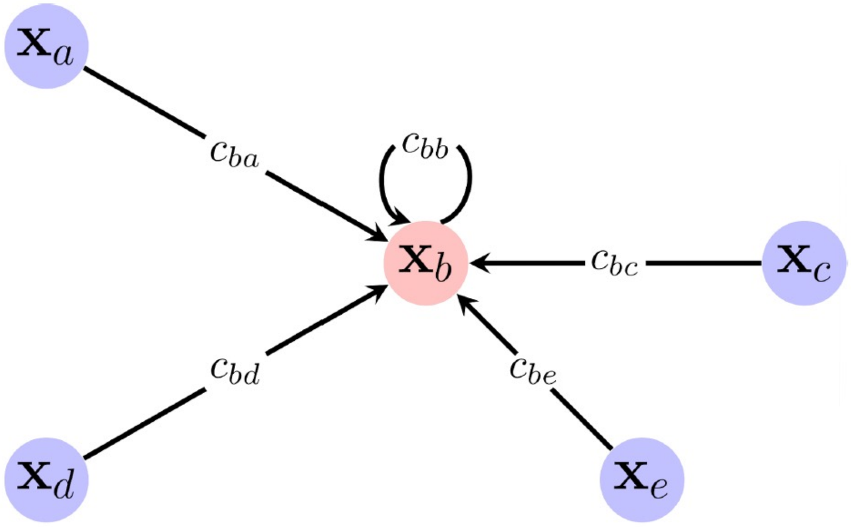
$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



Convolutional

The three "flavours" of GNN layers

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$

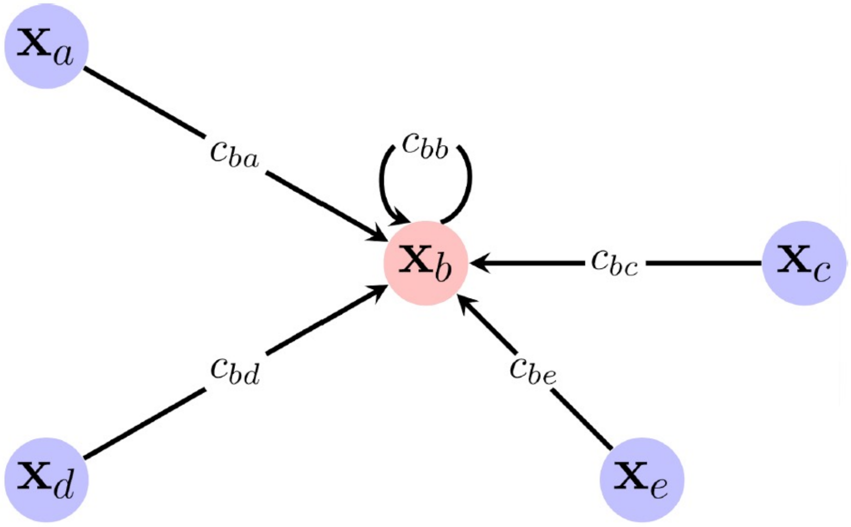


Convolutional

- $c_{i,j}$ as weights
- Edges = similarity (himophilous graphs)
- Scalable

The three "flavours" of GNN layers

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$



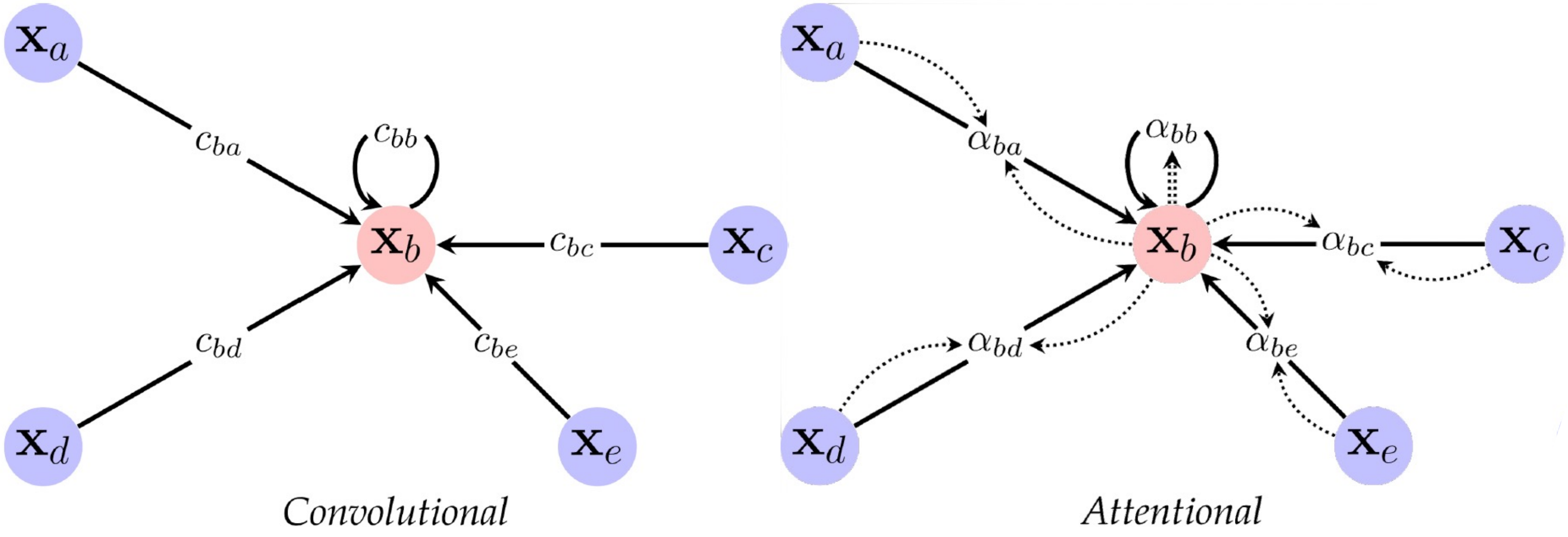
Convolutional

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right)$$

- $c_{i,j}$ as weights
- Edges = similarity (himophilous graphs)
- Scalable

The three "flavours" of GNN layers

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$

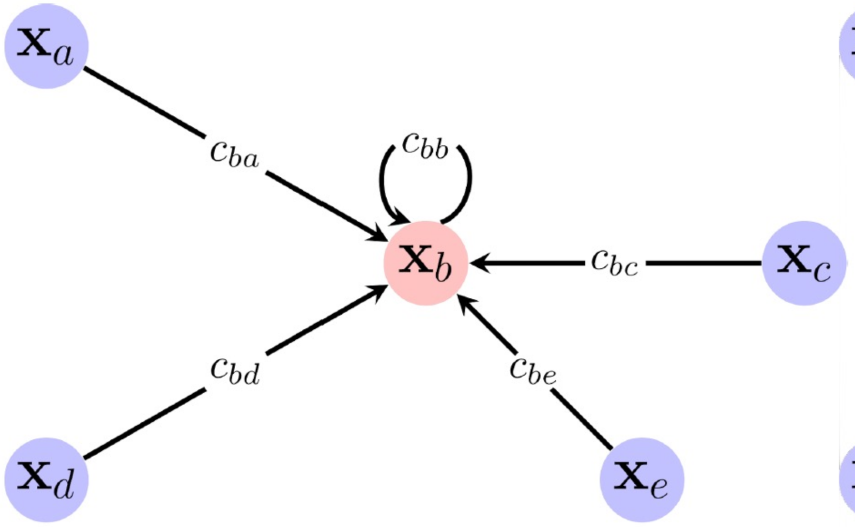


$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right)$$

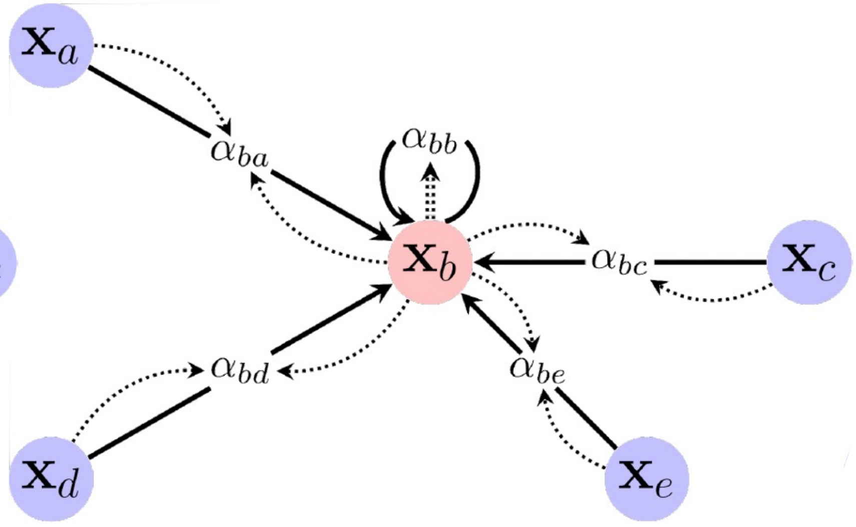
- $c_{i,j}$ as weights
- Edges = similarity (himophilous graphs)
- Scalable

The three "flavours" of GNN layers

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



Convolutional



Attentional

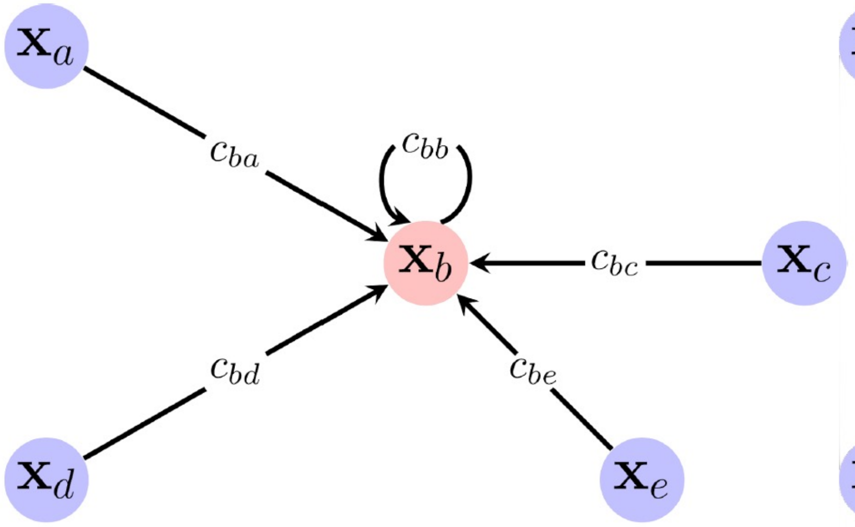
$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right)$$

- $c_{i,j}$ as weights
- Edges = similarity (himophilous graphs)
- Scalable

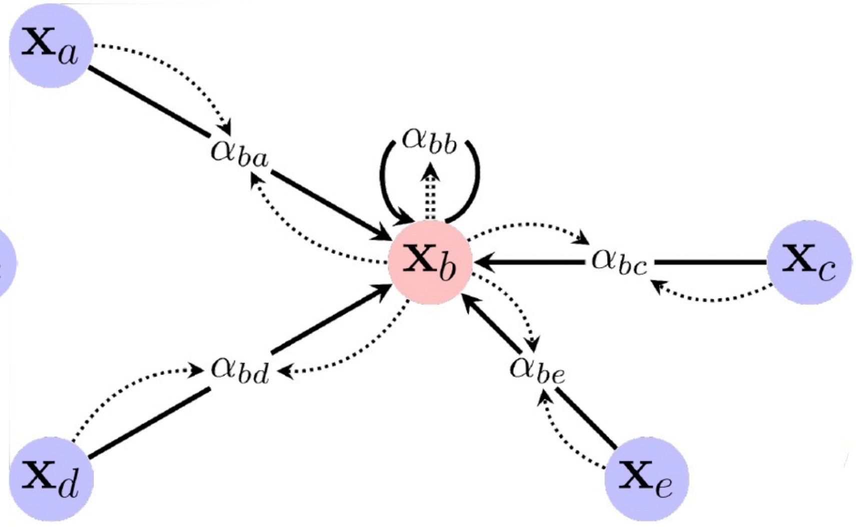
- $\alpha_{i,j} = a(x_i, x_j)$
- E.g. some edges = dissimilarity
- Still quite scalable

The three "flavours" of GNN layers

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



Convolutional



Attentional

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right)$$

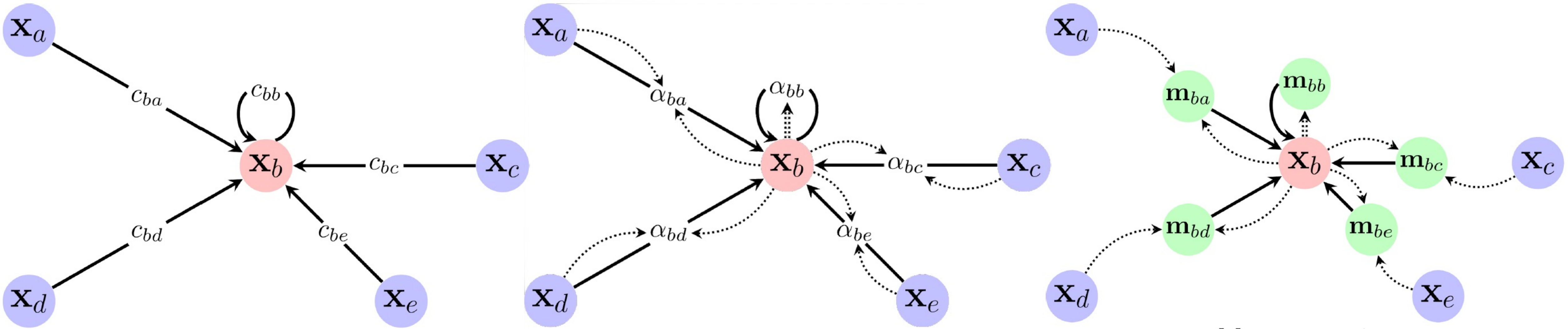
$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} a(\mathbf{x}_i, \mathbf{x}_j)\psi(\mathbf{x}_j)\right)$$

- $c_{i,j}$ as weights
- Edges = similarity (himophilous graphs)
- Scalable

- $\alpha_{i,j} = a(x_i, x_j)$
- E.g. some edges = dissimilarity
- Still quite scalable

The three "flavours" of GNN layers

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



Convolutional *Attentional* *Message-passing*

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right)$$

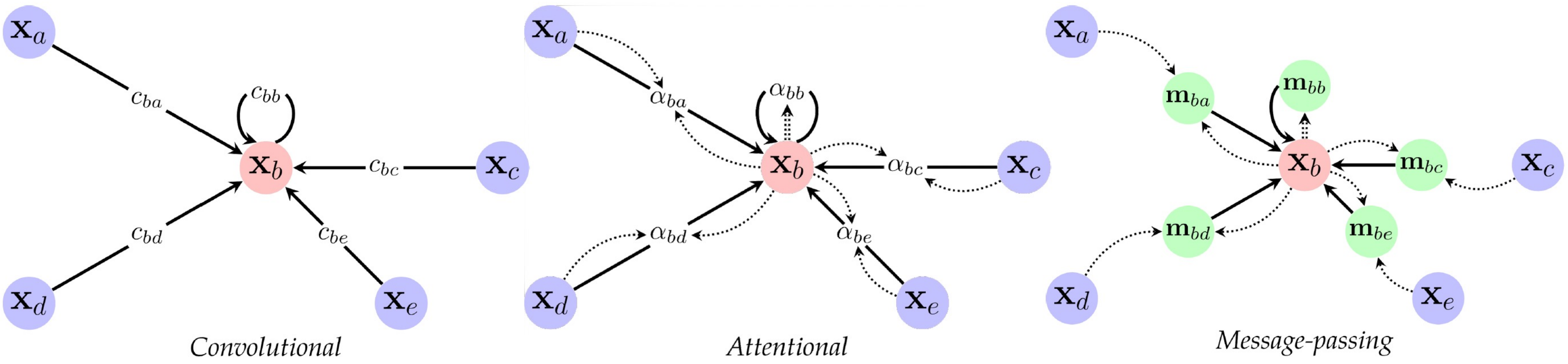
$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} a(\mathbf{x}_i,\mathbf{x}_j)\psi(\mathbf{x}_j)\right)$$

- $c_{i,j}$ as weights
- Edges = similarity (himophilous graphs)
- Scalable

- $\alpha_{i,j} = a(x_i, x_j)$
- E.g. some edges = dissimilarity
- Still quite scalable

The three "flavours" of GNN layers

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right)$$

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} a(\mathbf{x}_i,\mathbf{x}_j)\psi(\mathbf{x}_j)\right)$$

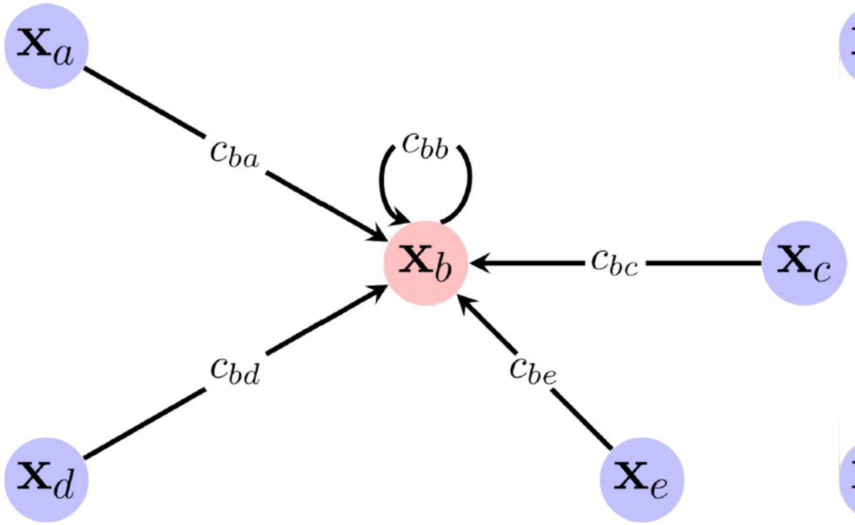
- $c_{i,j}$ as weights
- Edges = similarity (himophilous graphs)
- Scalable

- $\alpha_{i,j} = a(x_i, x_j)$
- E.g. some edges = dissimilarity
- Still quite scalable

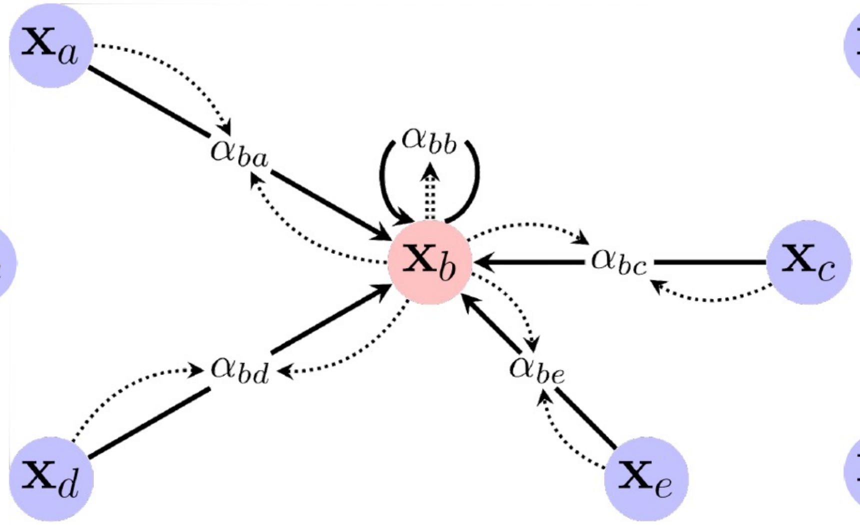
- $m_{i,j} = \psi(x_i, x_j)$
- Compute arbitrary vectors m

The three "flavours" of GNN layers

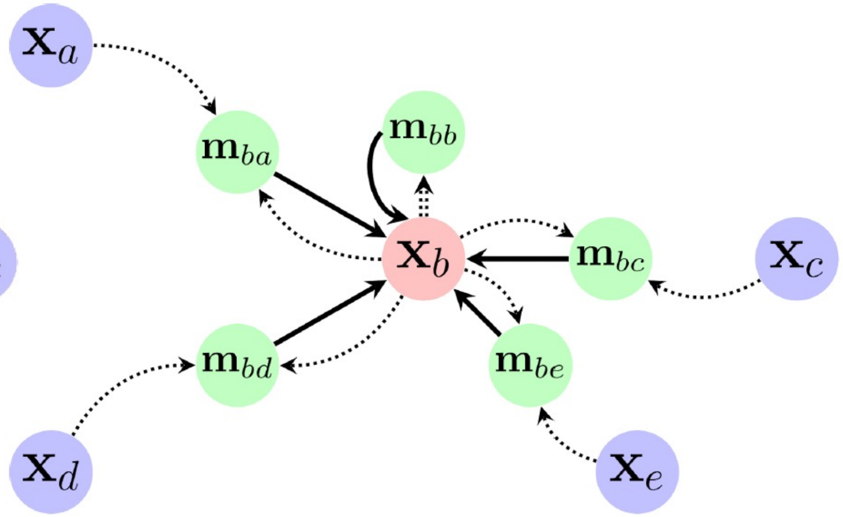
$$f(\mathbf{X}) = \phi\left(\bigoplus_{i\in\mathcal{V}} \psi(\mathbf{x}_i)\right)$$



Convolutional



Attentional



Message-passing

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right)$$

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} a(\mathbf{x}_i,\mathbf{x}_j)\psi(\mathbf{x}_j)\right)$$

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} \psi(\mathbf{x}_i,\mathbf{x}_j)\right)$$

- $c_{i,j}$ as weights
- Edges = similarity (himophilous graphs)
- Scalable

- $\alpha_{i,j} = a(x_i, x_j)$
- E.g. some edges = dissimilarity
- Still quite scalable

- $m_{i,j} = \psi(x_i, x_j)$
- Compute arbitrary vectors m