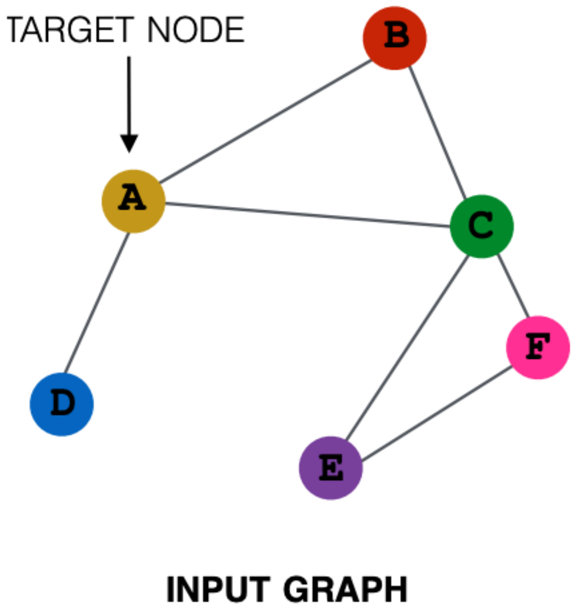
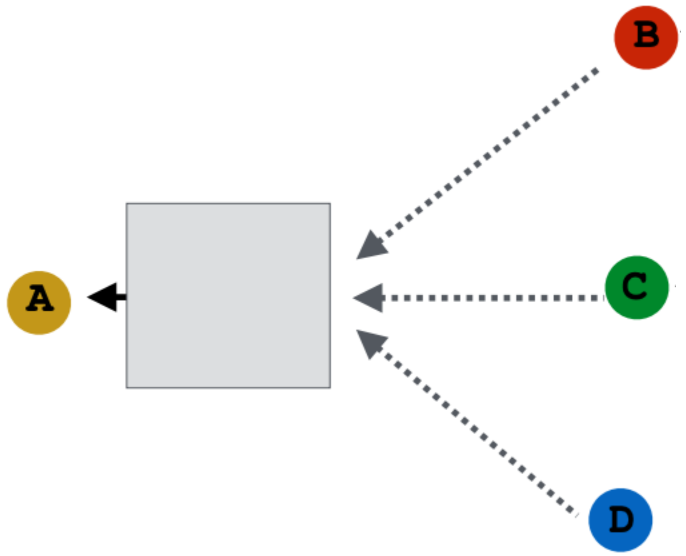
Graph Neural Networks (contd.)

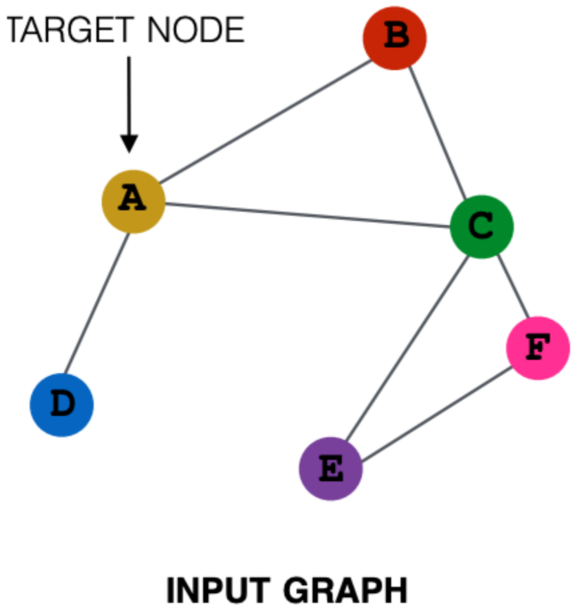
Neural Networks Design And Application

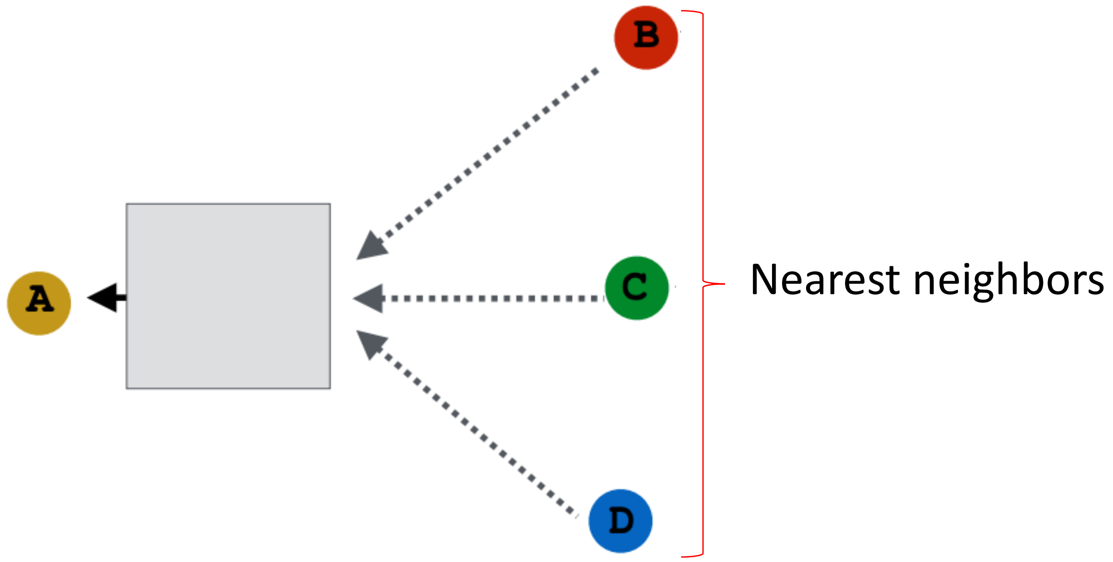
Graph networks: aggregate neighbors



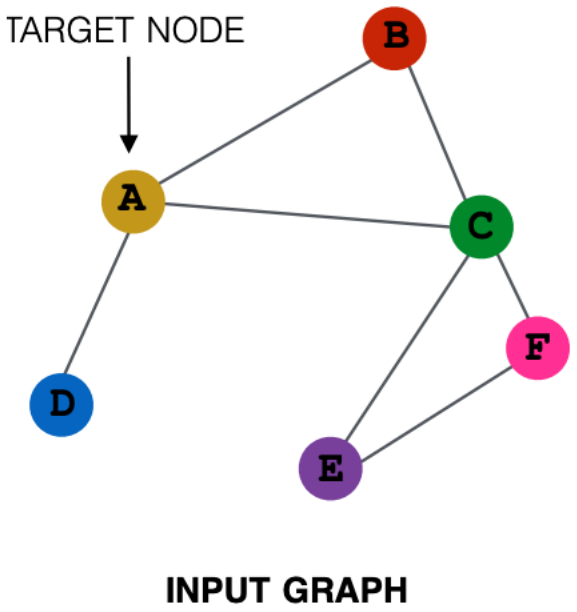


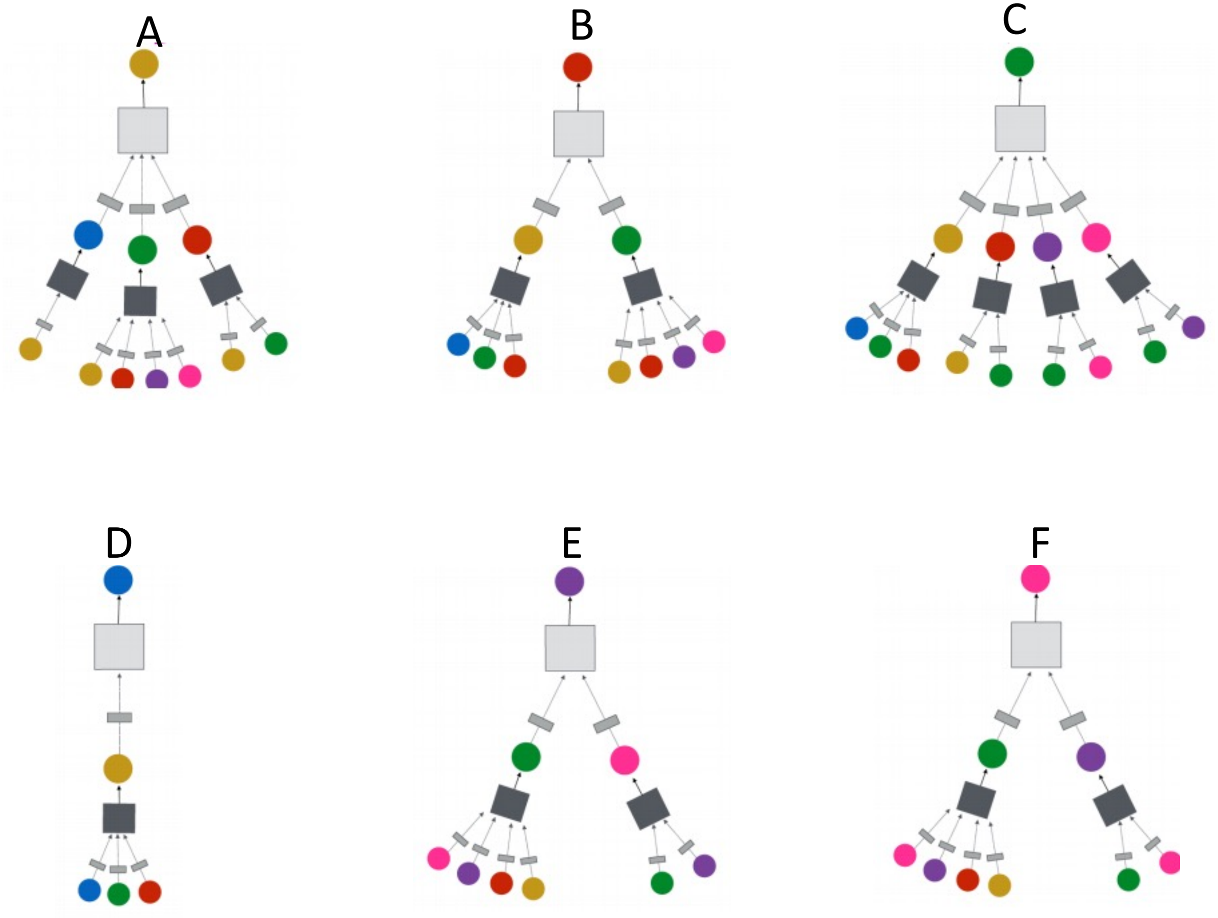
Graph networks: aggregate neighbors



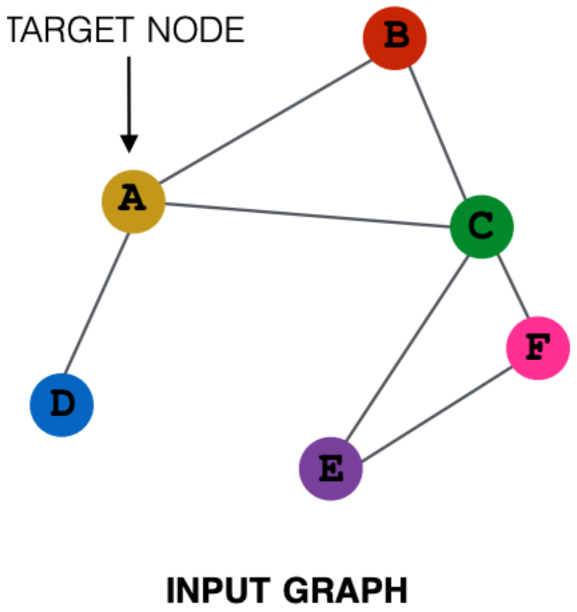


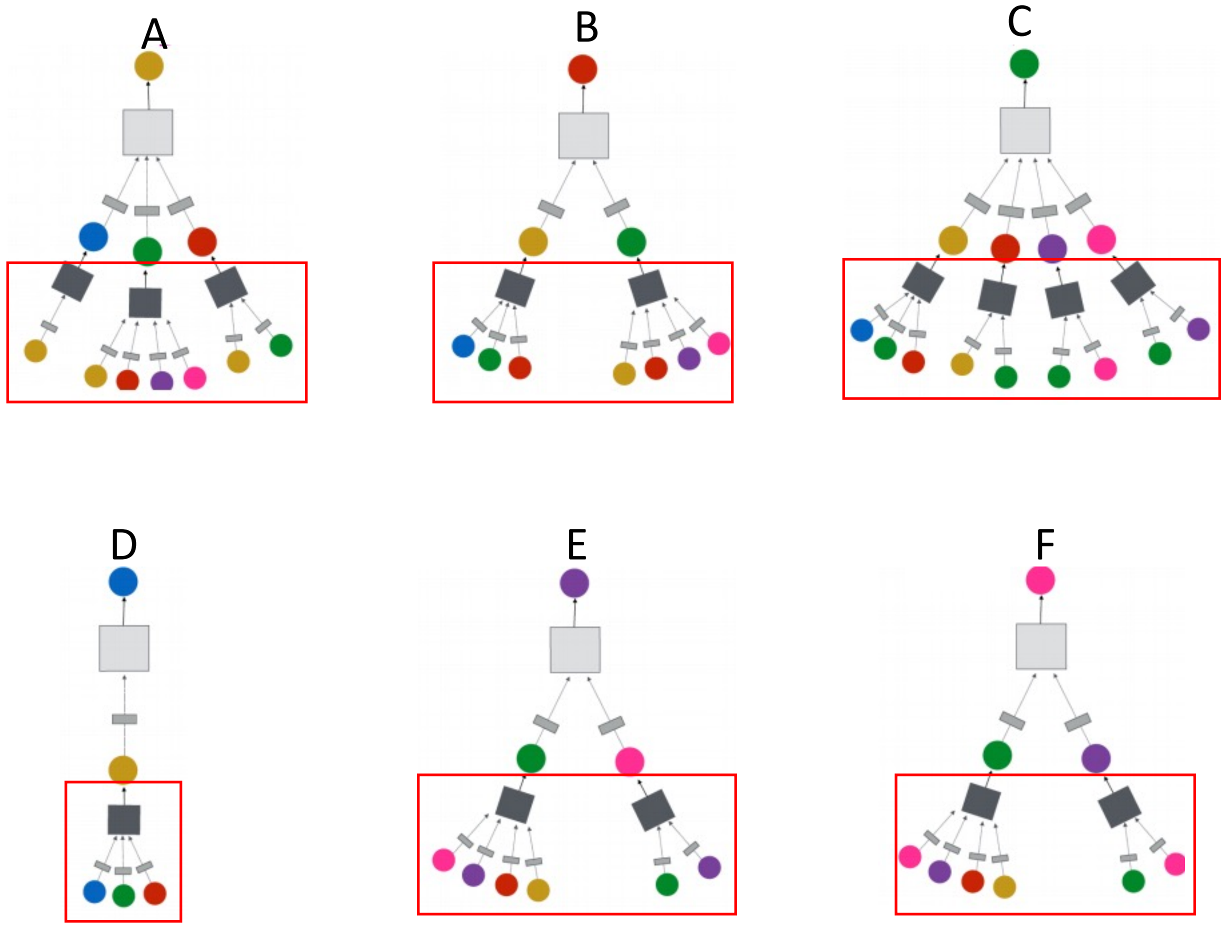
Graph networks: aggregate neighbors





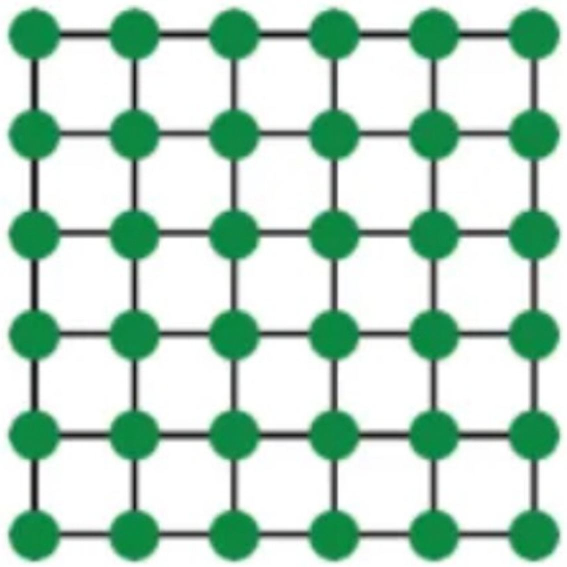
Graph networks: aggregate neighbors





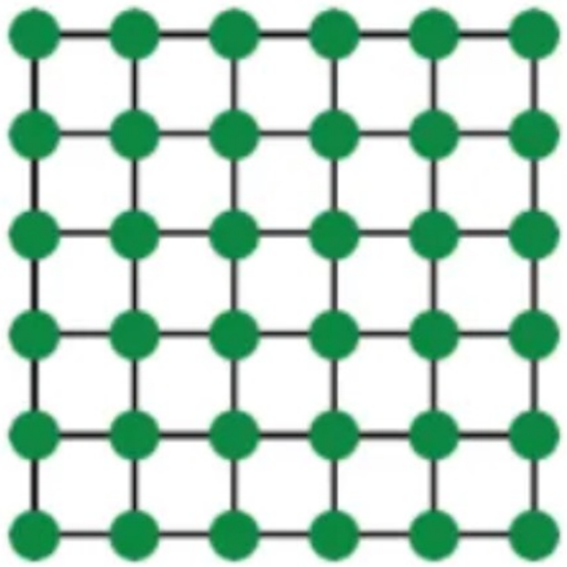
Two hops (two nearest neighbors)

Graph networks: aggregate neighbors

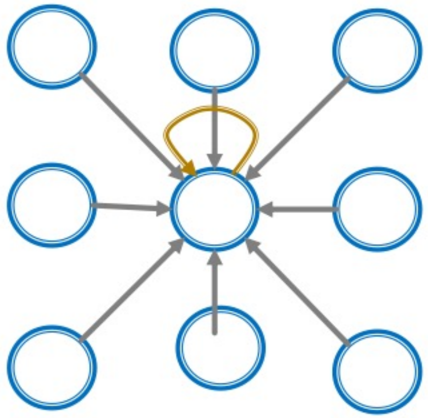


Grid graph

Graph networks: aggregate neighbors



Grid graph



General graph

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\boxed{h_v^0 = x_v}$$

embedding of v at layer l

$$h_v^{(l+1)} = \boxed{\sigma}\left(W_l \boxed{\sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|}} + B_l \boxed{h_v^{(l)}}\right), \forall l \in \{0, \ldots, \boxed{L} - 1\}$$

$$\boxed{z_v} = h_v^{(L)}$$

Average of neighbor's previous layer embeddings

Total number of layers

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\boxed{\mathrm{h}_v^0 = \mathrm{x}_v}$$

embedding of v at layer l

$$\mathrm{h}_v^{(l+1)} = \boxed{\sigma}\Big(\mathrm{W}_l \boxed{\sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|}} + \mathrm{B}_l \boxed{\mathrm{h}_v^{(l)}}\Big), \forall l \in \{0, \dots, \boxed{L} - 1\}$$

$$\boxed{\mathrm{z}_v} = \mathrm{h}_v^{(L)}$$

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Average of neighbor's previous layer embeddings

Total number of layers

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\mathrm{h}_v^0 = \mathrm{x}_v$$

embedding of v at layer l

$$\mathrm{h}_v^{(l+1)} = \sigma\left(\mathrm{W}_l \sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|} + \mathrm{B}_l \mathrm{h}_v^{(l)}\right), \forall l \in \{0, \ldots, L-1\}$$

$$\mathrm{z}_v = \mathrm{h}_v^{(L)}$$

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Average of neighbor's previous layer embeddings

Total number of layers

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$h_v^0 = x_v$$

embedding of v at layer l

$$h_v^{(l+1)} = \sigma\left(W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l h_v^{(l)}\right), \forall l \in \{0, \dots, L-1\}$$

$$z_v = h_v^{(L)}$$

Average of neighbor's previous layer embeddings

Total number of layers

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\boxed{h_v^0 = x_v}$$

embedding of v at layer l

$$h_v^{(l+1)} = \boxed{\sigma}\left(\boxed{W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|}} + B_l \boxed{h_v^{(l)}}\right), \forall l \in \{0, \ldots, \boxed{L} - 1\}$$

$$\boxed{z_v} = h_v^{(L)}$$

Average of neighbor's previous layer embeddings

Total number of layers

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Q: instead of averaging, can we propose a general way to aggregate neighbors?

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\mathrm{h}_v^0 = \mathrm{x}_v$$

embedding of v at layer l

$$\mathrm{h}_v^{(l+1)} = \boxed{\sigma}\left(\mathrm{W}_l \boxed{\sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|}} + \mathrm{B}_l \boxed{\mathrm{h}_v^{(l)}}\right), \forall l \in \{0, \dots, \boxed{L} - 1\}$$

GraphSAGE

$$\boxed{\mathrm{h}_v^{(l+1)} = \sigma\left(\left[\mathrm{W}_l \cdot \mathrm{AGG}\left(\left\{\mathrm{h}_u^{(l)}, \forall u \in N(v)\right\}\right), \mathrm{B}_l \mathrm{h}_v^{(l)}\right]\right)}$$

otal number of layers

layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\boxed{h_v^0 = x_v}$$

embedding of v at layer l

$$h_v^{(l+1)} = \boxed{\sigma}\left(W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l \boxed{h_v^{(l)}}\right), \forall l \in \{0, \dots, \boxed{L} - 1\}$$

otal number of layers

GraphSAGE

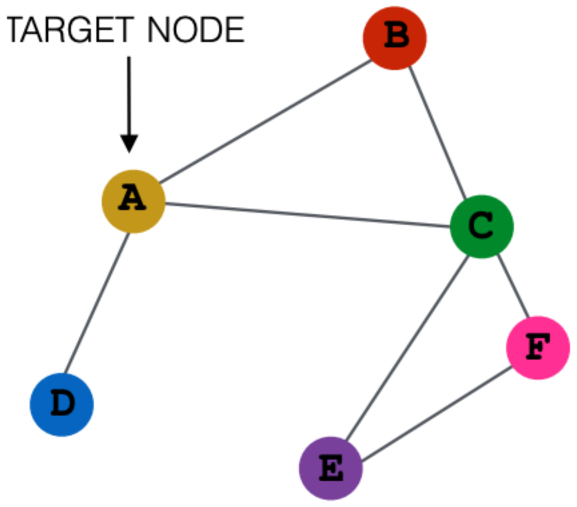
$$h_v^{(l+1)} = \sigma([W_l \cdot \mathrm{AGG}\left(\left\{h_u^{(l)}, \forall u \in N(v)\right\}\right), B_l h_v^{(l)}])$$

layers of neighborhood aggregation

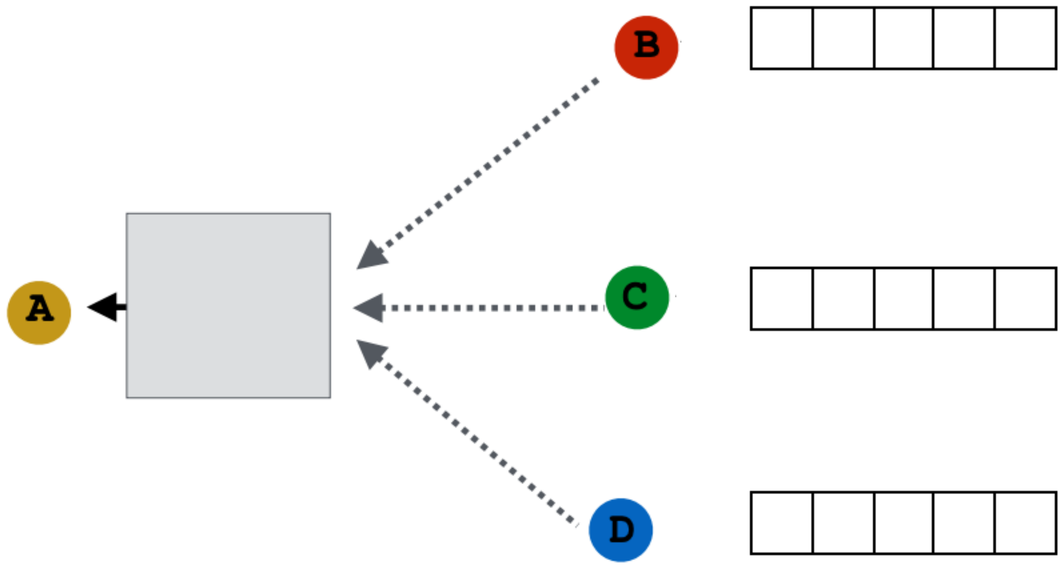
Non-linearity (e.g., ReLU)

Q: why GraphSAGE can be more general?

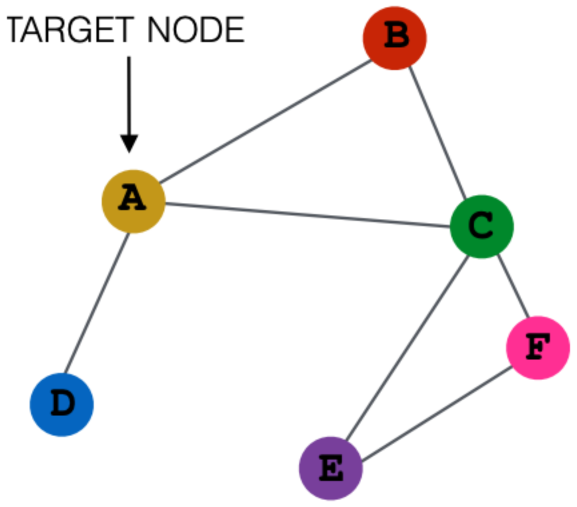
Graph networks: aggregate neighbors



INPUT GRAPH

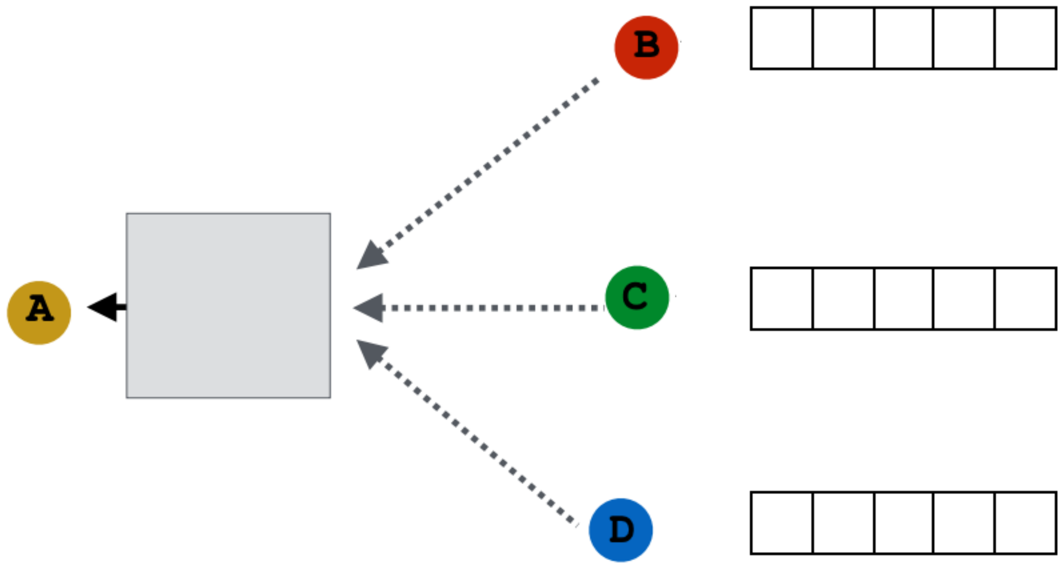


Graph networks: aggregate neighbors

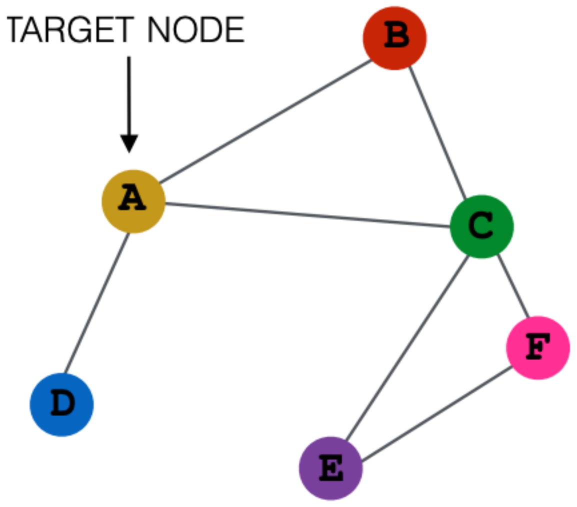


INPUT GRAPH

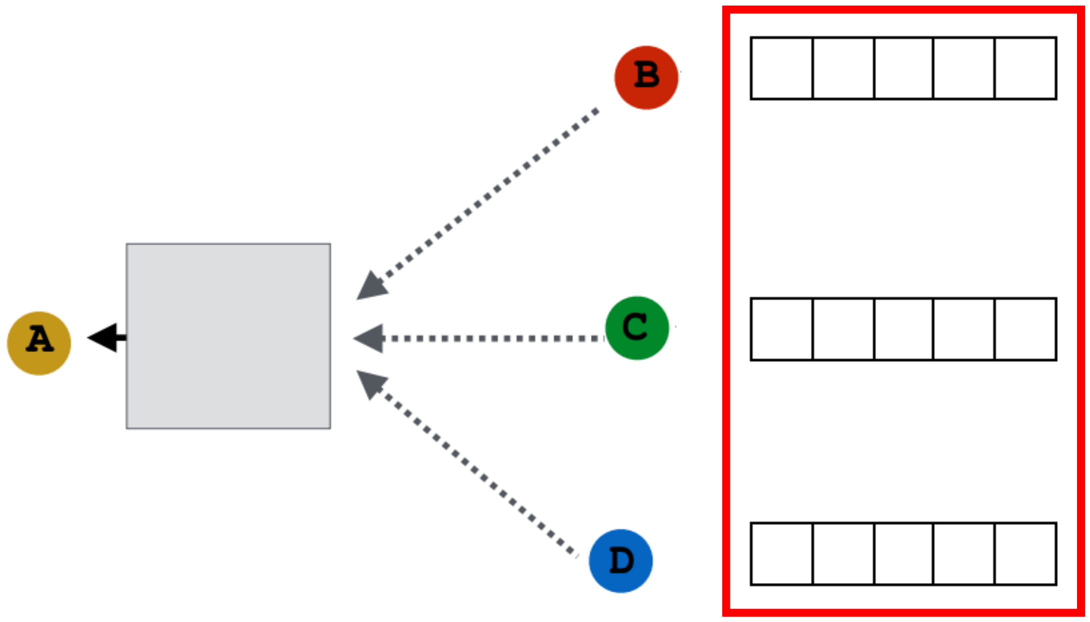
Suppose: we have 5d node features



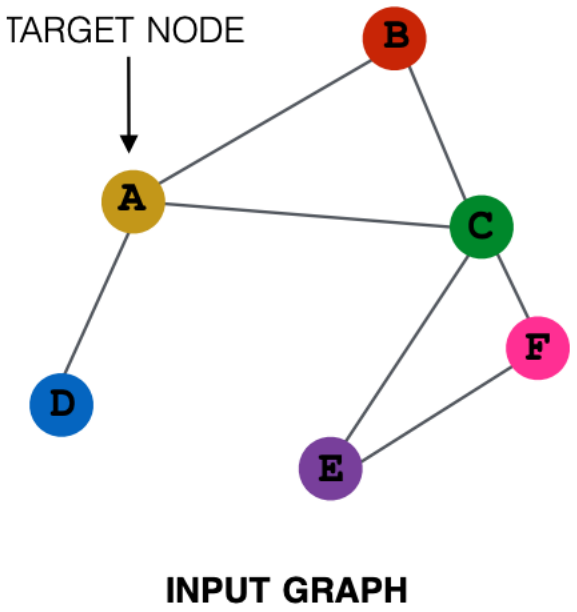
Graph networks: aggregate neighbors

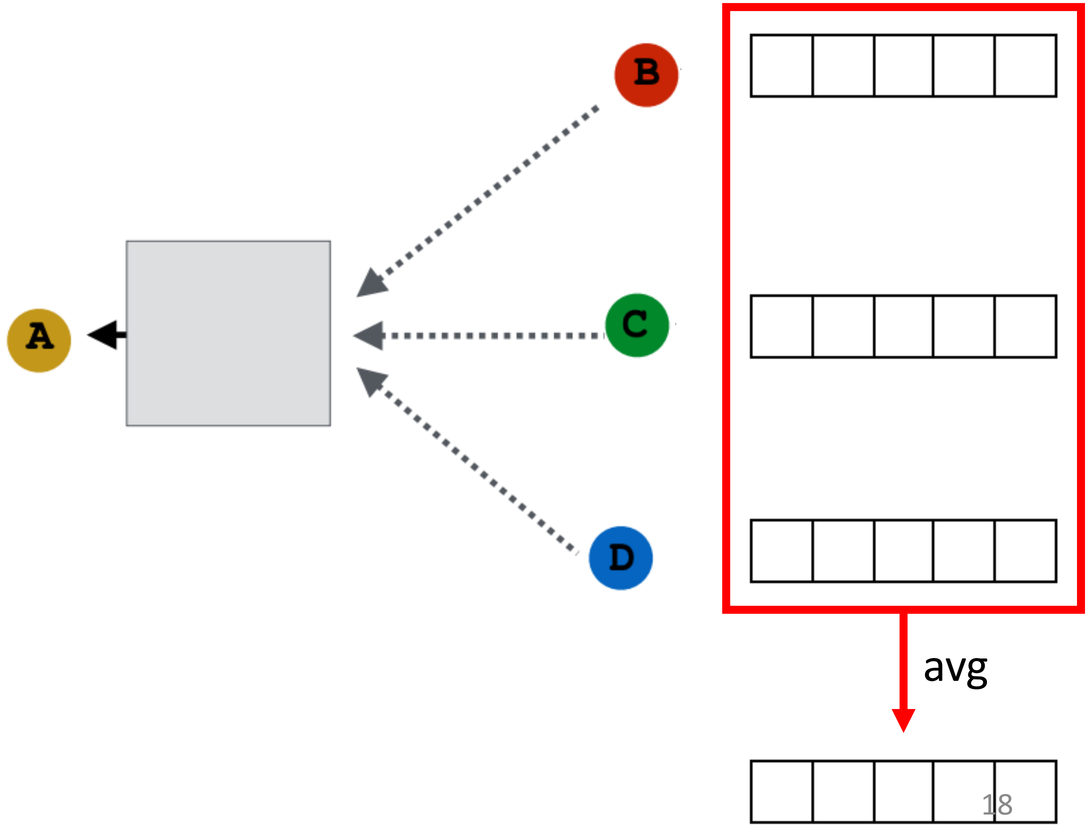


INPUT GRAPH

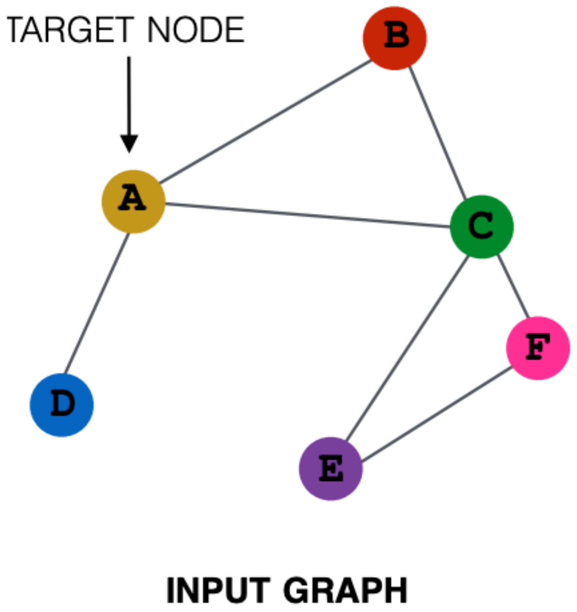


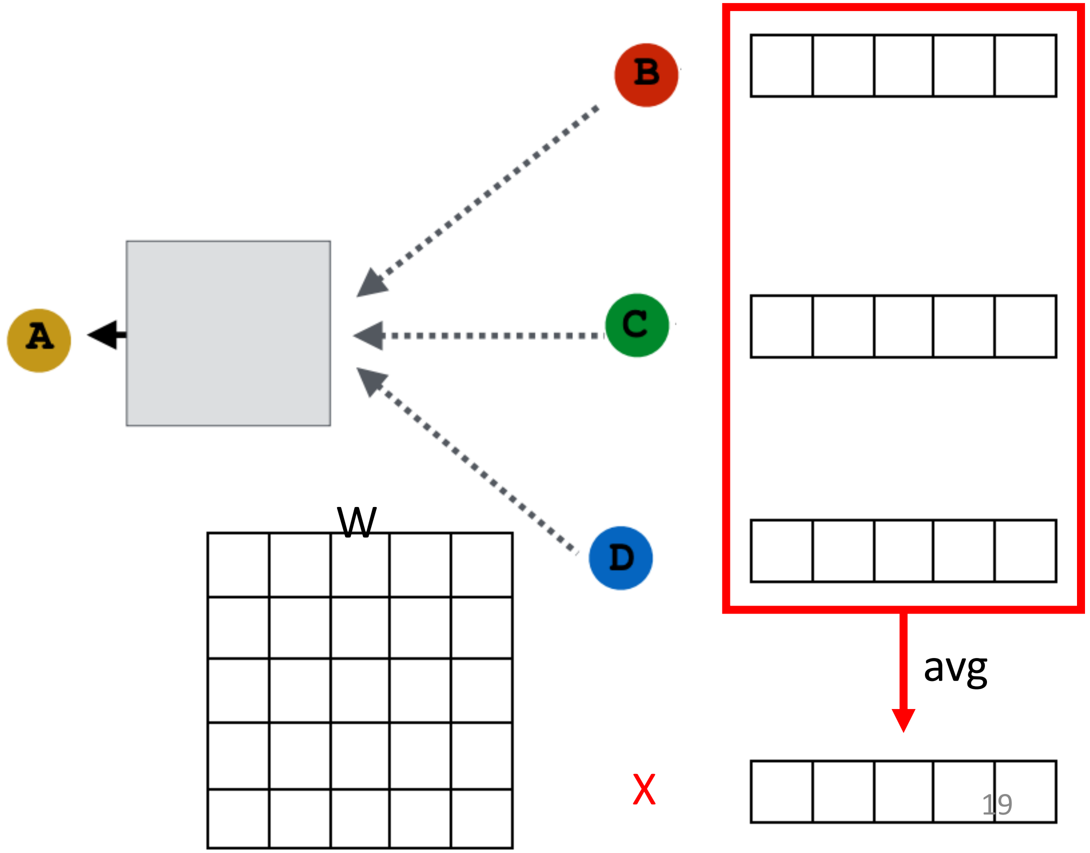
Graph networks: aggregate neighbors



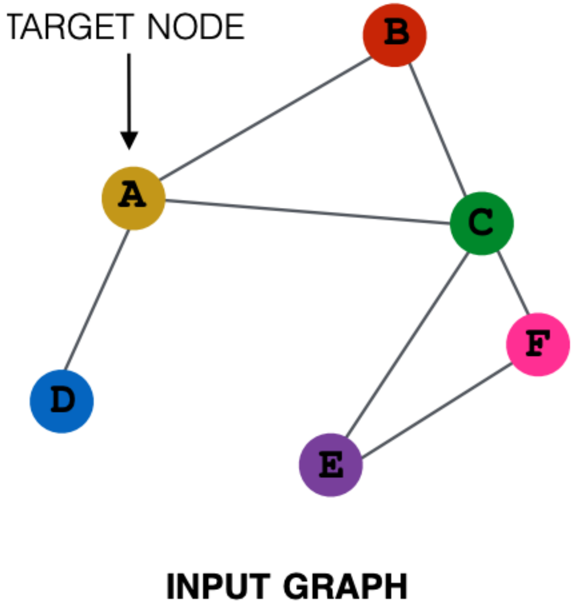


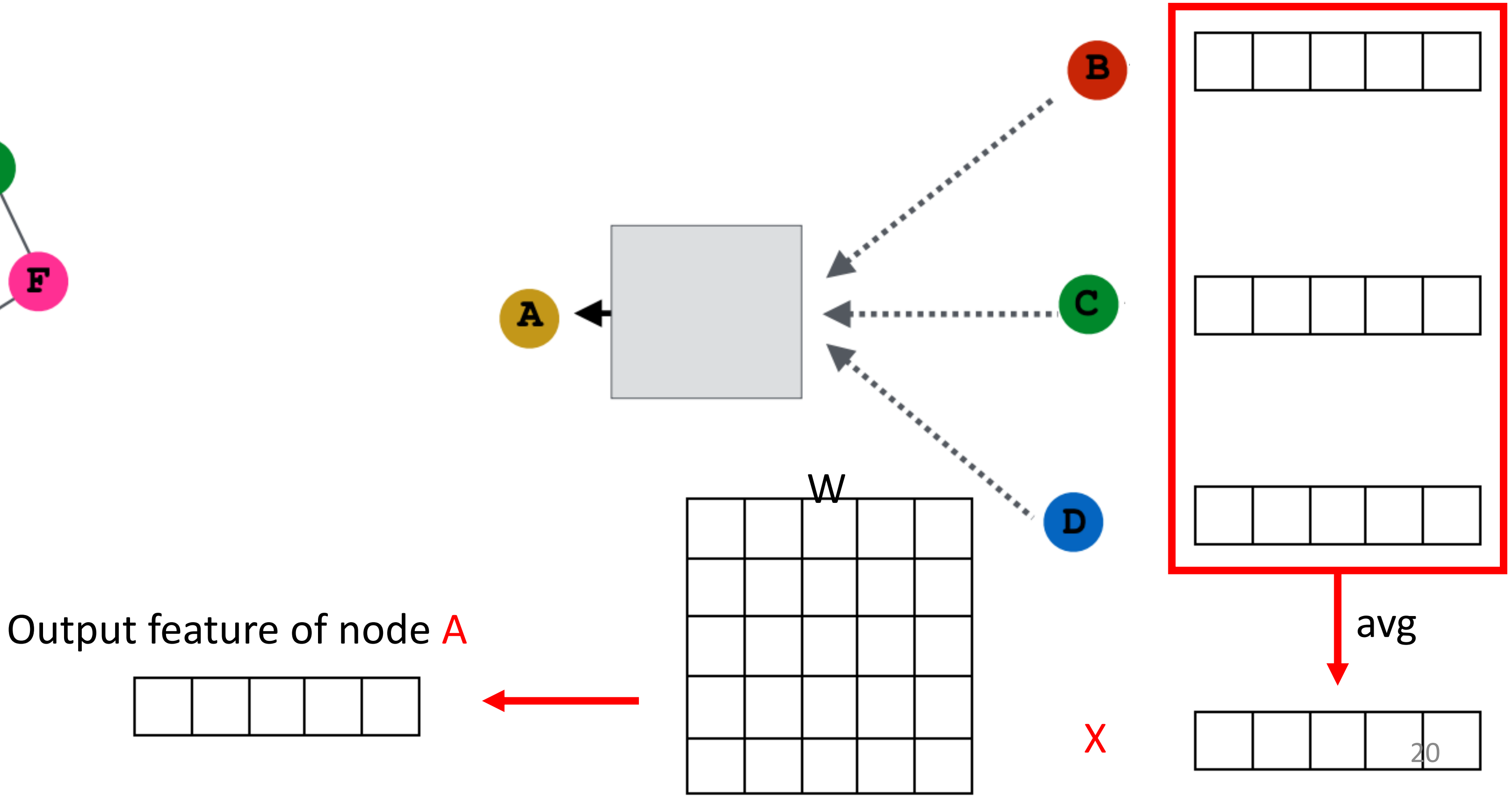
Graph networks: aggregate neighbors



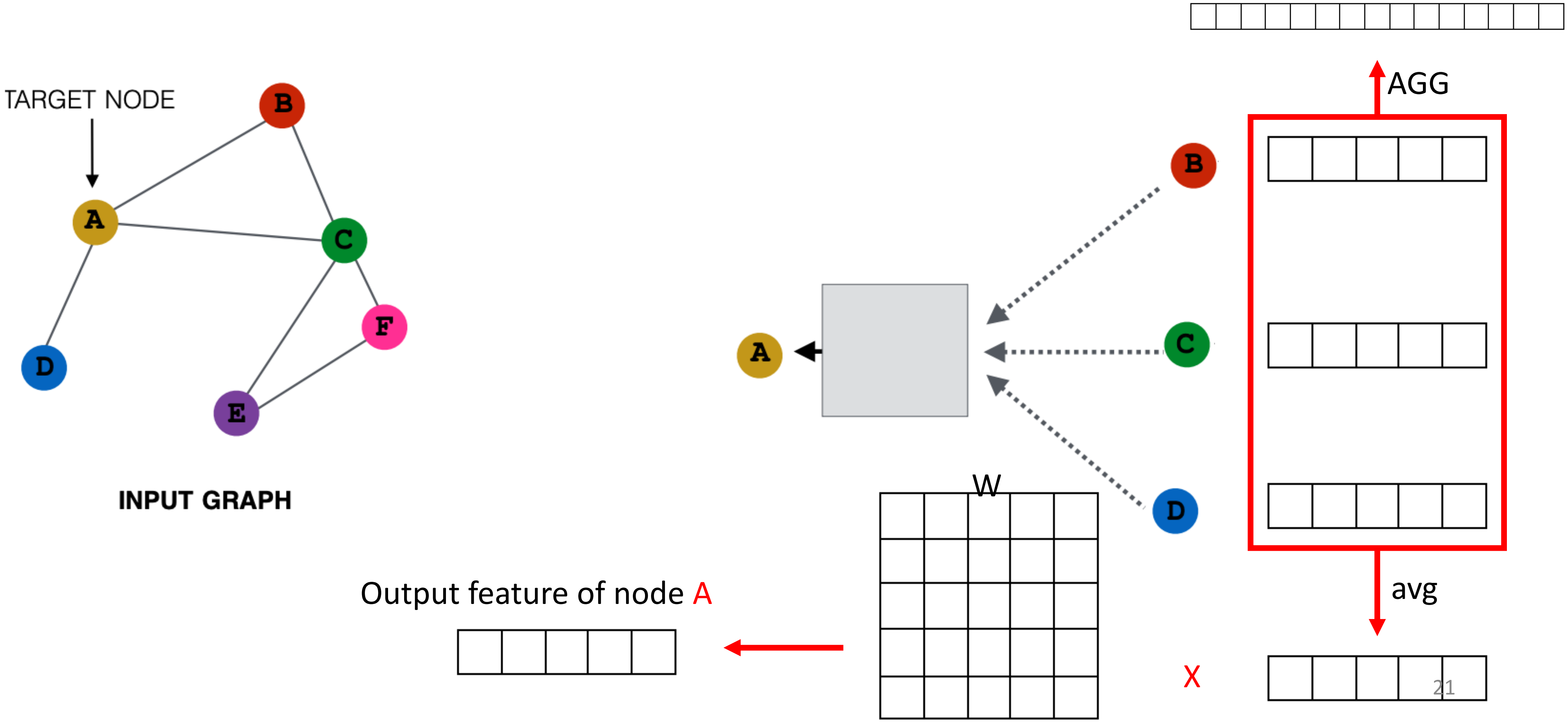


Graph networks: aggregate neighbors

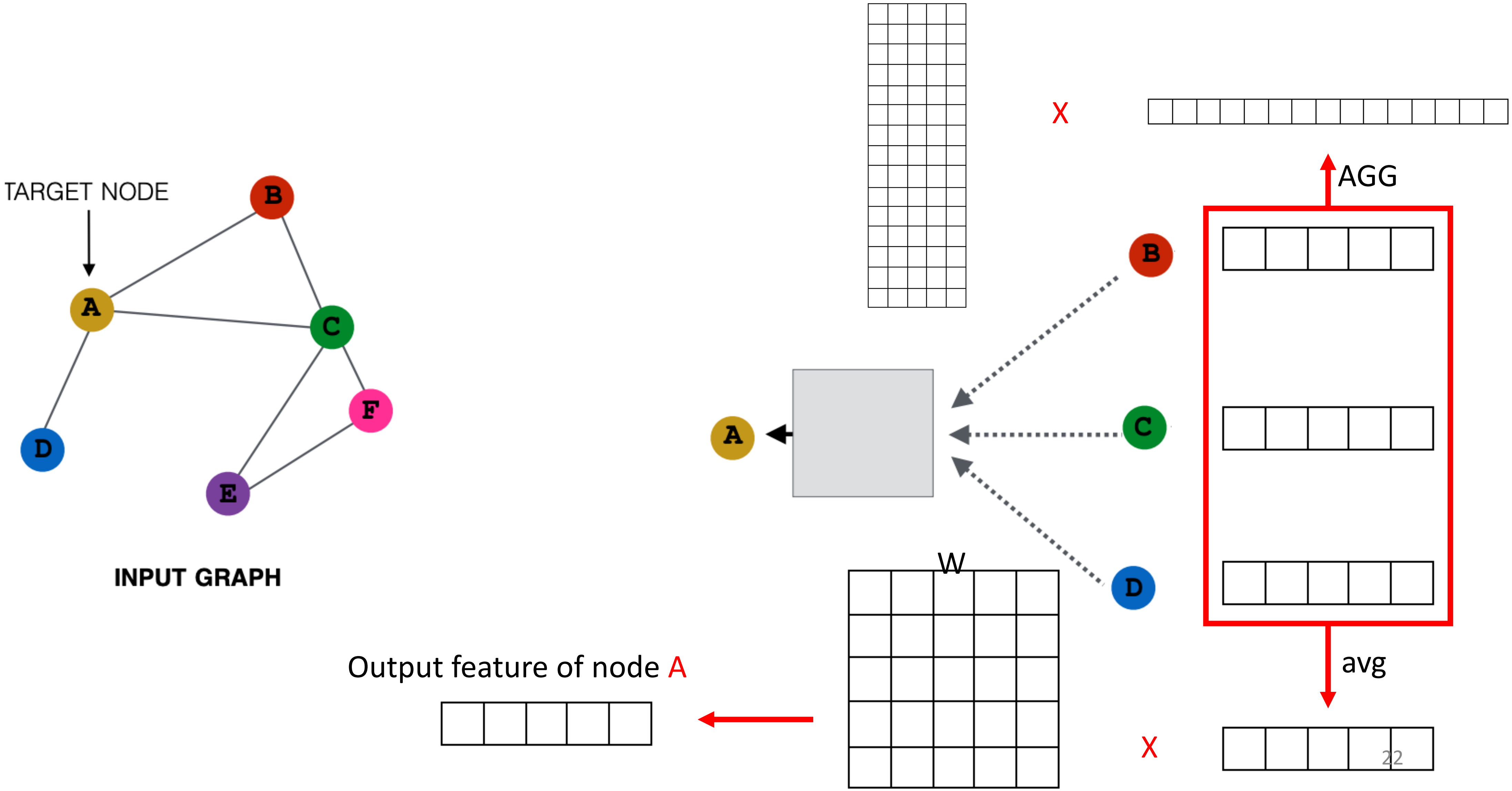




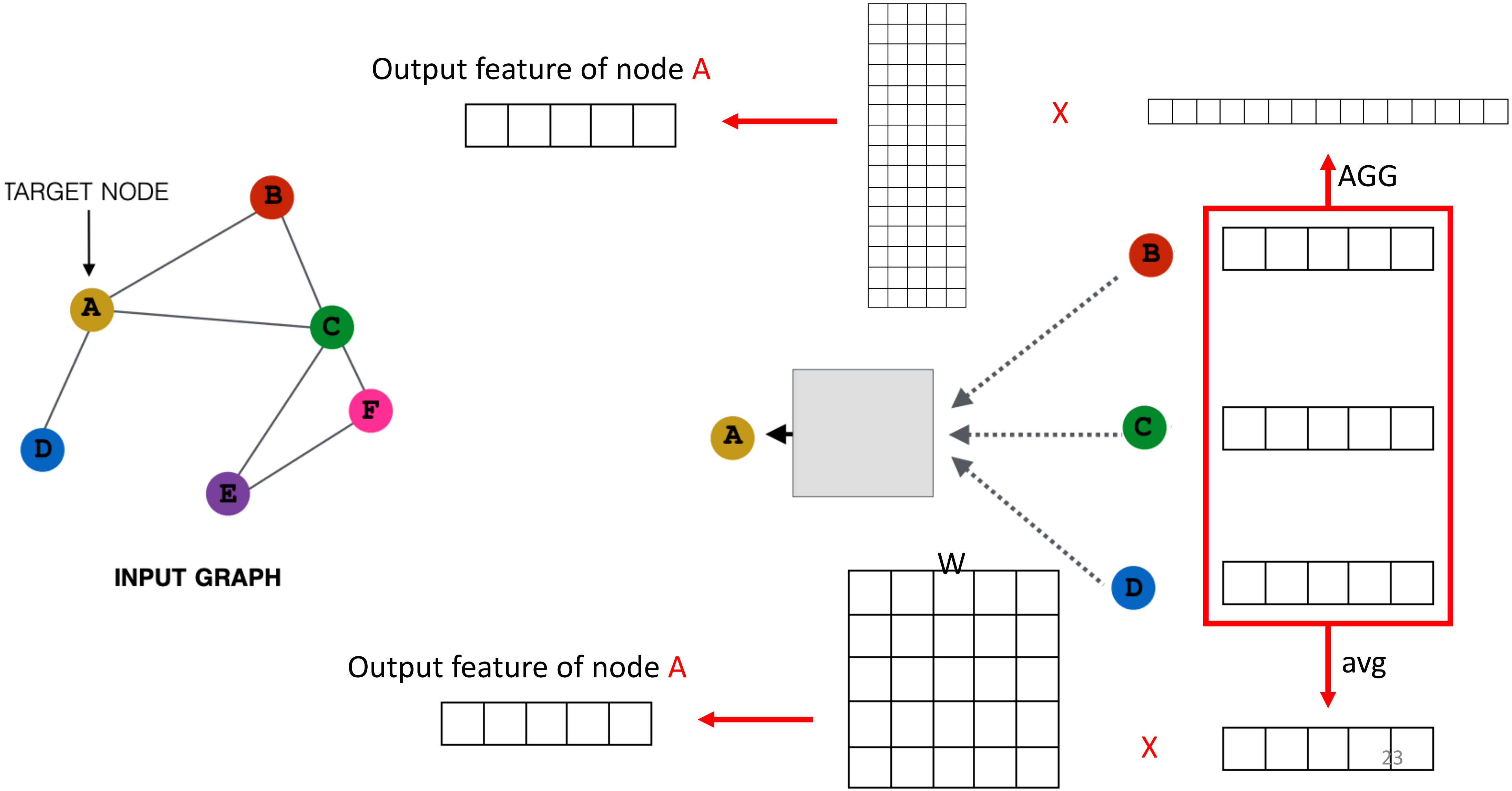
Graph networks: aggregate neighbors



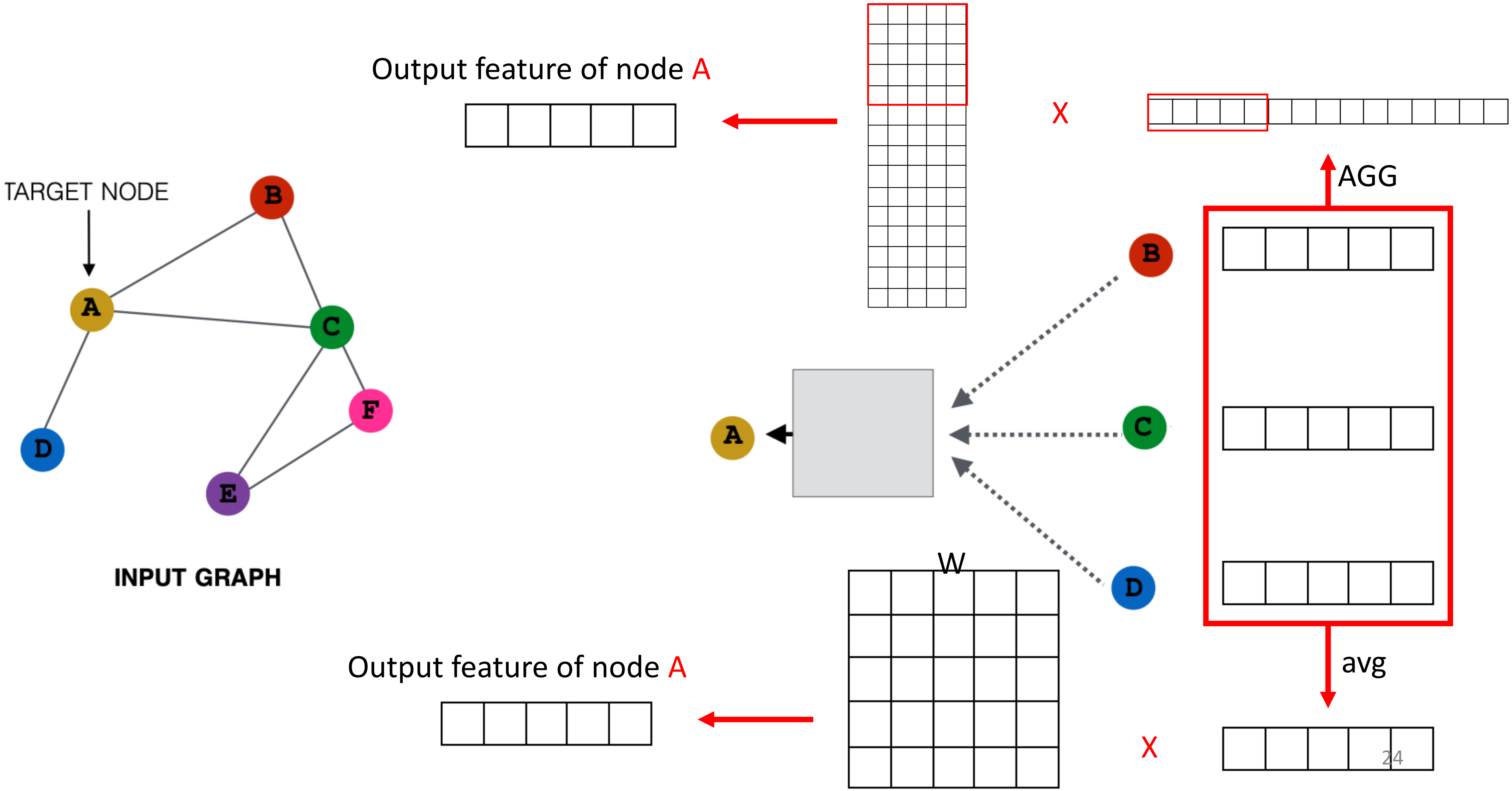
Graph networks: aggregate neighbors



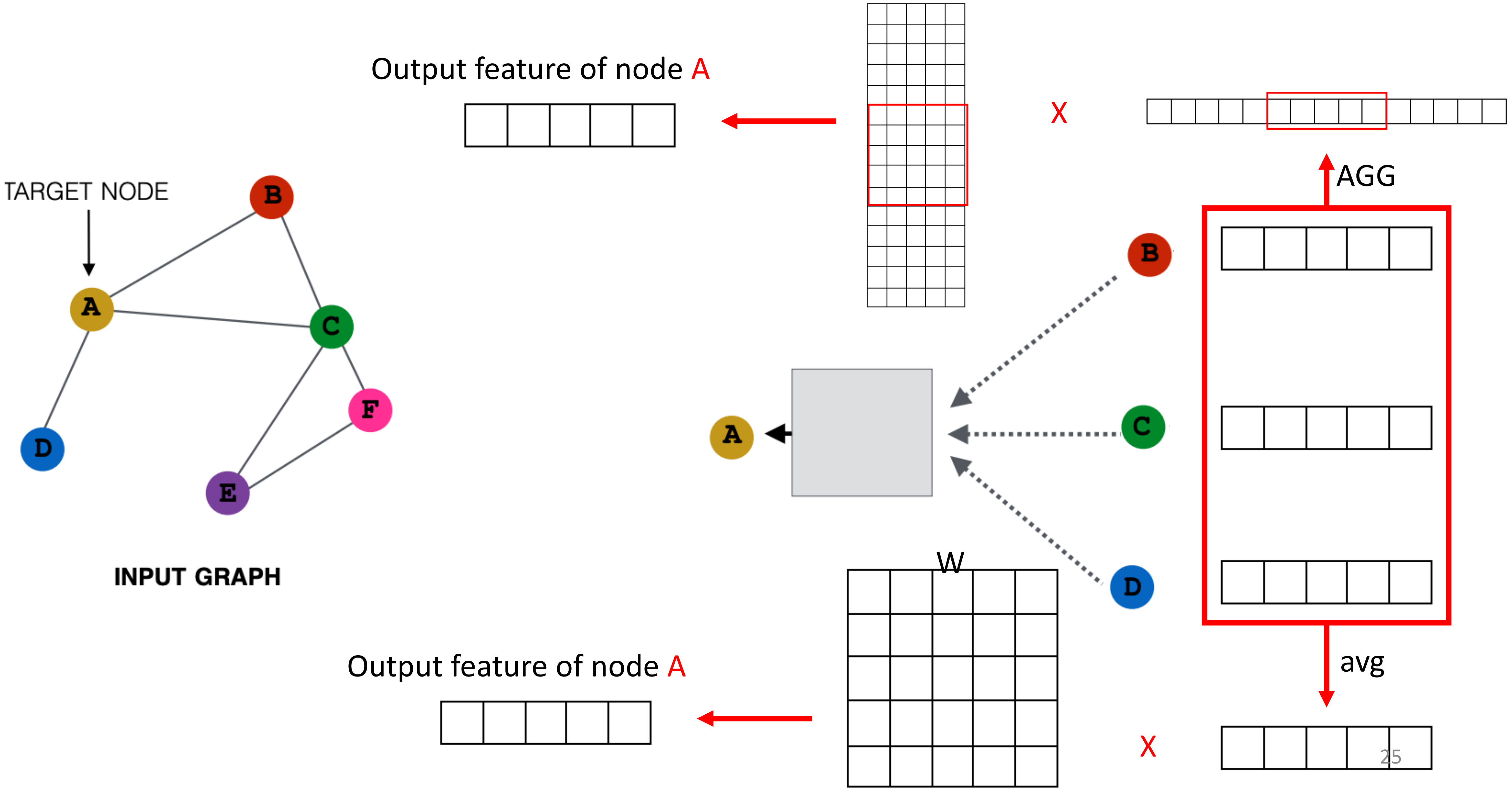
Graph networks: aggregate neighbors



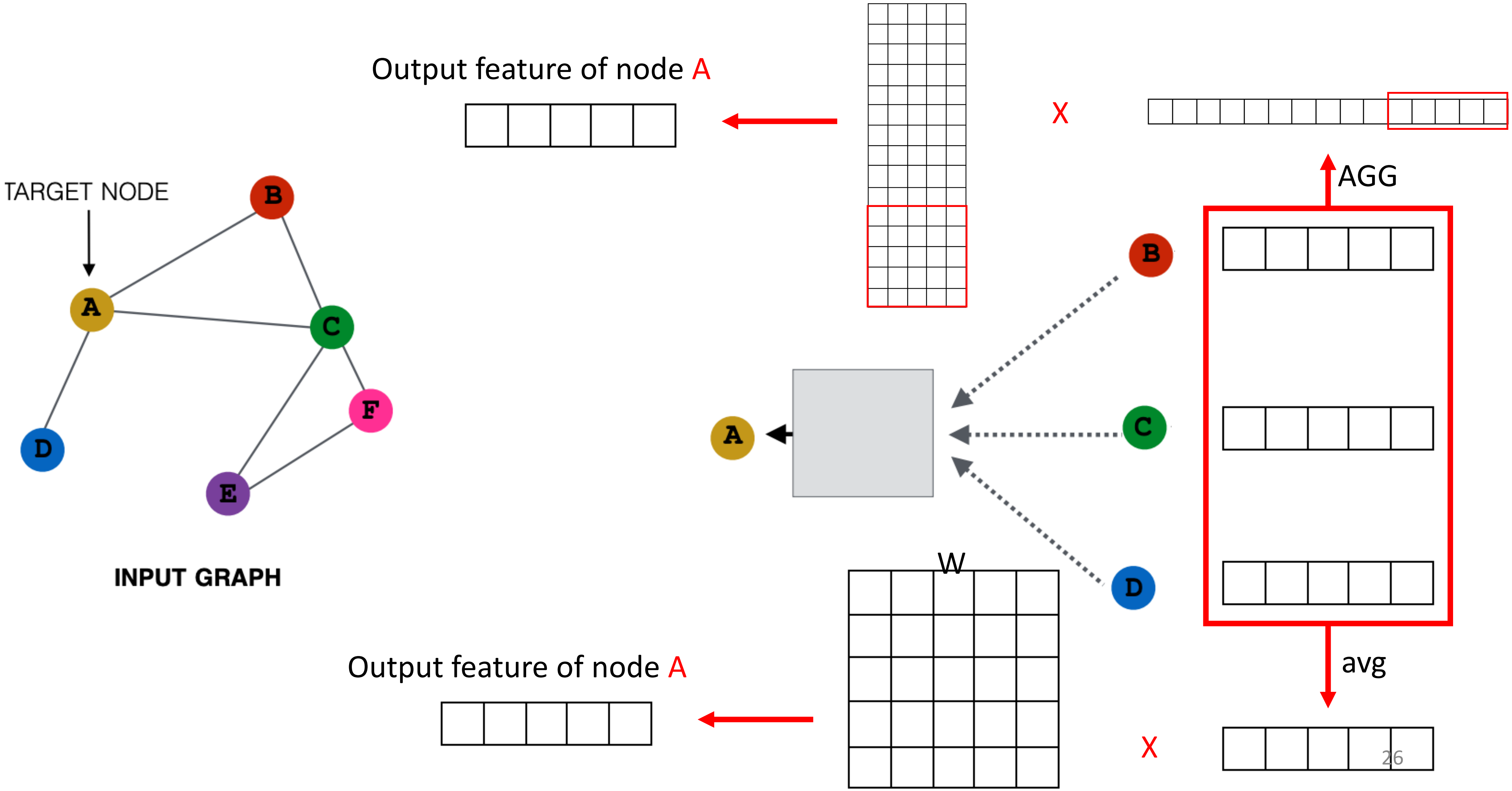
Graph networks: aggregate neighbors



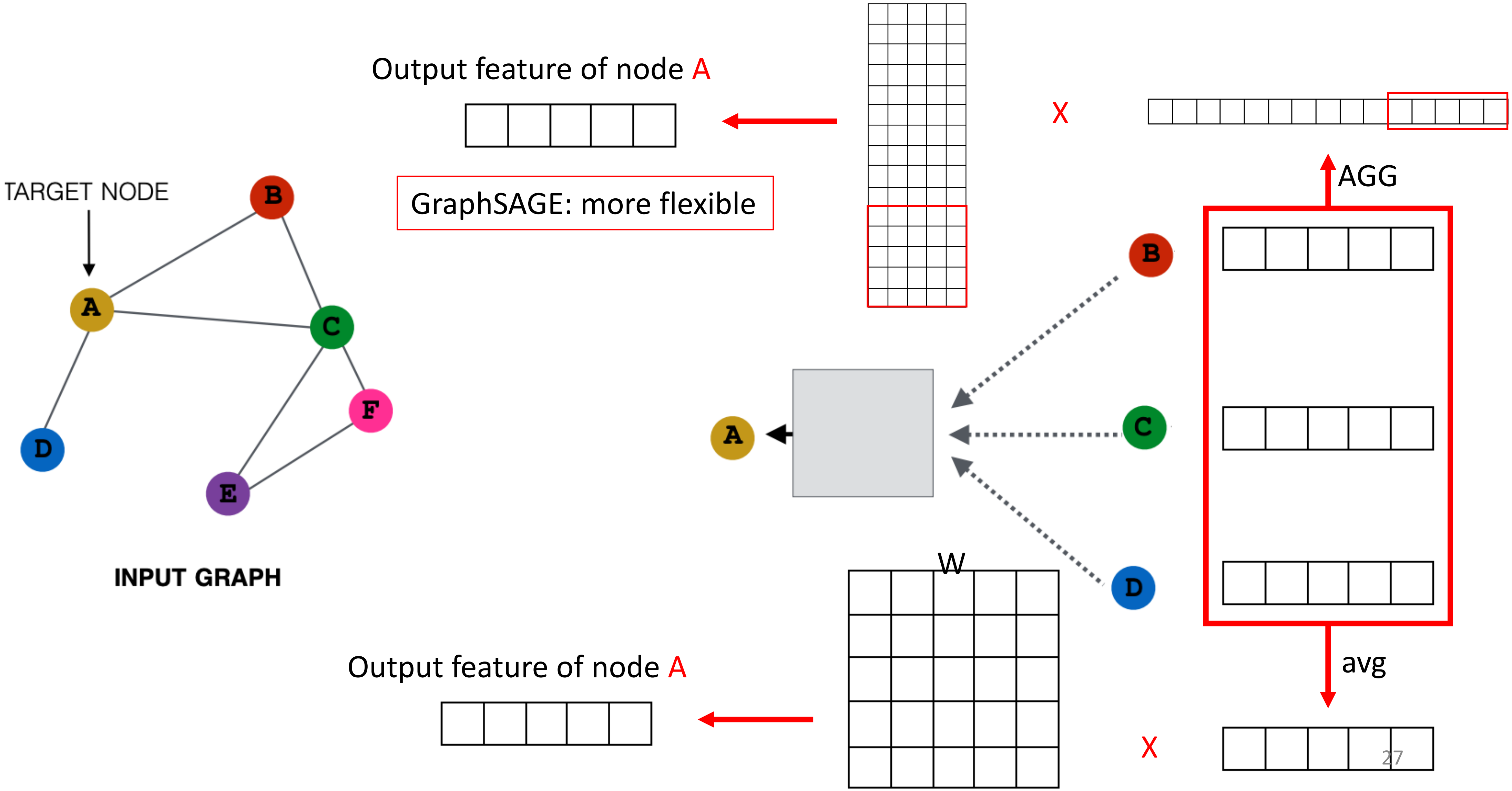
Graph networks: aggregate neighbors



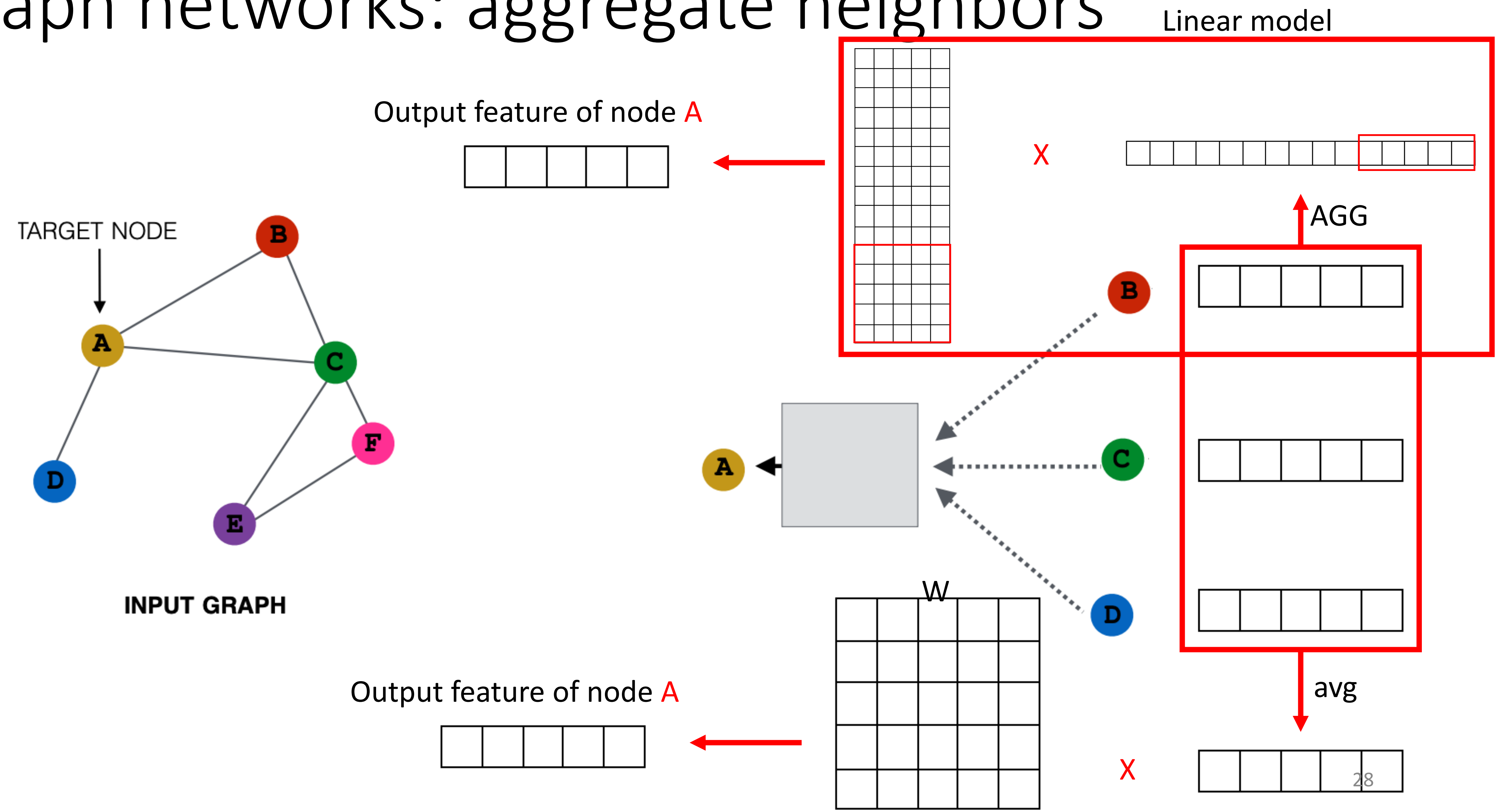
Graph networks: aggregate neighbors



Graph networks: aggregate neighbors

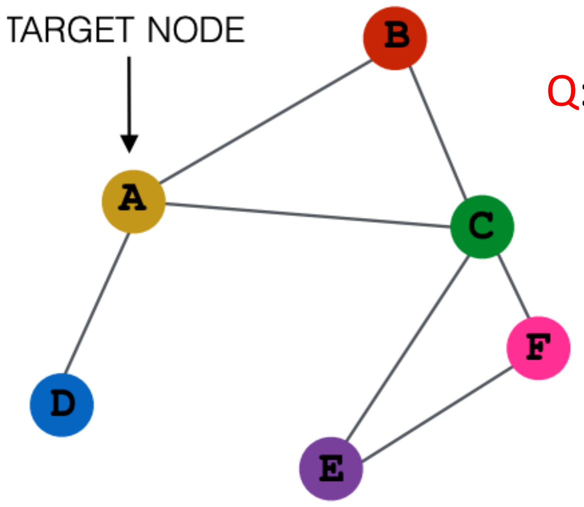


Graph networks: aggregate neighbors

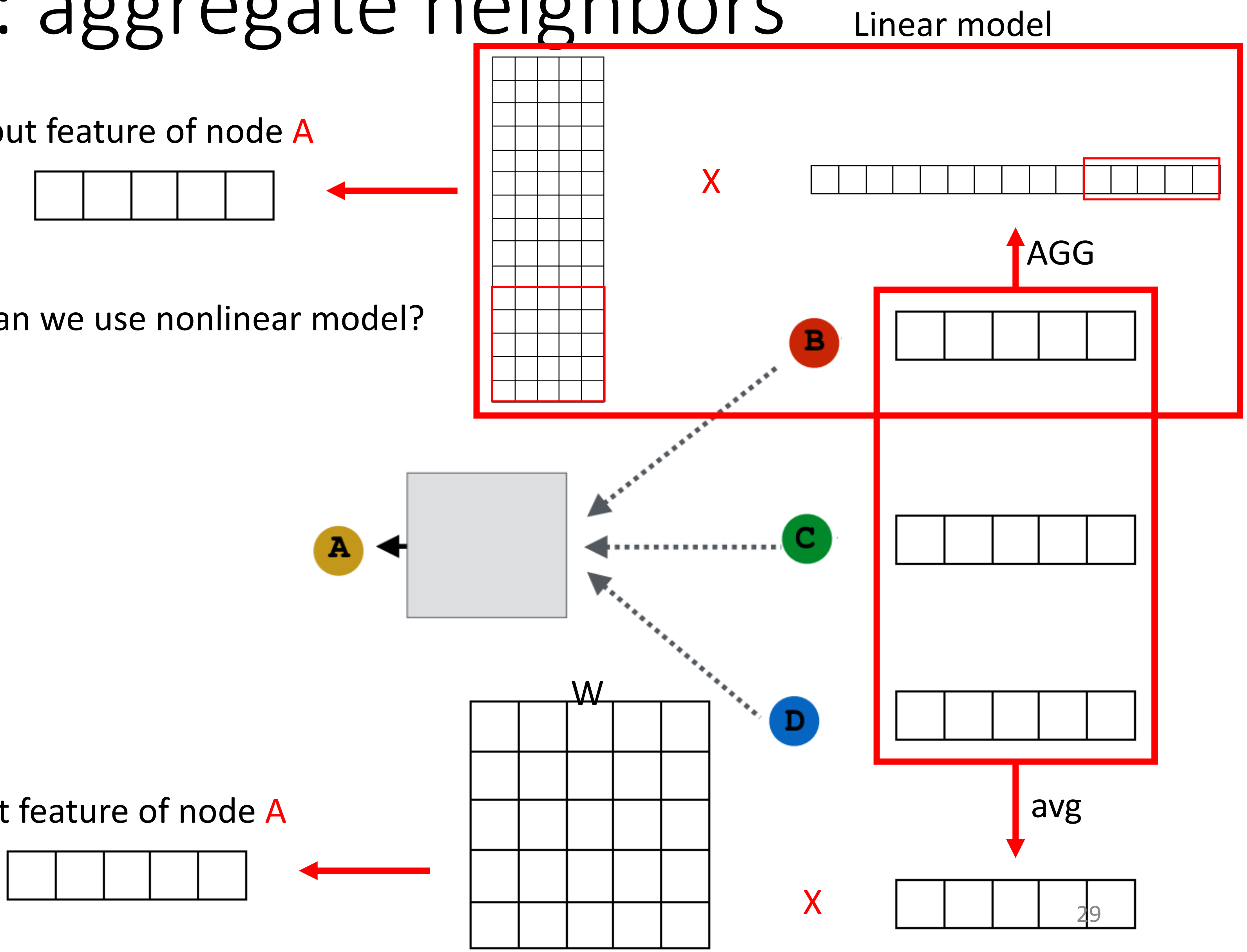


Graph networks: aggregate neighbors

Q: can we use nonlinear model?



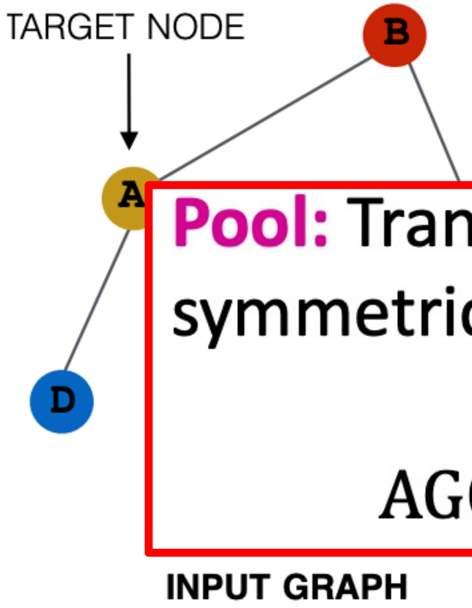
INPUT GRAPH



Graph networks: aggregate neighbors

Output feature of node A

Q: can we use nonlinear model?

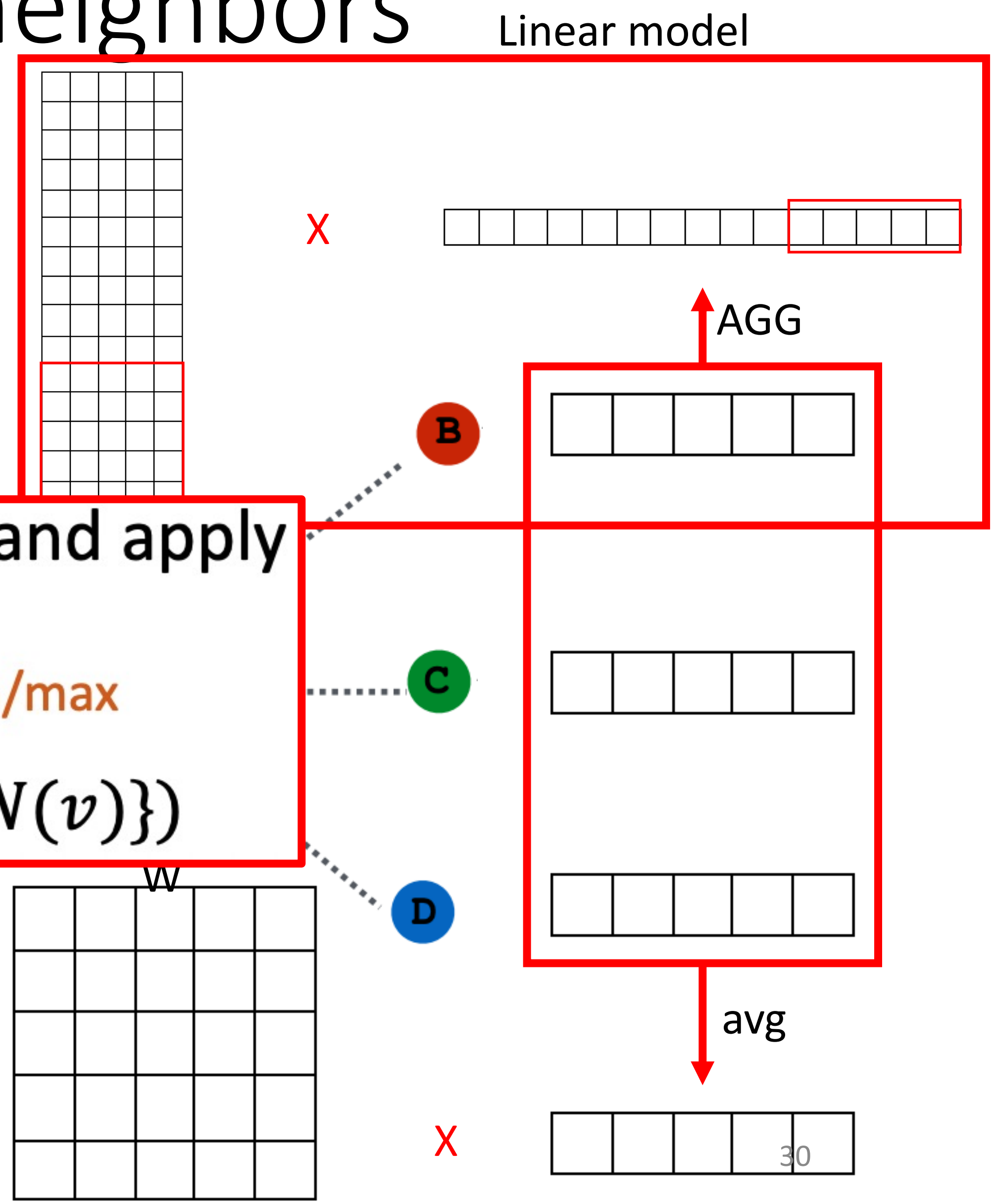


Pool: Transform neighbor vectors and apply symmetric vector function

Element-wise mean/max

$$\mathrm{AGG} = \gamma(\{\mathrm{MLP}(\mathrm{h}_u^{(l)}), \forall u \in N(v)\})$$

Output feature of node A

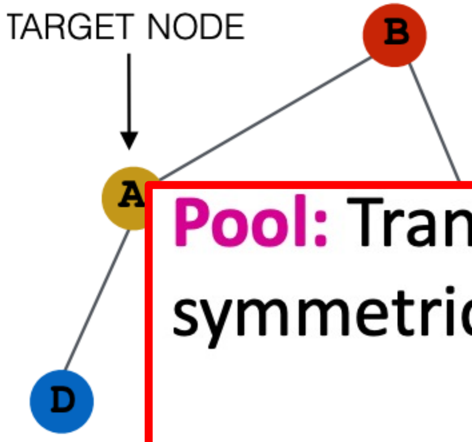


Graph networks: aggregate neighbors

Linear model

Output feature of node A

Q: can we use nonlinear model?

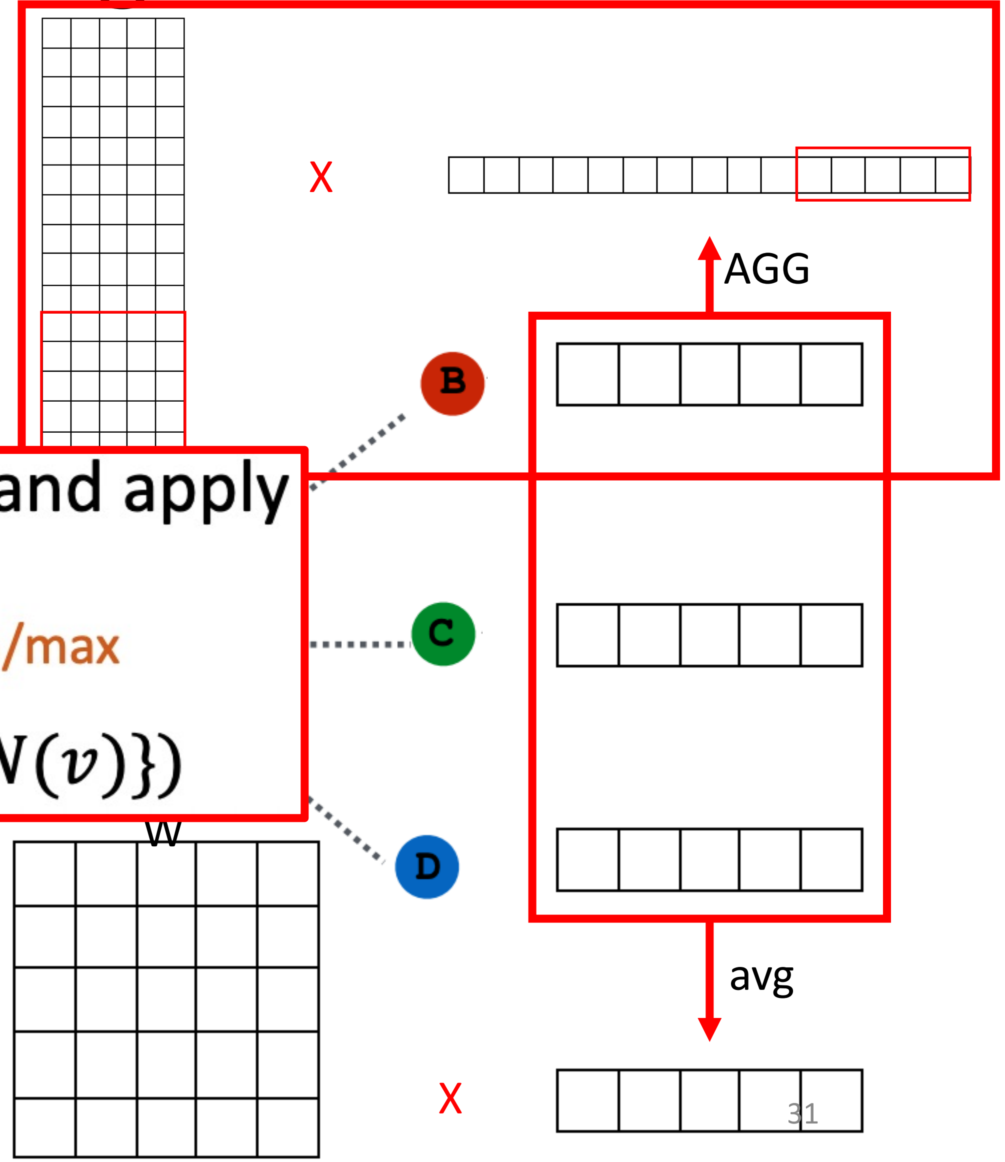


Pool: Transform neighbor vectors and apply symmetric vector function

Element-wise mean/max

$$\mathrm{AGG} = \gamma(\boxed{\{\mathrm{MLP}(\mathrm{h}_u^{(l)}),}\ \forall u \in N(v)\})$$

INPUT GRAPH



Output feature of node A

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\boxed{h_v^0 = x_v}$$

embedding of v at layer l

$$h_v^{(l+1)} = \boxed{\sigma}\left(\boxed{W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|}} + B_l \boxed{h_v^{(l)}}\right), \forall l \in \{0, \dots, \boxed{L} - 1\}$$

$$\boxed{z_v} = h_v^{(L)}$$

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Average of neighbor's previous layer embeddings

Total number of layers

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\boxed{\mathrm{h}_v^0 = \mathrm{x}_v}$$

embedding of v at layer l

$$\mathrm{h}_v^{(l+1)} = \boxed{\sigma}\left(\mathrm{W}_l \boxed{\sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|}} + \mathrm{B}_l \boxed{\mathrm{h}_v^{(l)}}\right), \forall l \in \{0, \dots, \boxed{L} - 1\}$$

$$\boxed{\mathrm{z}_v} = \mathrm{h}_v^{(L)}$$

Average of neighbor's previous layer embeddings

Total number of layers

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Q: can we use attention mechanism?

Graph networks: aggregate neighbors

Initial 0-th layer embeddings are equal to node features

$$\boxed{h_v^0 = x_v}$$

embedding of v at layer l

$$h_v^{(l+1)} = \boxed{\sigma}\left(W_l \boxed{\sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|}} + B_l \boxed{h_v^{(l)}}\right), \forall l \in \{0, \dots, \boxed{L} - 1\}$$

$$\boxed{z_v} = h_v^{(L)}$$

Average of neighbor's previous layer embeddings

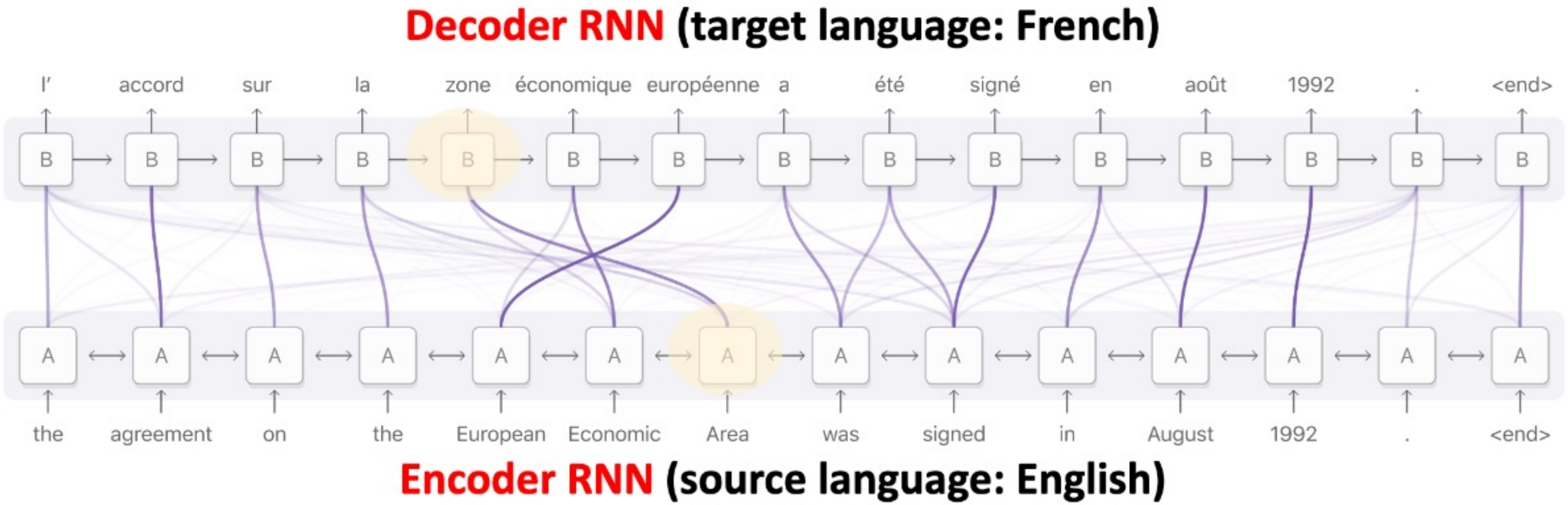
Total number of layers

Embedding after L layers of neighborhood aggregation

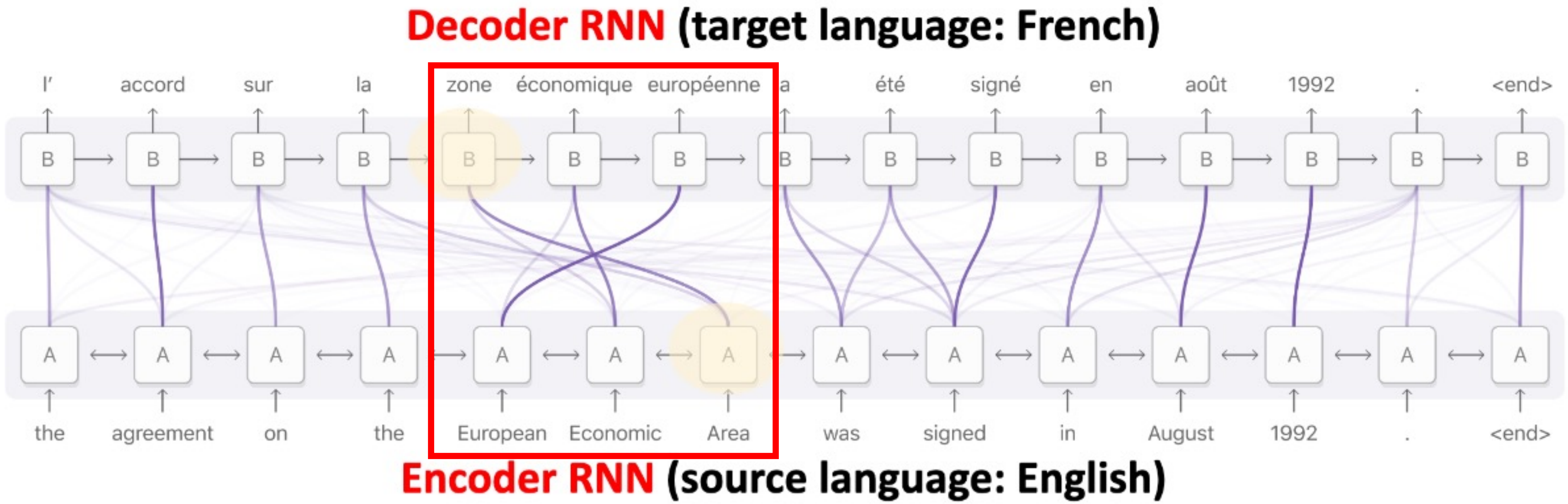
Non-linearity (e.g., ReLU)

Q: can we use attention mechanism?← Not all node's neighbors are equally important

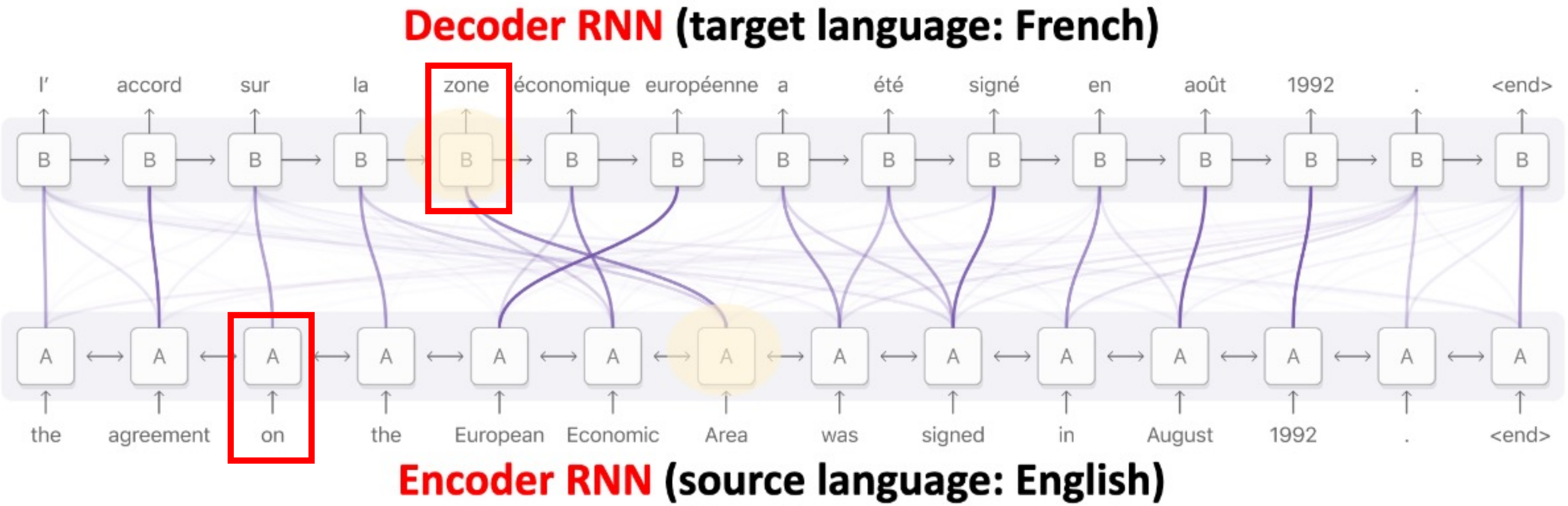
Input-output correlation



Input-output correlation



Input-output correlation



Graph attention networks

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \underbrace{\boxed{\alpha_{vu}}}_{\text{Attention weights}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Graph attention networks

Q: what stand for importance of a node?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \underbrace{\boxed{\alpha_{vu}}}_{\text{Attention weights}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Graph attention networks

Node degree?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{\boxed{u \in N(v)}} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}\right)$$

Attention weights

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\sum_{u \in N(v)} \underbrace{\boxed{\alpha_{vu}}}_{\text{Attention weights}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights

Between v and u

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights

Between v and u

$$e_{vu} = a(\mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}, \mathbf{W}^{(l)} \boldsymbol{h}_v^{(l-1)})$$

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights
Between v and u

$$e_{vu} = a(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \boldsymbol{h}_{\boxed{v}}^{(l-1)}})$$

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights

Between v and u

$$e_{vu} = a(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \boldsymbol{h}_{\boxed{v}}^{(l-1)}})$$

Make use of last layer's output

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights

Between v and u

$$e_{vu} = \boxed{a}(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \boldsymbol{h}_{\boxed{v}}^{(l-1)}})$$

Make use of last layer's output

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights
Between v and u

$$e_{vu} = \boxed{a}(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \boldsymbol{h}_{\boxed{v}}^{(l-1)}}) \Longrightarrow \alpha_{vu} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$

Make use of last layer's output

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u\in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights
Between v and u

Softmax function

$$e_{vu} = \boxed{a}(\boxed{\mathbf{W}^{(l)}\mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)}\mathbf{h}_{\boxed{v}}^{(l-1)}}) \Rightarrow \alpha_{vu} = \frac{\exp(e_{vu})}{\sum_{k\in N(v)} \exp(e_{vk})}$$

Make use of last layer's output

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights

Between v and u

Softmax function→normalized weights

$$e_{vu} = \boxed{a}(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{v}}^{(l-1)}}) \Longrightarrow \alpha_{vu} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$

Make use of last layer's output

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights

Between v and u

Softmax function→normalized weights

$$e_{vu} = \boxed{a}(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \boldsymbol{h}_{\boxed{v}}^{(l-1)}}) \Longrightarrow \boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$

Make use of last layer's output

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \alpha_{vu} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Attention weights

Between v and u

Q: what is a's structure?

Softmax function→normalized weights

$$e_{vu} = \boxed{a}\left(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{u}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \mathbf{h}_{v}^{(l-1)}}\right) \Longrightarrow \boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$

Make use of last layer's output

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Attention weights

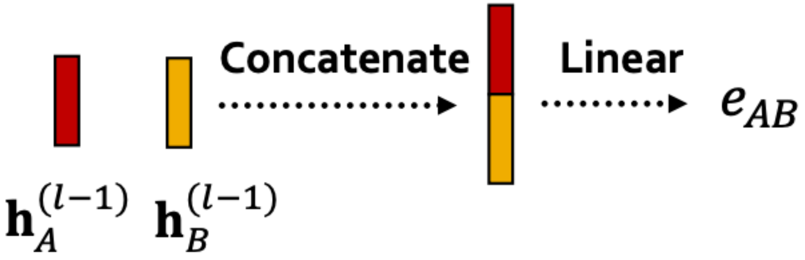
Between v and u

Q: what is a's structure?

Softmax function→normalized weights

$$e_{vu} = \boxed{a}\left(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{v}}^{(l-1)}}\right) \Rightarrow \boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$

Make use of last layer's output



http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Attention weights

Between v and u

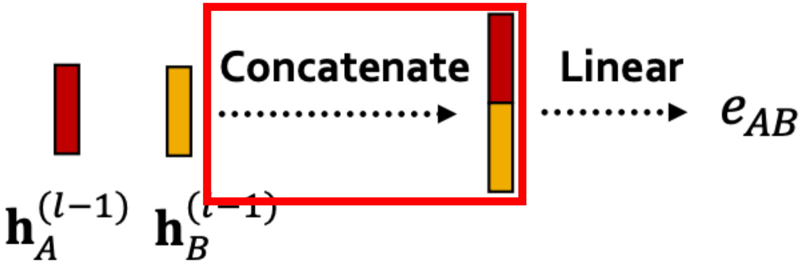
Q: what is a's structure?

$$e_{vu} = \boxed{a}\left(\boxed{\mathbf{W}^{(l)}\mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)}\mathbf{h}_{\boxed{v}}^{(l-1)}}\right)$$

Make use of last layer's output

Softmax function→normalized weights

$$\boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$



http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma(\textstyle\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)})$$

Attention weights

Between v and u

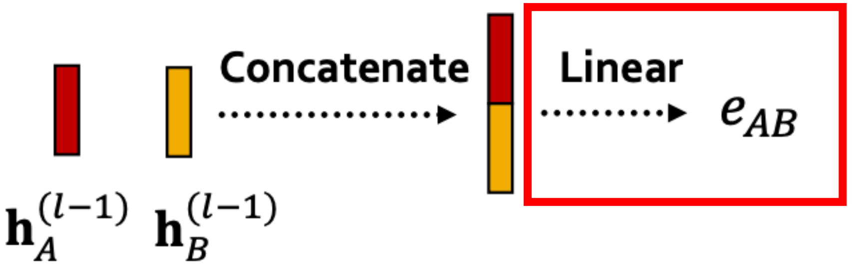
Q: what is a's structure?

$$e_{vu} = \boxed{a}(\boxed{\mathbf{W}^{(l)}\mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)}\mathbf{h}_{\boxed{v}}^{(l-1)}})$$

Make use of last layer's output

Softmax function→normalized weights

$$\boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$



http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Attention weights
Between v and u

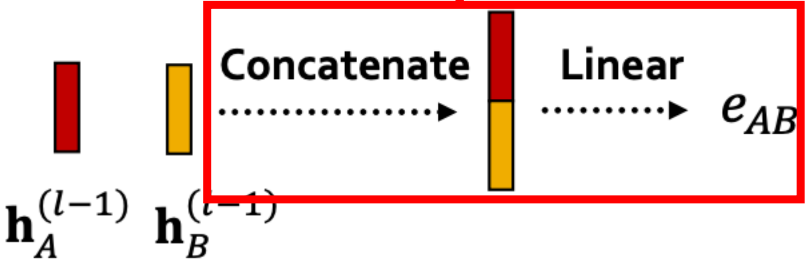
Q: what is a's structure?

$$e_{vu} = \boxed{a}\left(\boxed{\mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \mathbf{h}_v^{(l-1)}}\right)$$

Make use of last layer's output

Softmax function→normalized weights

$$\boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$



http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Attention weights

Between v and u

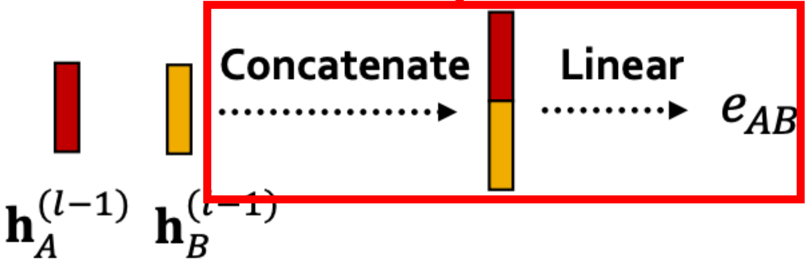
Softmax function→normalized weights

Q: what is a's structure?

$$e_{vu} = \boxed{a}\left(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{v}}^{(l-1)}}\right)$$

Make use of last layer's output

$$\boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$



$$e_{AB} = a\left(\mathbf{W}^{(l)} \mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)} \mathbf{h}_B^{(l-1)}\right)$$
$$= \text{Linear}\left(\text{Concat}\left(\mathbf{W}^{(l)} \mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)} \mathbf{h}_B^{(l-1)}\right)\right)$$

http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Attention weights

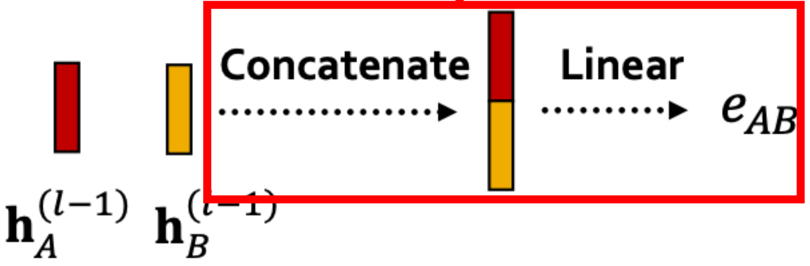
Between v and u

Q: what is a's structure?

Softmax function→normalized weights

$$e_{vu} = \boxed{a}\left(\boxed{\mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \mathbf{h}_v^{(l-1)}}\right) \Rightarrow \boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$

Make use of last layer's output



$$e_{AB} = a\left(\mathbf{W}^{(l)}\mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)}\mathbf{h}_B^{(l-1)}\right)$$
$$= \text{Linear}\left(\text{Concat}\left(\boxed{\mathbf{W}^{(l)}\mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)}\mathbf{h}_B^{(l-1)}}\right)\right)$$

http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u\in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)}\mathbf{h}_u^{(l-1)}\right)$$

Attention weights

Between v and u

Q: what is a's structure?

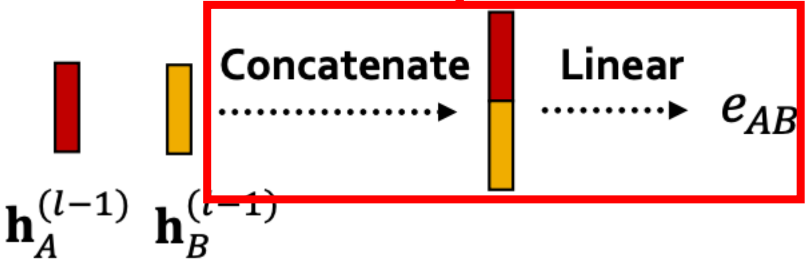
$$e_{vu} = \boxed{a}\left(\boxed{\mathbf{W}^{(l)}\mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)}\mathbf{h}_{\boxed{v}}^{(l-1)}}\right)$$

Make use of last layer's output

$\Longrightarrow$

Softmax function→normalized weights

$$\boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k\in N(v)} \exp(e_{vk})}$$



$\Longrightarrow$

$$e_{AB} = a\left(\mathbf{W}^{(l)}\mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)}\mathbf{h}_B^{(l-1)}\right)$$
$$= \text{Linear}\left(\boxed{\text{Concat}\left(\mathbf{W}^{(l)}\mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)}\mathbf{h}_B^{(l-1)}\right)}\right)$$

http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Graph attention networks

Learnable parameters?

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \boxed{\alpha_{vu}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Attention weights
Between v and u

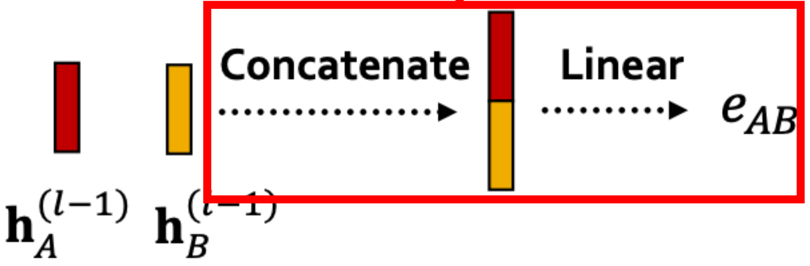
Softmax function→normalized weights

Q: what is a's structure?

$$e_{vu} = \boxed{a}\left(\boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{u}}^{(l-1)}}, \boxed{\mathbf{W}^{(l)} \mathbf{h}_{\boxed{v}}^{(l-1)}}\right)$$

Make use of last layer's output

$$\boxed{\alpha_{vu}} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$



$$e_{AB} = a\left(\mathbf{W}^{(l)} \mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)} \mathbf{h}_B^{(l-1)}\right)$$
$$= \boxed{\text{Linear}\left(\text{Concat}\left(\mathbf{W}^{(l)} \mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)} \mathbf{h}_B^{(l-1)}\right)\right)}$$

http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf