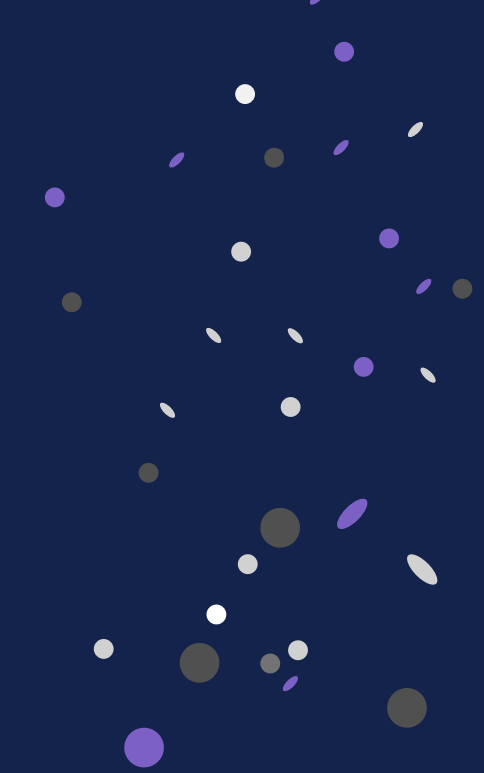
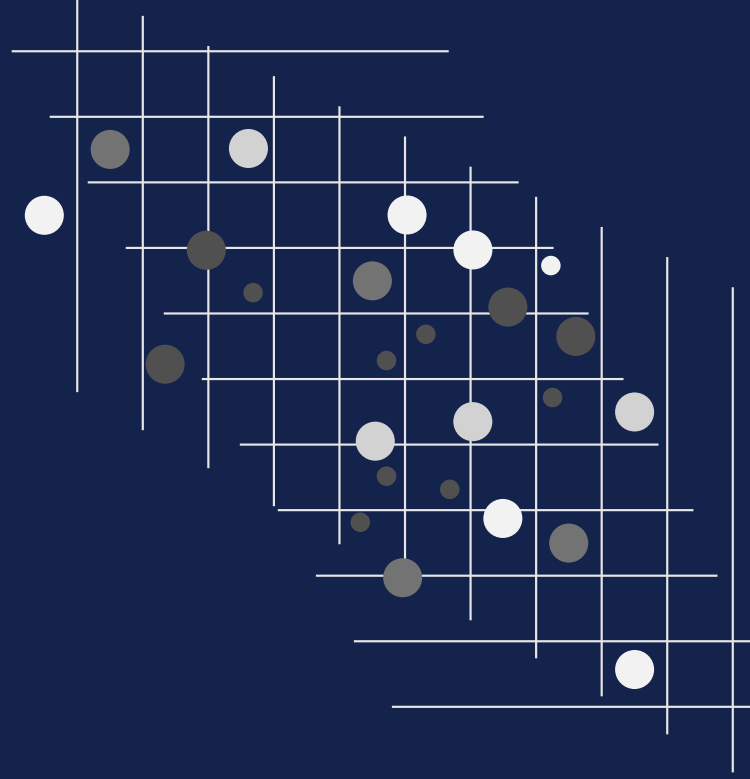
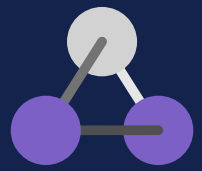




Introduction to PyTorch Geometric

Adapted from: https://petar-v.com/talks/GNN-Wednesday.pdf



Notebook:
https://tinyurl.com/PyG3DVSS

PyG3DVSS

Slides:
https://tinyurl.com/GNN3DVSS

GNN3DVSS

Representation of Sets

$$\mathbf{S} = \{X_1, X_2, \ldots, X_n\}$$

$$x_i \in \mathbb{R}^m \text{ be the feature vector of } X_i$$

$$\mathbf{X} = (x_1, x_2, \ldots, x_n)^{\mathbf{T}}$$

Symmetries in sets

$$\mathbf{X} = (x_1, x_2, \ldots, x_n)^{\mathbf{T}}$$

We implicitly imposed an ordering on the set items.

Is there a notion of ordering in sets?

Functions on sets should not depend on the ordering!

Permutation Invariance

Permutation invariant functions are required to model set-level outputs (like energy of molecules)

$$f(\mathbf{PX}) = f(\mathbf{X}) \text{ where } \mathbf{P} \text{ is a permutation matrix}$$

$$f(\mathbf{X}) = \Phi\left(\sum_{i=1}^{n} \psi(x_i)\right)$$

Permutation Equivariance

Permutation equivariant functions are required to model item-level outputs

$$f(\mathbf{PX}) = \mathbf{P}f(\mathbf{X}) \text{ where } \mathbf{P} \text{ is a permutation matrix}$$

$$h_i = \psi(x_i)$$

$$\mathbf{H} = (h_1, h_2, \ldots, h_n) = f(\mathbf{X})$$

Moving to graphs!

$$\mathbf{V} = \{V_1, V_2, \ldots, V_n\} \qquad \mathbf{X} = (x_1, x_2, \ldots, x_n)^{\mathbf{T}}$$

$$\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$$

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix

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

Symmetries in graphs

Invariance:

$$f(\mathbf{PX}, \mathbf{PAP^T}) = f(\mathbf{X}, \mathbf{A})$$

Equivariance:

$$f(\mathbf{PX}, \mathbf{PAP^T}) = \mathbf{P}f(\mathbf{X}, \mathbf{A})$$

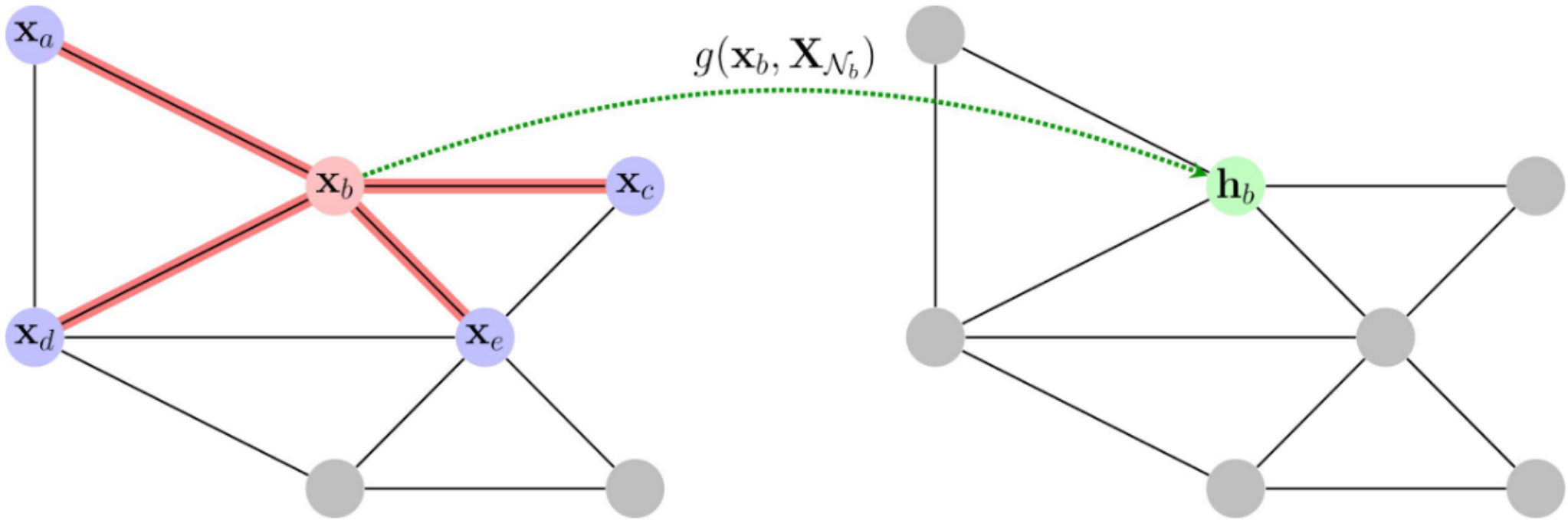
Neighborhoods

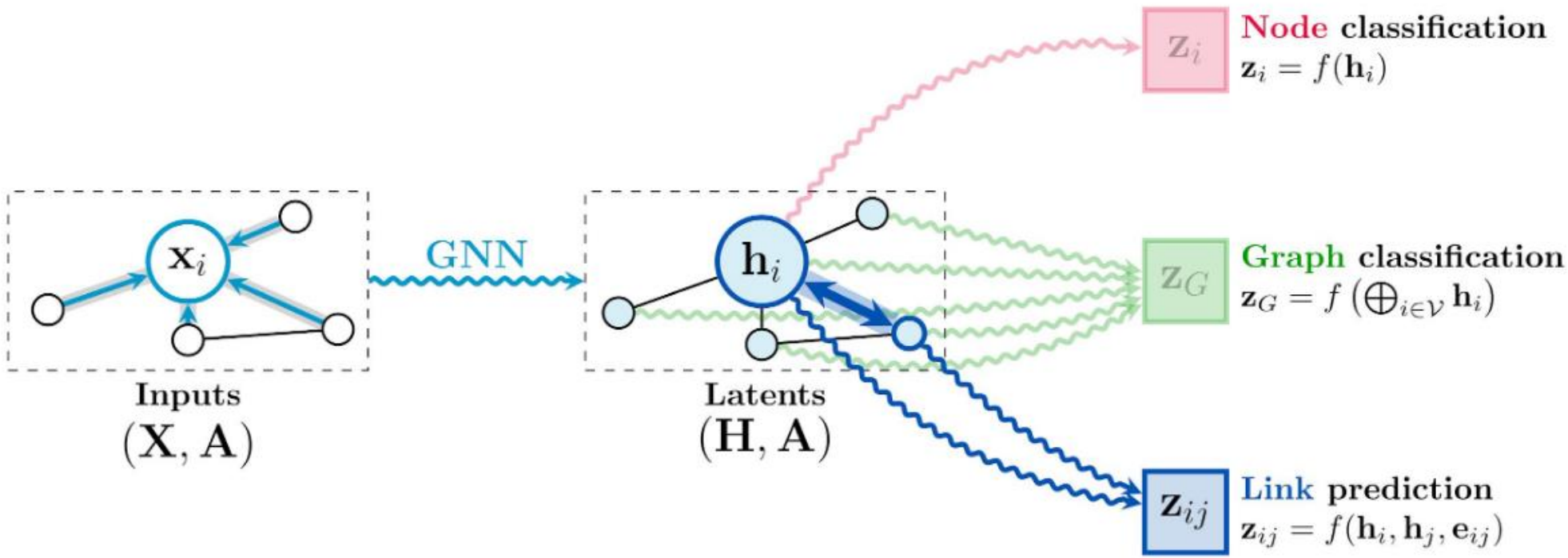
Let N_i be the set of neighbors of node i

$$X_{N_i} = \{\{x_j | j \in N_i\}\}$$

We can now define a local function that operates in the neighborhood only: $g(x_i, X_{N_i})$

$$f(\mathbf{X}, \mathbf{A}) = \begin{bmatrix} — & g(\mathbf{x}_1, \mathbf{X}_{\mathcal{N}_1}) & — \\ — & g(\mathbf{x}_2, \mathbf{X}_{\mathcal{N}_2}) & — \\ & \vdots & \\ — & g(\mathbf{x}_n, \mathbf{X}_{\mathcal{N}_n}) & — \end{bmatrix}$$


$\mathbf{x}_a$
$\mathbf{x}_b$
$\mathbf{x}_c$
$\mathbf{x}_d$
$\mathbf{x}_e$
$g(\mathbf{x}_b, \mathbf{X}_{\mathcal{N}_b})$
$\mathbf{h}_b$


$\mathbf{x}_i$
GNN
$\mathbf{h}_i$
$\mathbf{z}_i$
Node classification
$\mathbf{z}_i = f(\mathbf{h}_i)$
$\mathbf{z}_G$
Graph classification
$\mathbf{z}_G = f\left(\bigoplus_{i \in \mathcal{V}} \mathbf{h}_i\right)$
$\mathbf{z}_{ij}$
Link prediction
$\mathbf{z}_{ij} = f(\mathbf{h}_i, \mathbf{h}_j, \mathbf{e}_{ij})$
Inputs
$(\mathbf{X}, \mathbf{A})$
Latents
$(\mathbf{H}, \mathbf{A})$

Message Passing Framework

Design Space:

1. message function
2. aggregation function
3. update function

$$h_i = \phi\left(x_i, \bigoplus_{j \in N_i} \psi(x_i, x_j)\right)$$

Adding edge features

$$h_i = \phi\left(x_i, \bigoplus_{j \in N_i} \psi(x_i, x_j)\right)$$

$$h_i = \phi\left(x_i, \bigoplus_{j \in N_i} \psi(x_i, x_j, e_{ij})\right)$$

https://www.youtube.com/watch?v=uF53xsT7mjc

https://petar-v.com/talks/GNN-Wednesday.pdf