392133 Recent Advances in Deep Learning (S) (SoSe 2020)

Graph Neural Networks (GNN)

Presenter:

Muhammad Raheel **(3903141)**

Content:

- Distributed Vector Representation
- Graph Notation
- Introduction
- Graph Neural Network
- How GNN works
- Neural message passing
- Graph Neural Network Architectures
- Gated GNNs
- GCNs
- Expressing GGNNs as Matrix Operations
- Use of GNNs
- References
- Questions & Answers Session

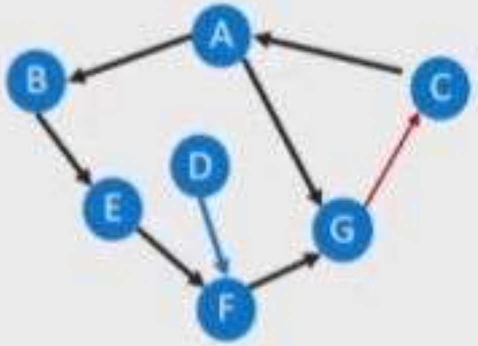
Distributed Vector Representations

$\boldsymbol{r} = E\ \mathbb{I}_w$ with E a DxV matrix

"vocabulary"

Graph Notation

- Nodes/Vertices
- Edges

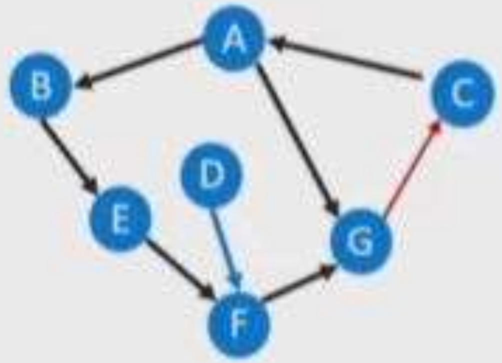


$$G = (V, E)$$

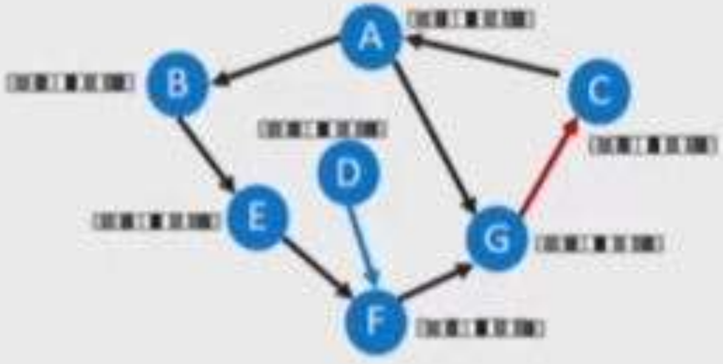
Graph Neural Networks

and Neural Message Passing

Graph Neural Networks

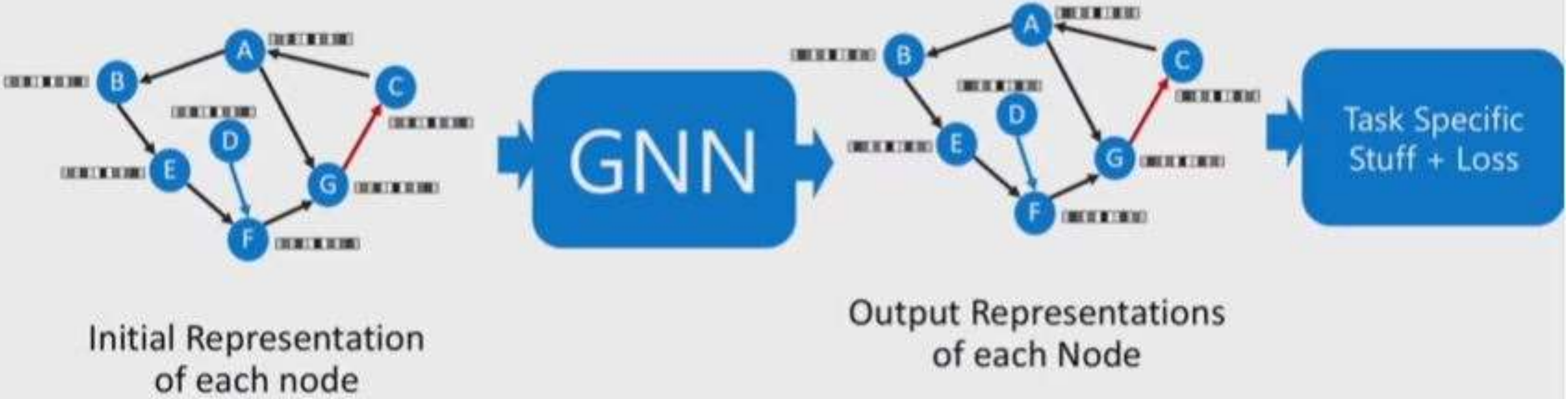


Graph Representation
of Problem

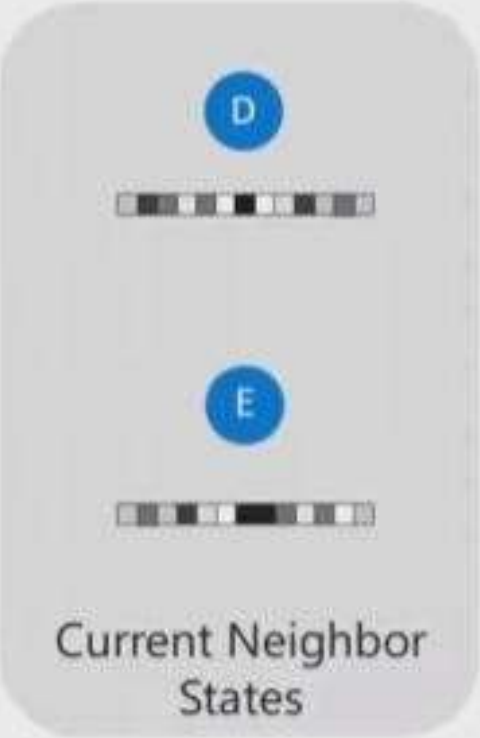


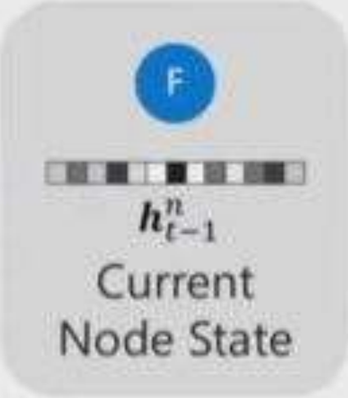
Initial Representation
of each node

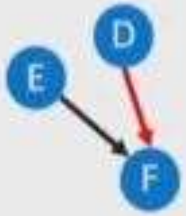
Graph Neural Networks



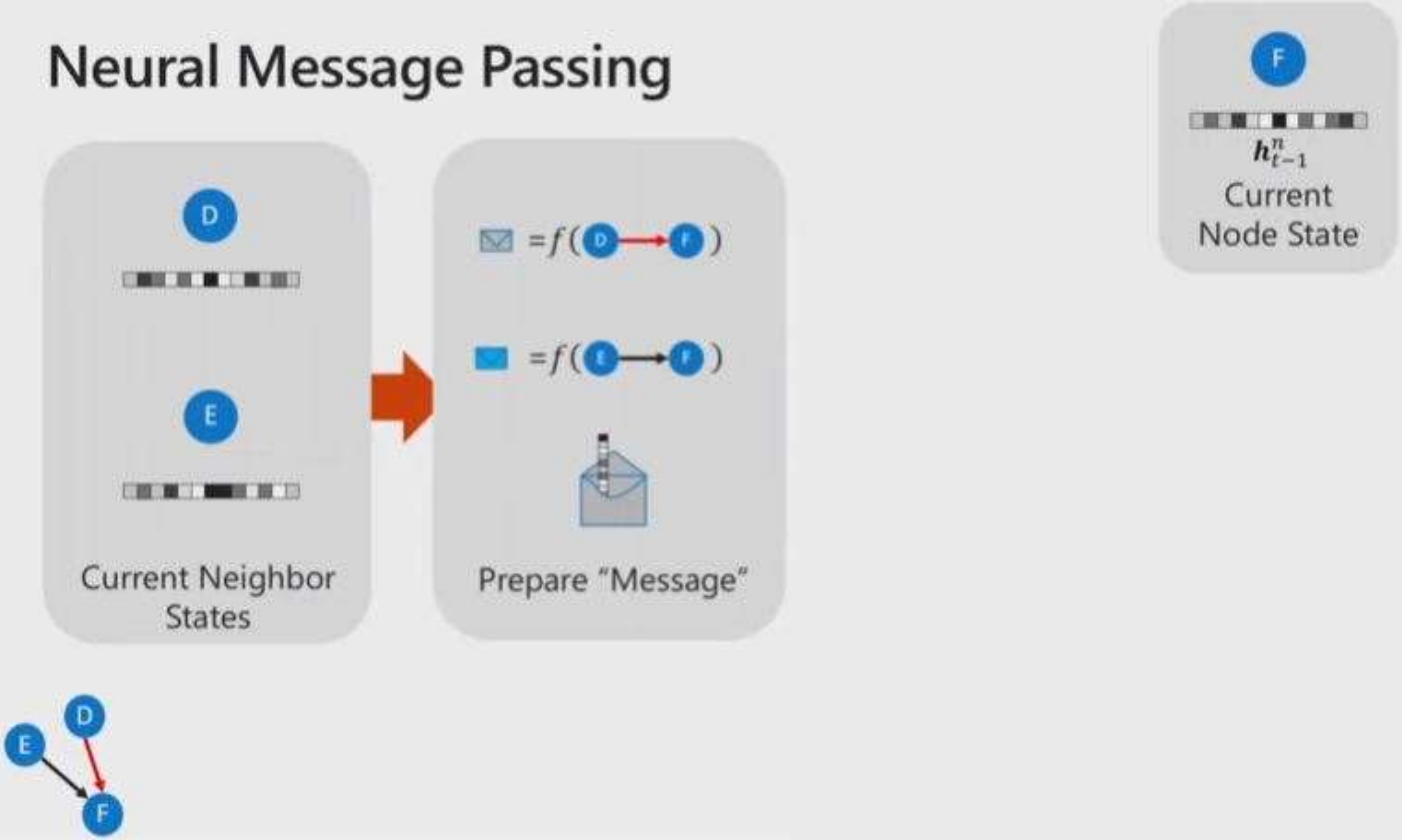
Neural Message Passing

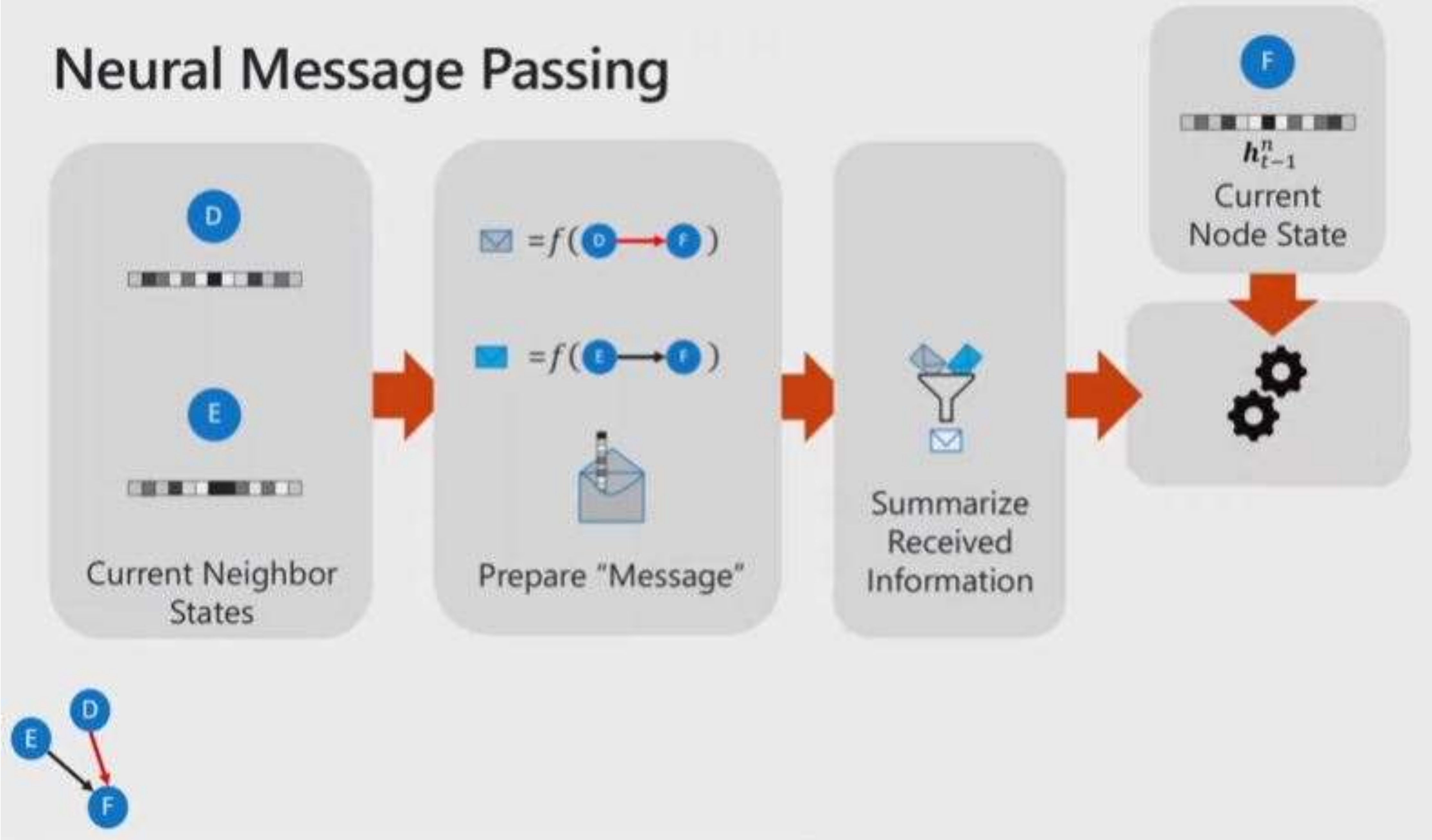


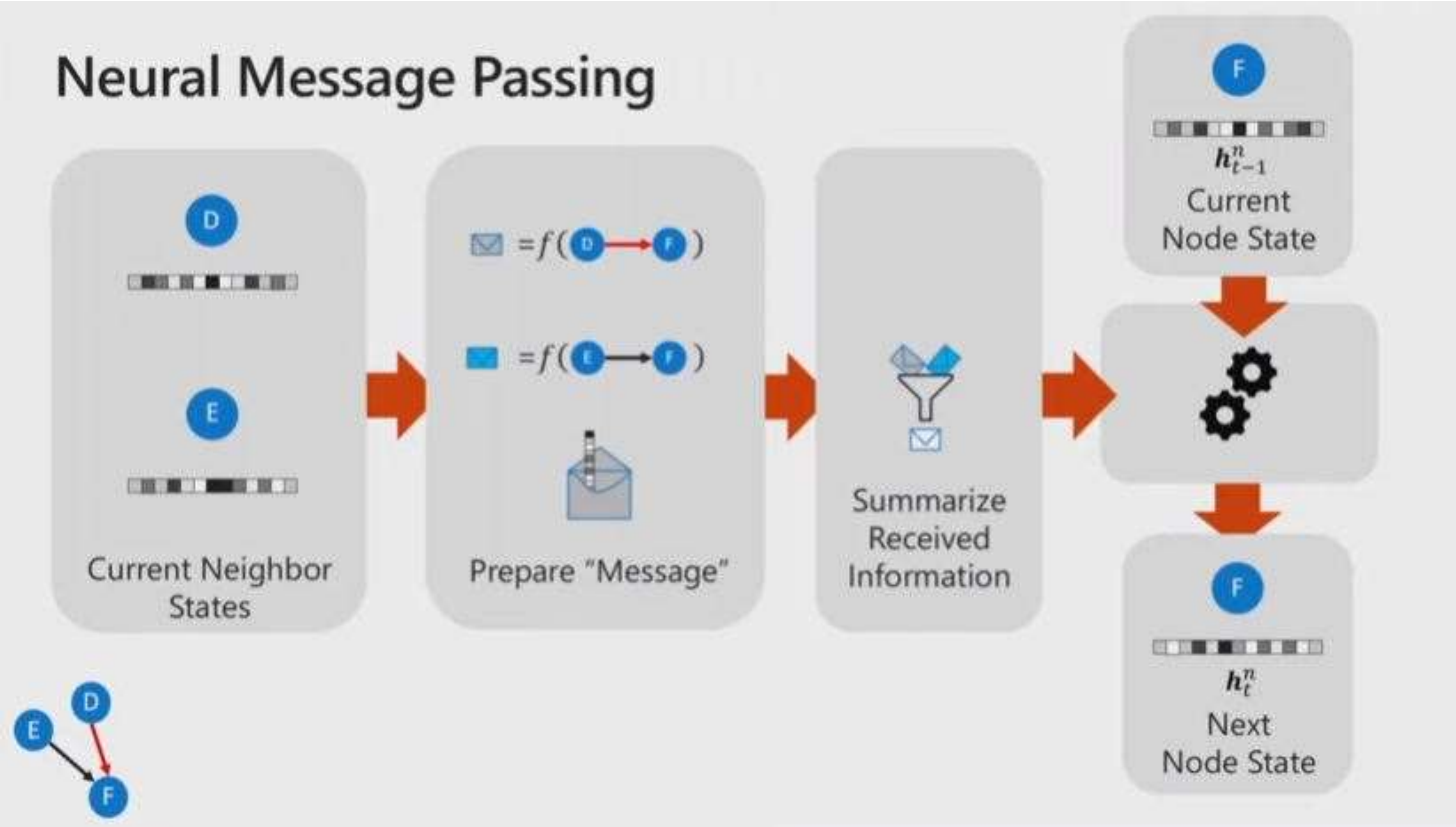




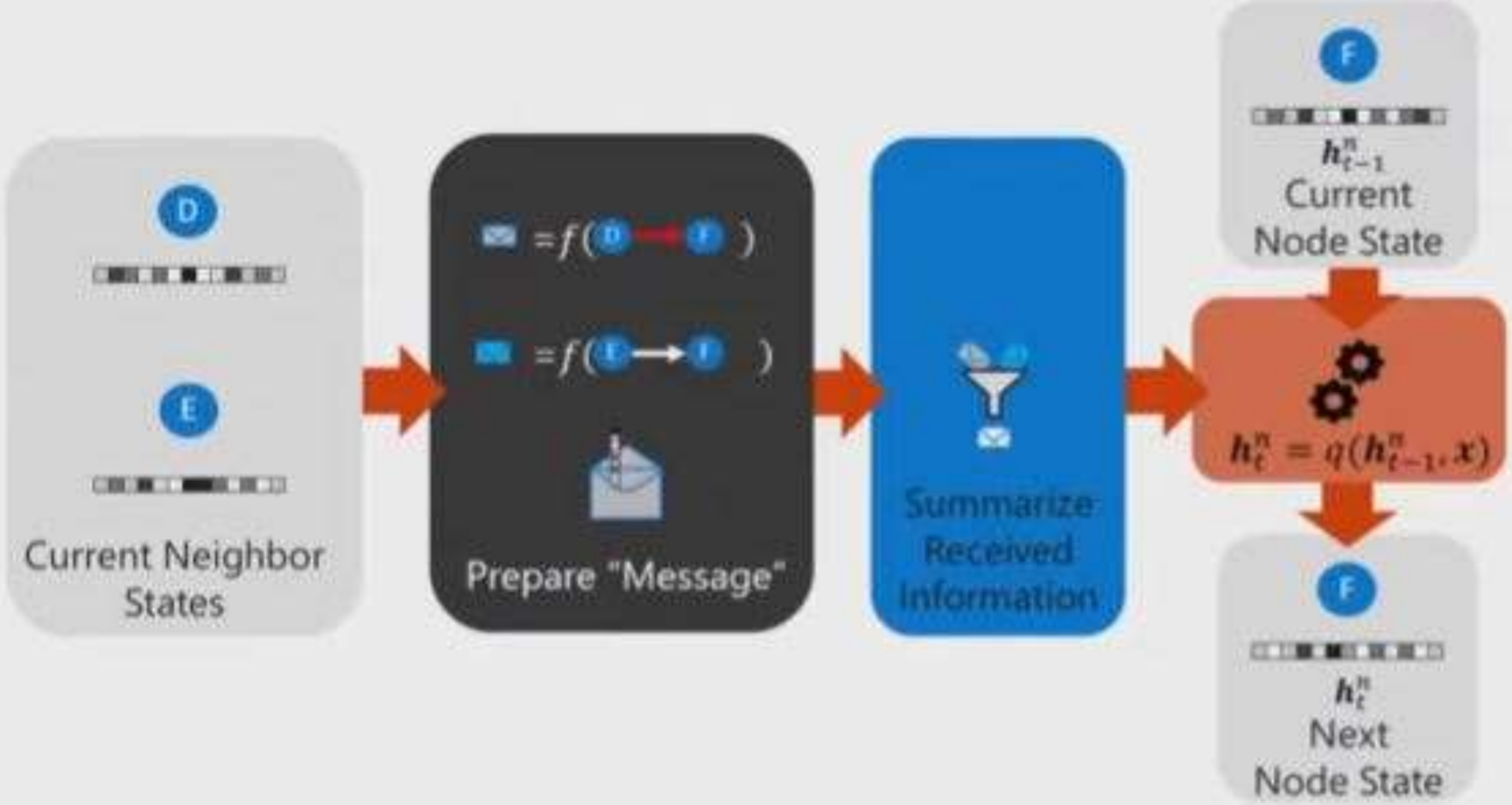
Neural Message Passing



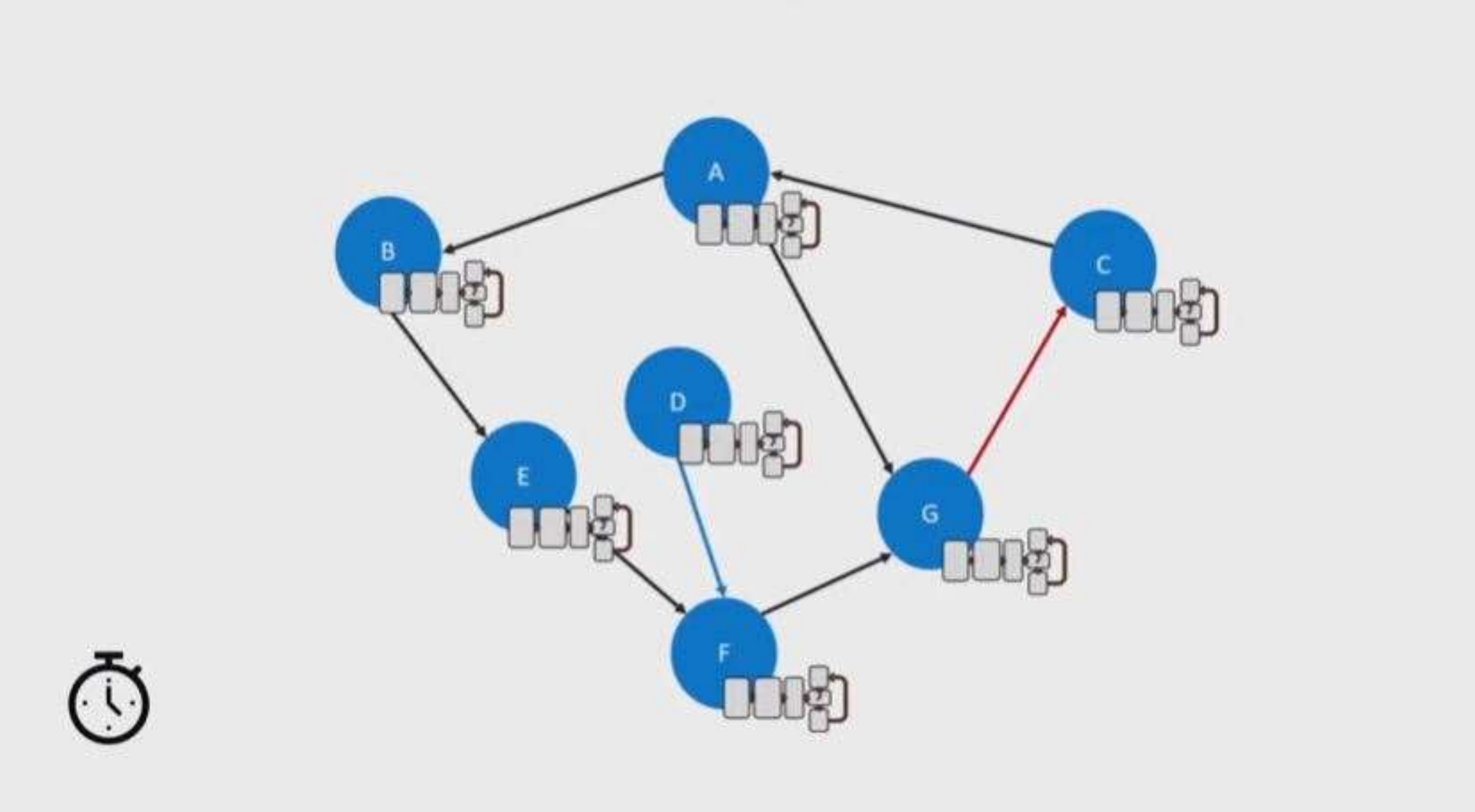

Neural Message Passing
D
E
Current Neighbor States
$= f(D \rightarrow F)$
$= f(E \rightarrow F)$
Prepare "Message"
Summarize Received Information
F
h_{t-1}^{n}
Current Node State
D
E
F



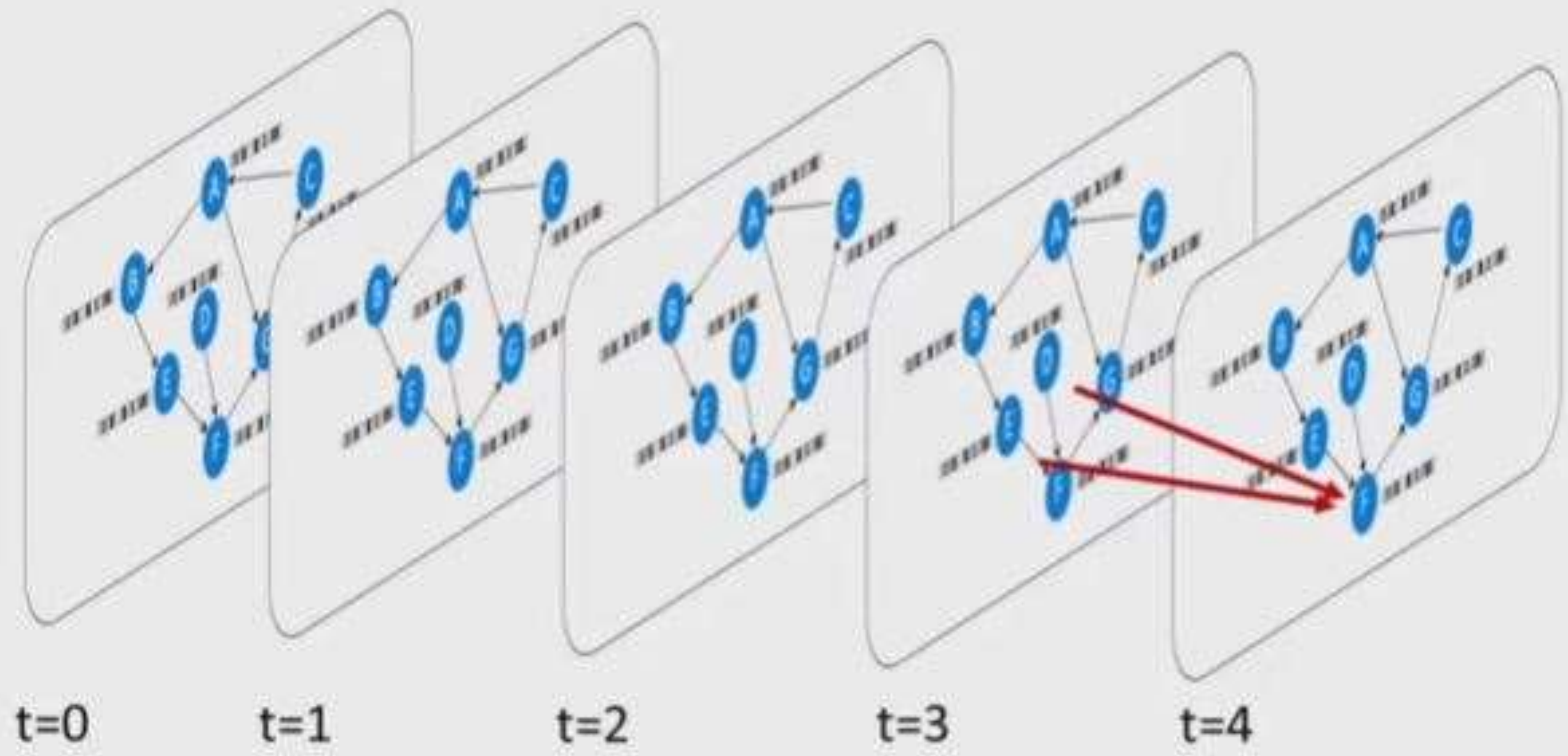
Neural Message Passing
D
E
Current Neighbor States
$= f($D → F$)$
$= f($E → F$)$
Prepare "Message"
Summarize Received Information
F
h_{t-1}^{n}
Current Node State
F
h_{t}^{n}
Next Node State
D
E
F

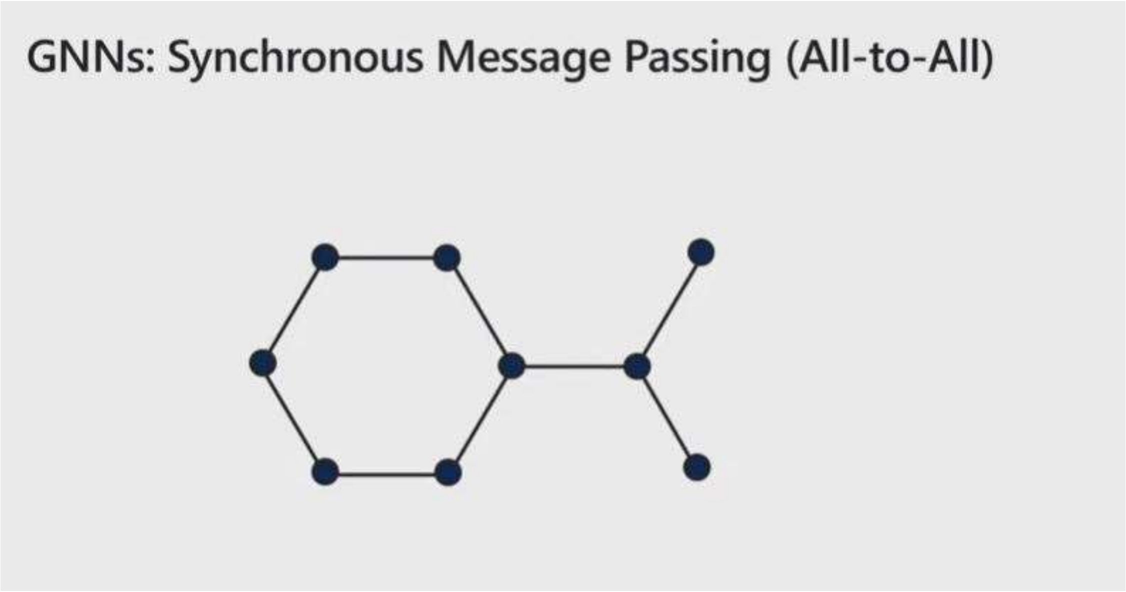


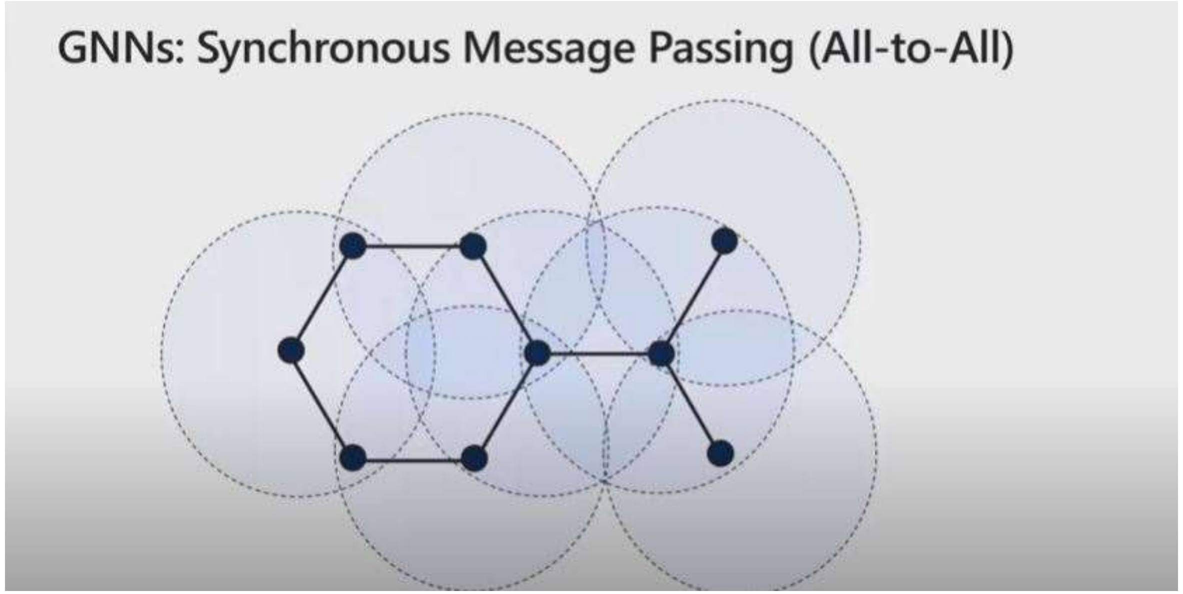
$$\boldsymbol{h}_t^n = q\left(\boldsymbol{h}_{t-1}^n, \bigcup_{\forall n_j : n \xrightarrow{k} n_j} f_t\left(\boldsymbol{h}_{t-1}^n, k, \boldsymbol{h}_{t-1}^{n_j}\right)\right)$$


A
B
C
D
E
F
G

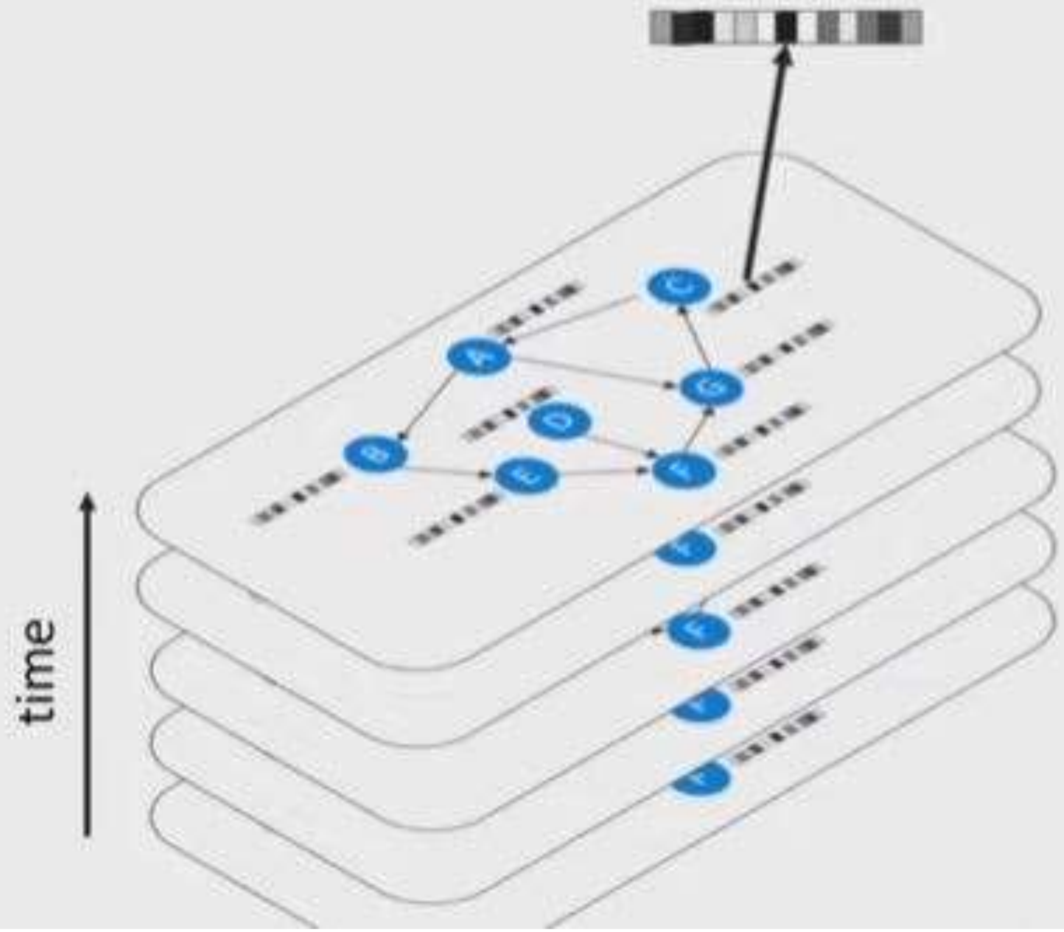
Graph Neural Networks: Message Passing



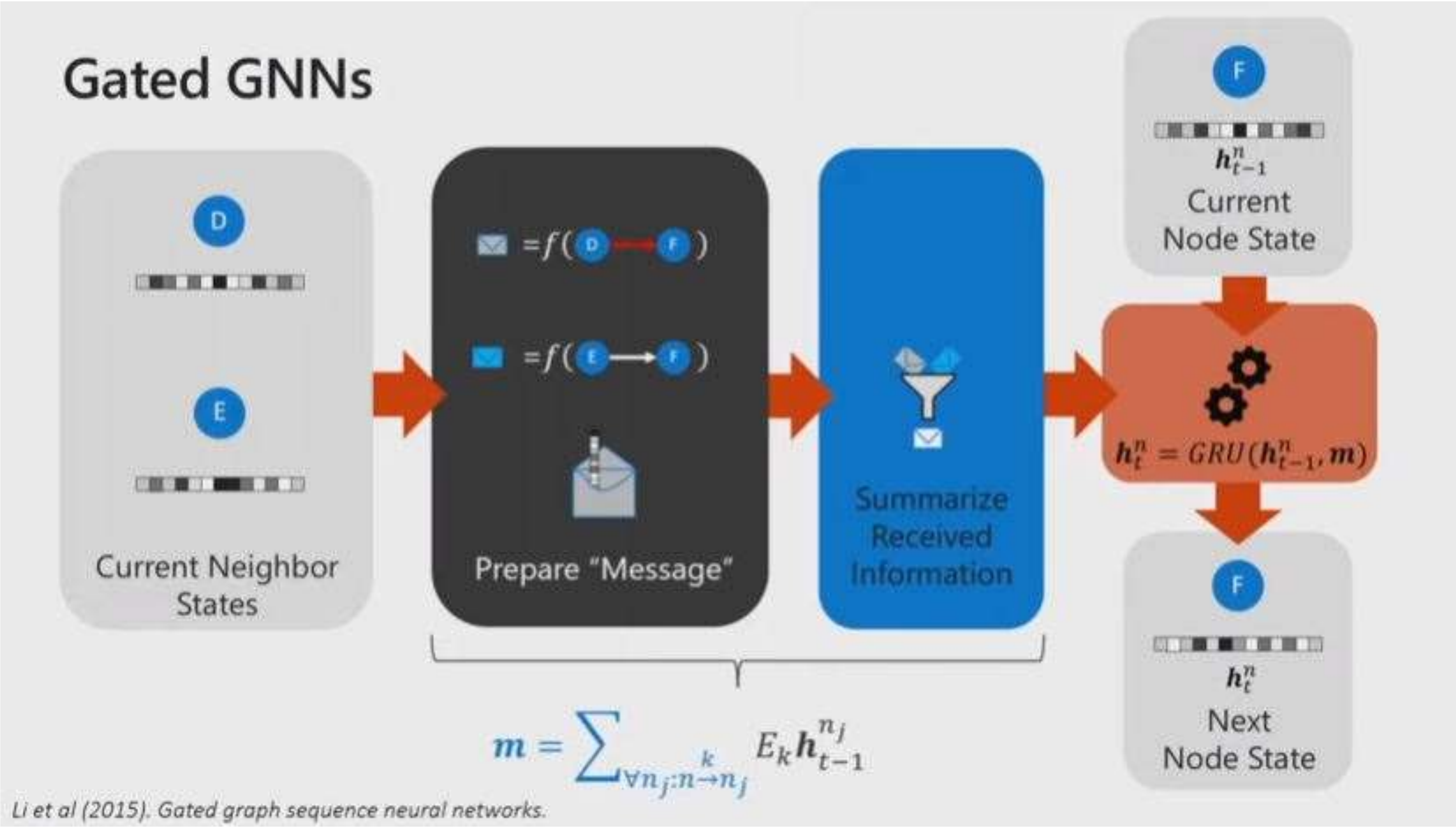

GNNs: Synchronous Message Passing (All-to-All)

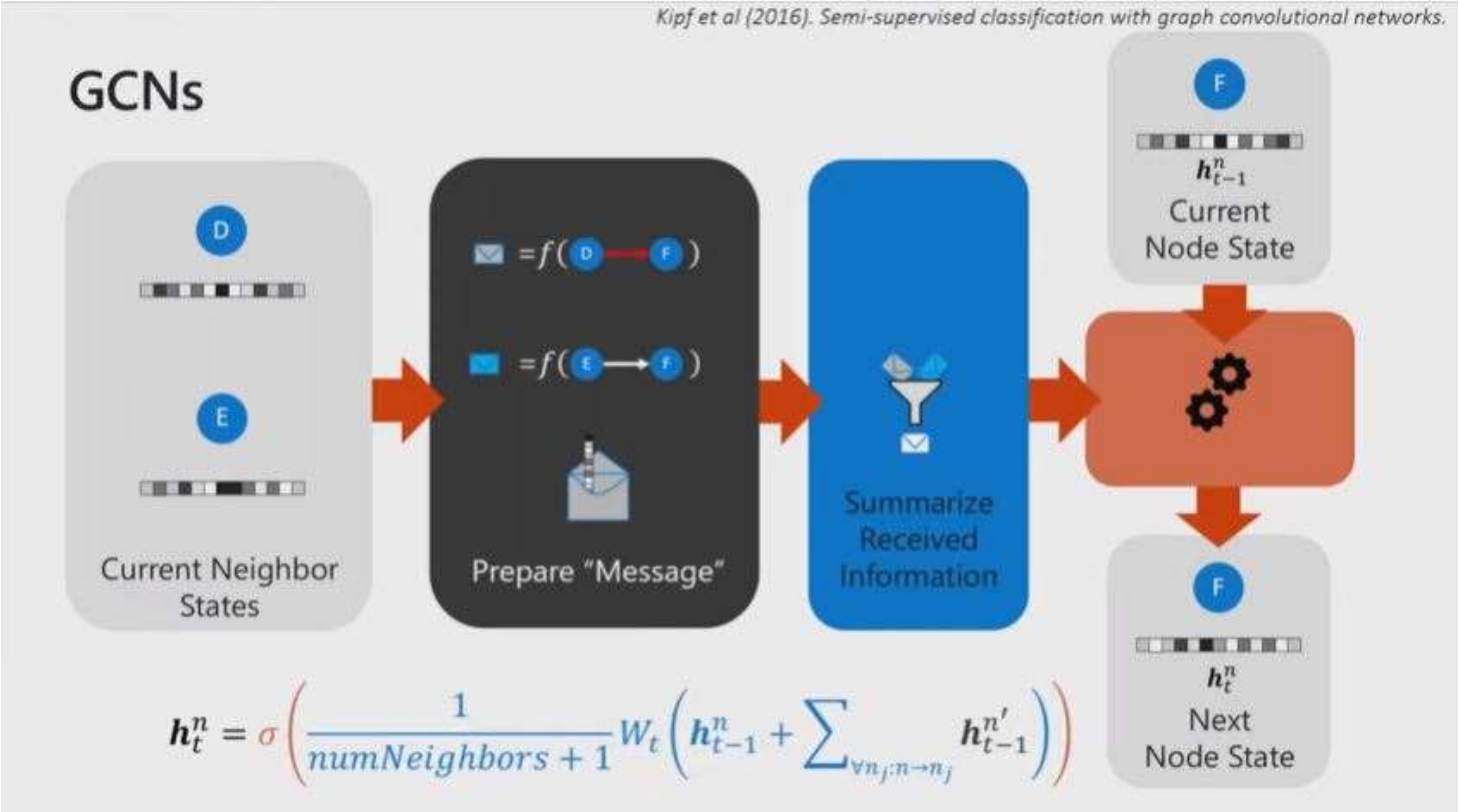

GNNs: Synchronous Message Passing (All-to-All)

Graph Neural Networks: Output



- node selection
- node classification
- graph classification

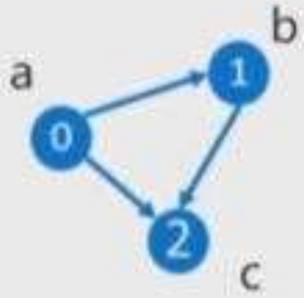

Gated GNNs
D
E
Current Neighbor States
$=f(D \to F)$
$=f(E \to F)$
Prepare "Message"
Summarize Received Information
F
h_{t-1}^{n}
Current Node State
$h_t^n = GRU(h_{t-1}^n, m)$
F
h_t^n
Next Node State
$m = \sum_{\forall n_j : n \xrightarrow{k} n_j} E_k h_{t-1}^{n_j}$
Li et al (2015). Gated graph sequence neural networks.


Kipf et al (2016). Semi-supervised classification with graph convolutional networks.
GCNs
D
E
Current Neighbor States
=f(D F)
=f(E F)
Prepare "Message"
Summarize Received Information
F
h_{t-1}^{n}
Current Node State
F
h_t^n
Next Node State
$h_t^n = \sigma\left(\frac{1}{numNeighbors+1} W_t \left(h_{t-1}^{n} + \sum_{\forall n_j : n \to n_j} h_{t-1}^{n'}\right)\right)$

Expressing GGNNs as Matrix Operations

Graph Notation (2) — Adjacency Matrix

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \ \mathrm{N} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

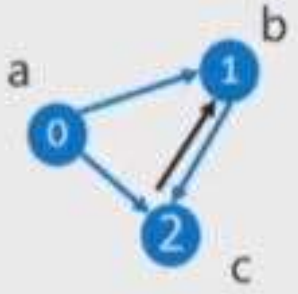


$$A \cdot N = \begin{bmatrix} 0 \\ a \\ a + b \end{bmatrix}$$

Graph Notation (2) — Adjacency Matrix

$$A_0 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



GGNN as Matrix Operation

Node States

$$H_t = \begin{bmatrix} \boldsymbol{h}_t^{n_0} \\ \vdots \\ \boldsymbol{h}_t^{n_K} \end{bmatrix} \text{ (num\_nodes x D)}$$

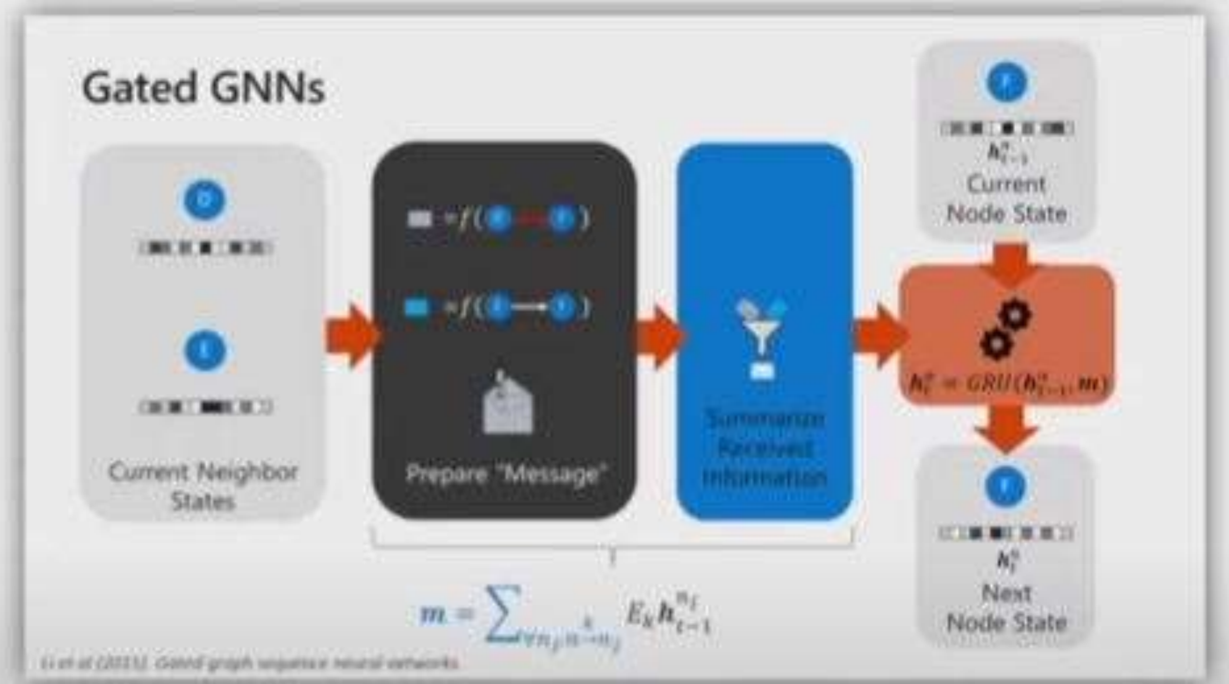
Messages to-be sent

$$M_t^k = E_k H_t \text{ (num\_nodes x M)}$$

Received Messages

$$R_t = \sum_k A M_t^k \text{ (num\_nodes x M)}$$

Update $H_{t+1} = GRU(H_t, R_t)$



Where to use GNN

- Financial Markets
- Search Engines
- Social Networks
- Chemistry
- Knowledge

Graph neural networks are already being used in **image and speech recognition**. Unstructured, natural information can potentially be processed more effectively with a GNN than with traditional neural networks.

Reference

- MSR Cambridge, AI Residency Advanced Lecture Series
- Semi supervised classification with graph convolutional networks kipf et al (2016)
- Gated graph sequence neural networks Li et al (2015)

QUESTION & ANSWER SESSION