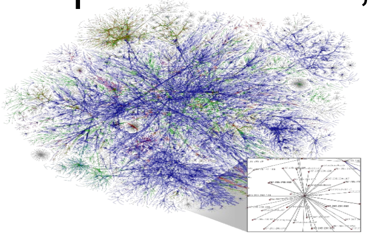
Graph Representation Learning and Generation

Jian Tang
Assistant Professor
Montréal Institute for Learning Algorithms (MILA)
HEC Montréal
tangjianpku@gmail.com

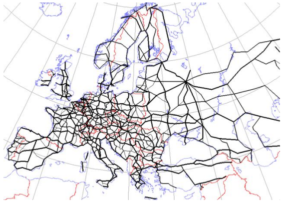
Graphs: general and flexible data structures

- Ubiquitous in real-world, arises in multiple disciplines
  - computer science, social science, healthcare, bioinformatics, ...

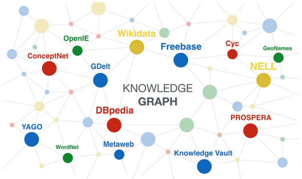


World Wide Web

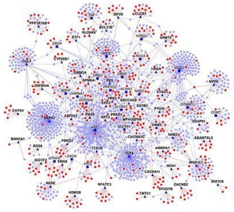
Social Network



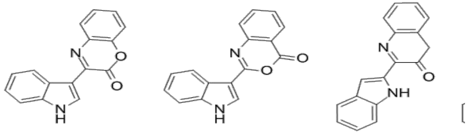
Road Network



Knowledge Graph



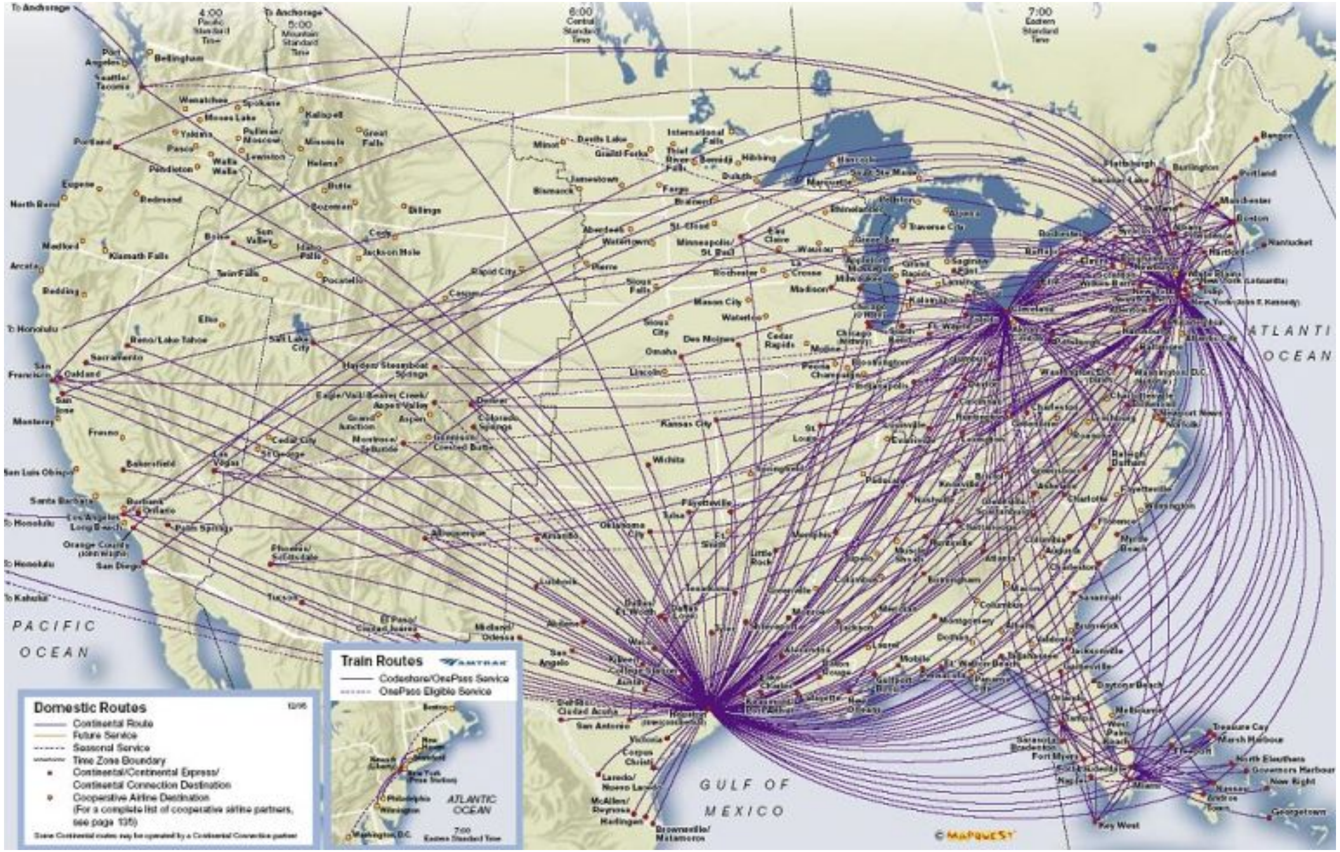
Protein-protein Interaction Graph



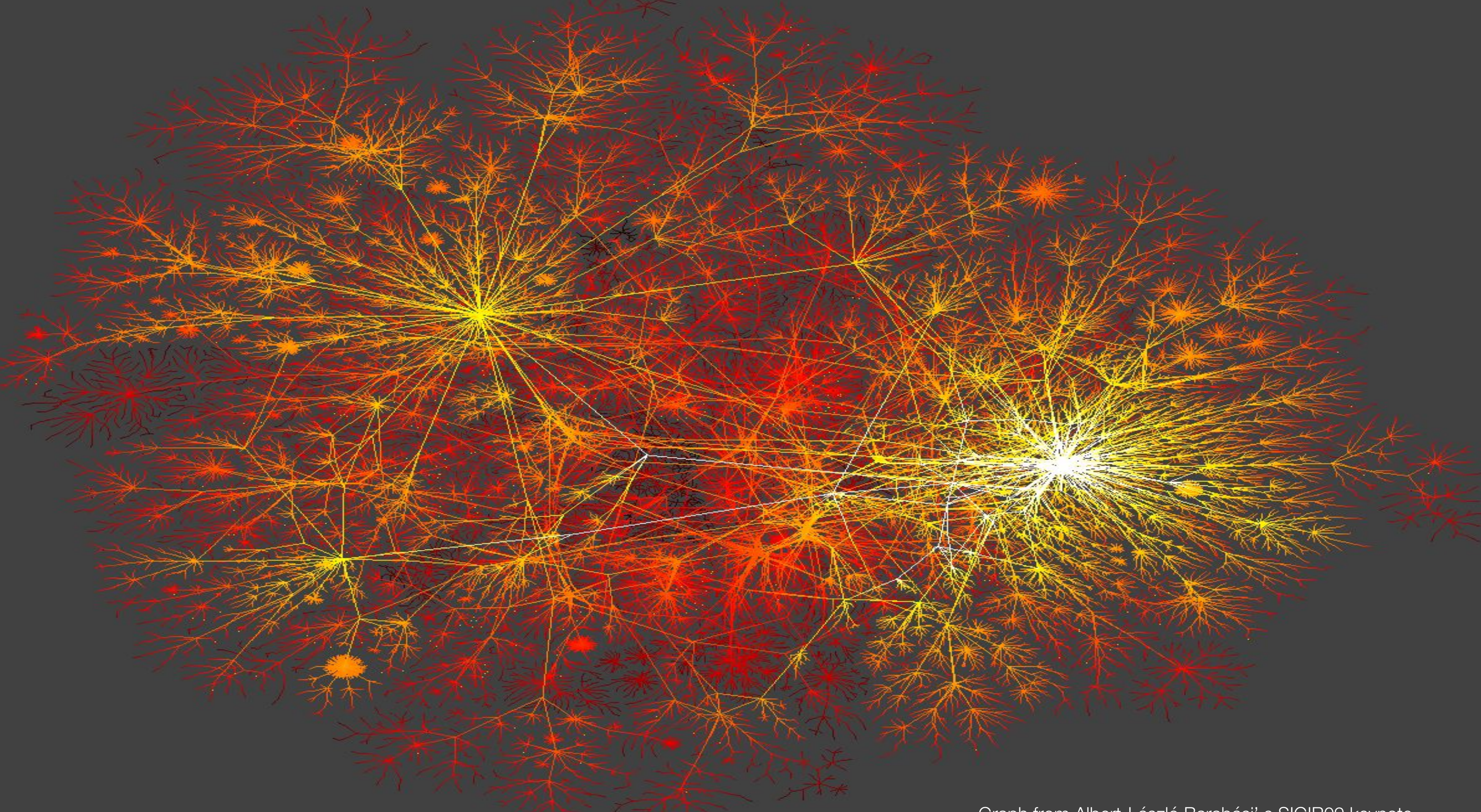
1 2 3

Molecular structures

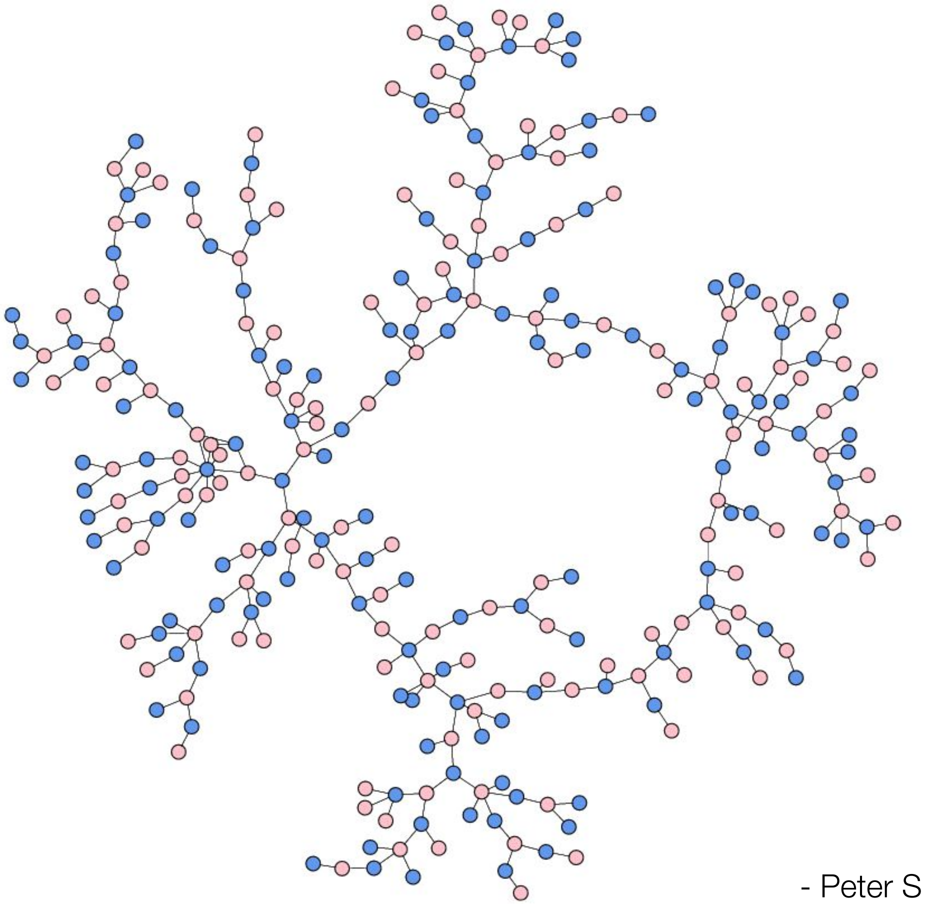
- Many data can be formulated as graphs
  - Images as graphs with two-dimensional grid structures



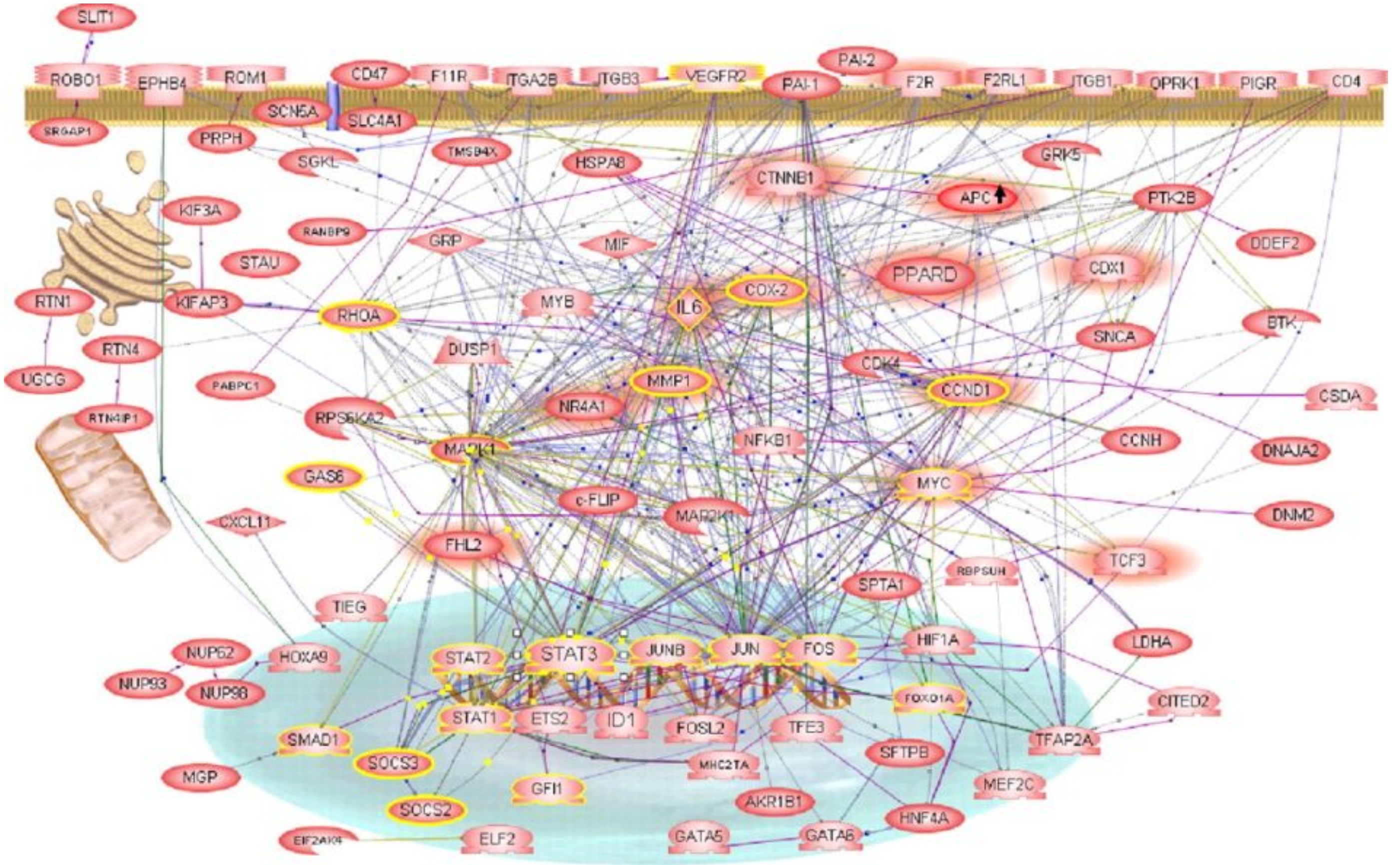
Continental Airline: Source: http://www.airlineroutemaps.com/



Graph from Albert-László Barabási’ s SIGIR09 keynote



- Peter S. Bearman, James Moody and Katherine Stovel
Chains of affection: The structure of adolescent romantic and sexual networks , *American Journal of Sociology* 110 44-91 (2004)



- Abdollahi et al. Transcriptional network governing the angiogenic switch in human pancreatic cancer. PNAS vol. 104 no. 31,2007

Gene-Regulatory Network

Various Applications on Graphs

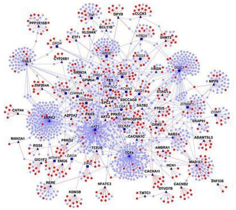
- Applications
  - Predicting political bias of people in social networks
  - Recommending friends in social networks
  - Recommending coauthors/papers in coauthor/author-paper graphs
  - Predict the roles of proteins in a protein-protein interaction graphs
  - Predict the properties of molecules
  - Generating new molecules for drug discovery
  - …

Most applications are essentially making predictions on graphs!

Research Problem (1) : Learning Node Representations of Graphs

- Learning **node representations** for large-scale graphs
  - e.g., social graphs, protein-protein interaction graphs

Social Network

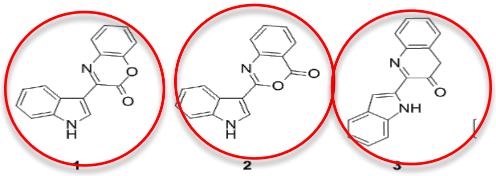


Protein-protein Interaction Graph

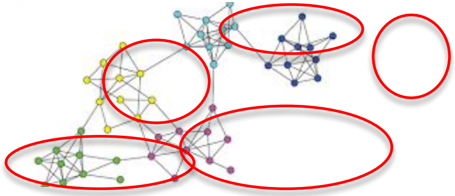
- Challenges:
  - Large-scale: >millions of nodes and billions of edges
  - Heterogeneous: multiple types of nodes and edges
  - Dynamic: structures evolve over time

Research Problem (2): Learning Representations of Entire Graphs

- Learning **representations** for entire graphs (subgraphs)
  - i.e., representing entire graphs (subgraphs) with a vector
  - E.g., molecular structures, community structures in social networks



Molecular structures

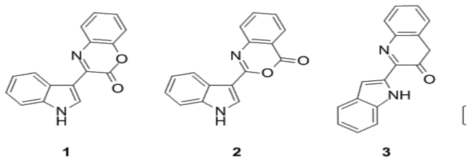


Communities in social networks

- Challenges:
  - The structures of different graphs are different
  - Existing models (e.g., CNN and RNN) are mainly developed for images and sequences.

Research Problem (3): Graph Generation

- How to generate graphs
    - E.g., for drug discovery



- Challenges:
    - Discrete
    - Arbitrary graph structure

Neural Message Passing Algorithms

...

Shagun Sodhani

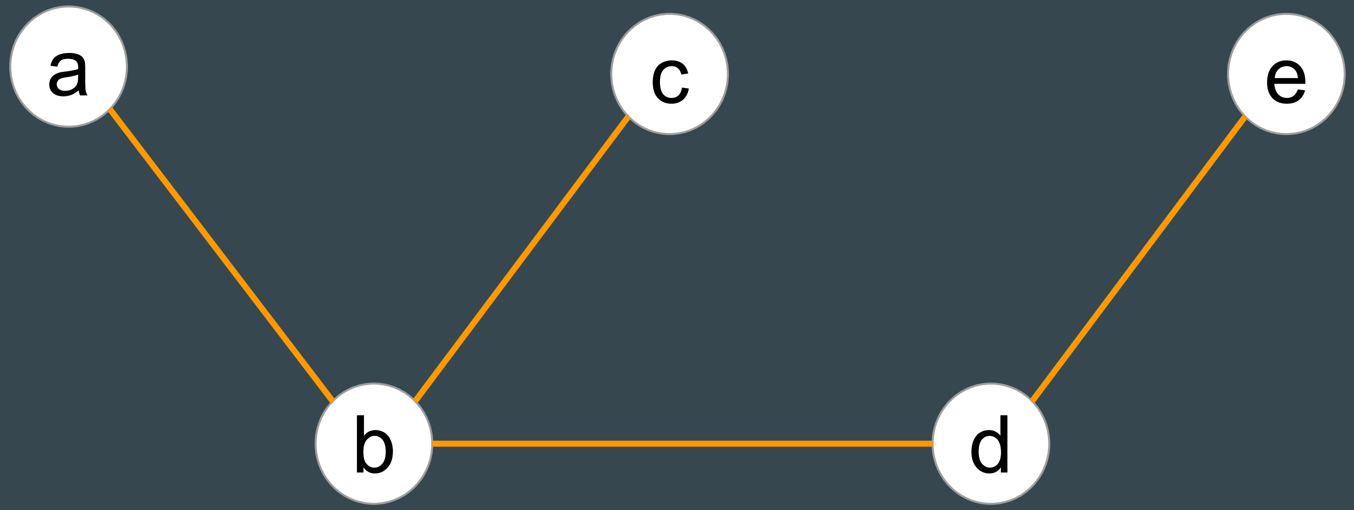
Preliminaries

- G - Undirected Graph
- x_v - Node Features
- $e_{v,w}$ - Edge Features
- h^t_v - Node Representation at time t

Message Passing Neural Network

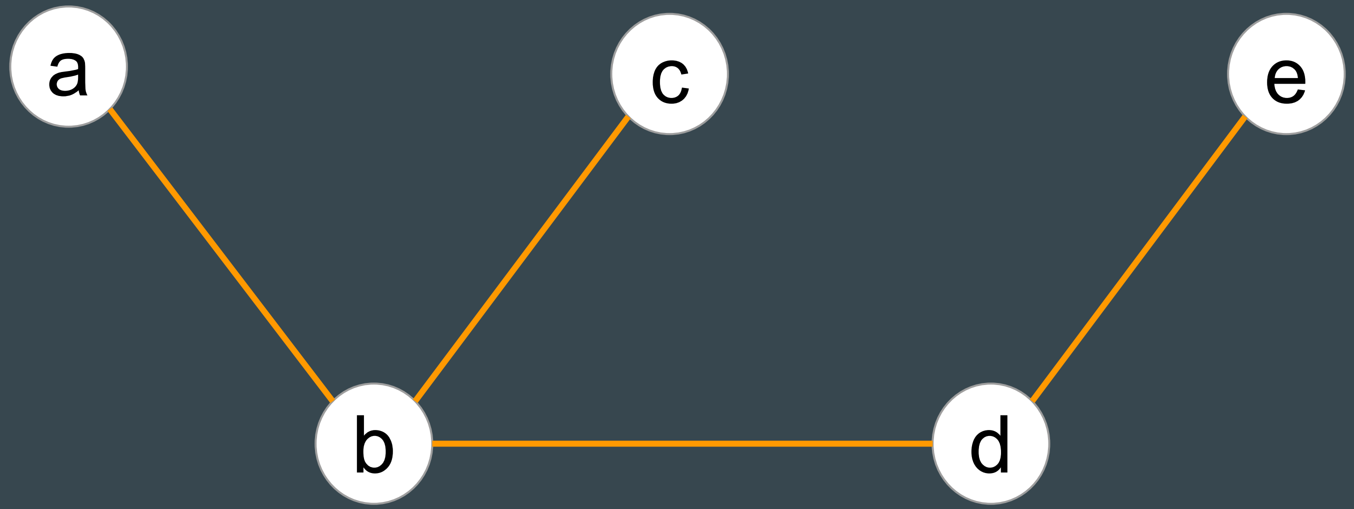
Message Passing Neural Network

Given an undirected graph



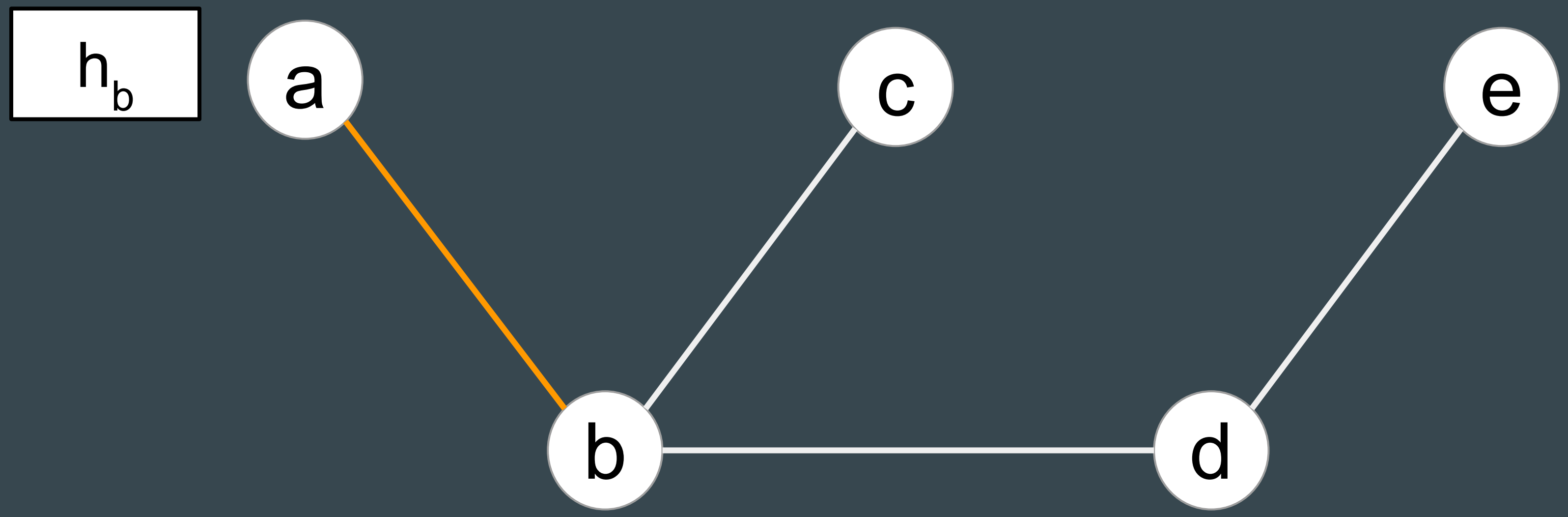
Message Passing Neural Network

Every node receives a message from its neighbour



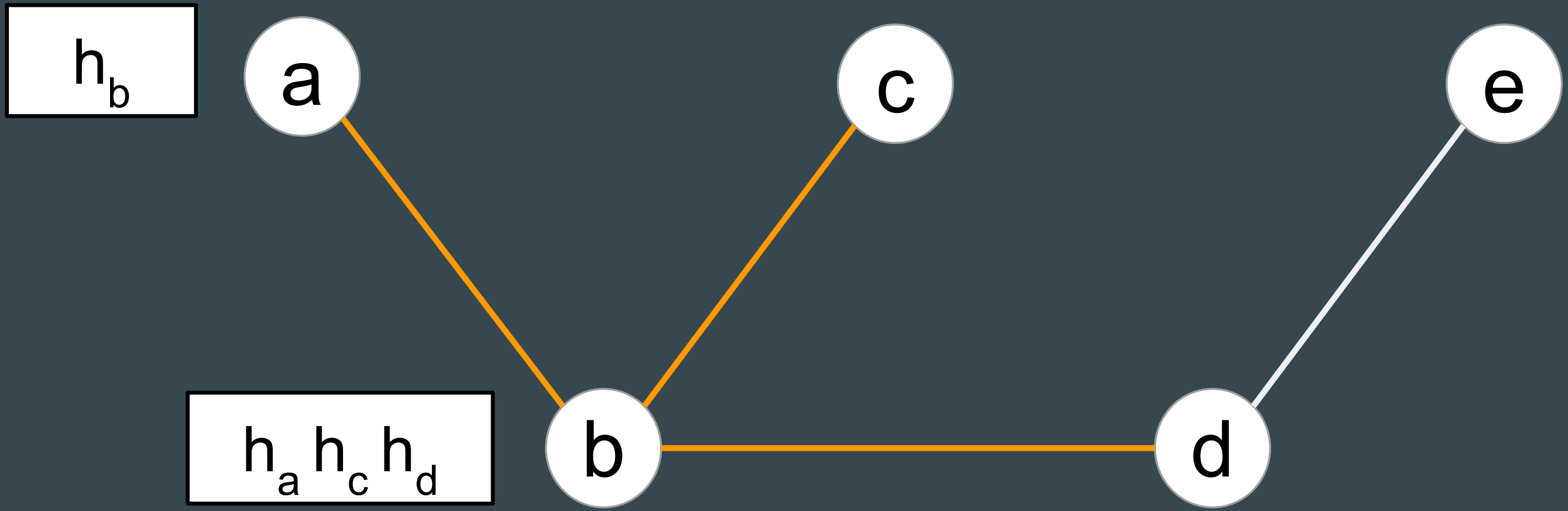
Message Passing Neural Network

Every node receives a message from its neighbour



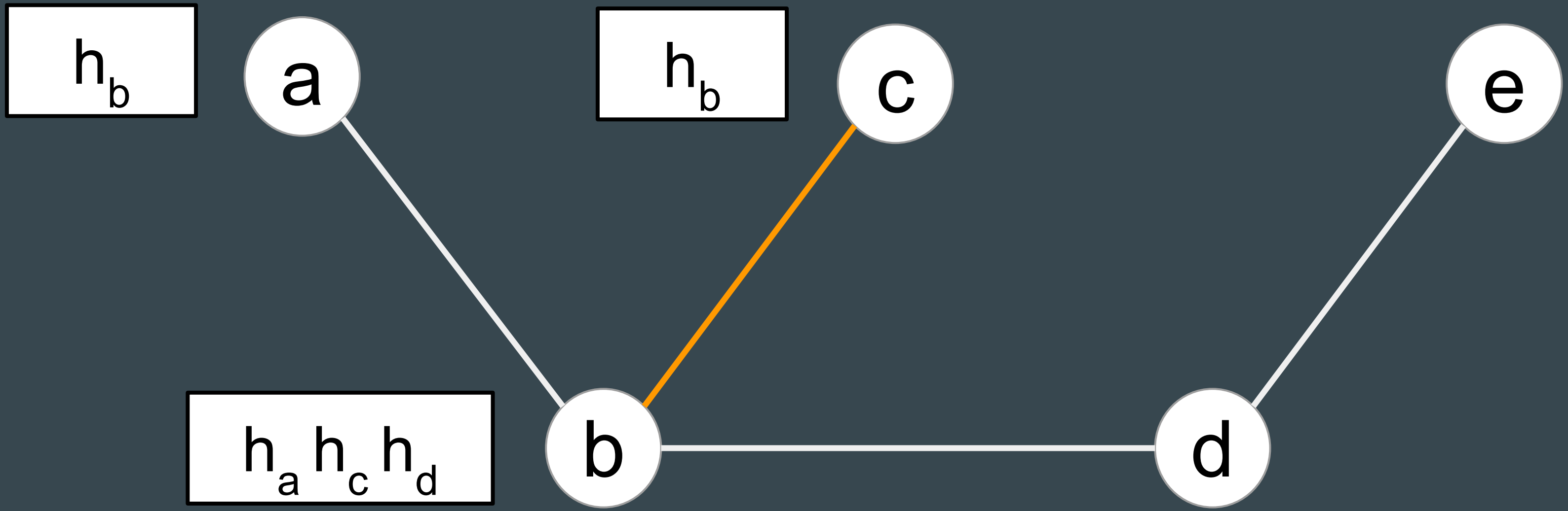
Message Passing Neural Network

Every node receives a message from its neighbour



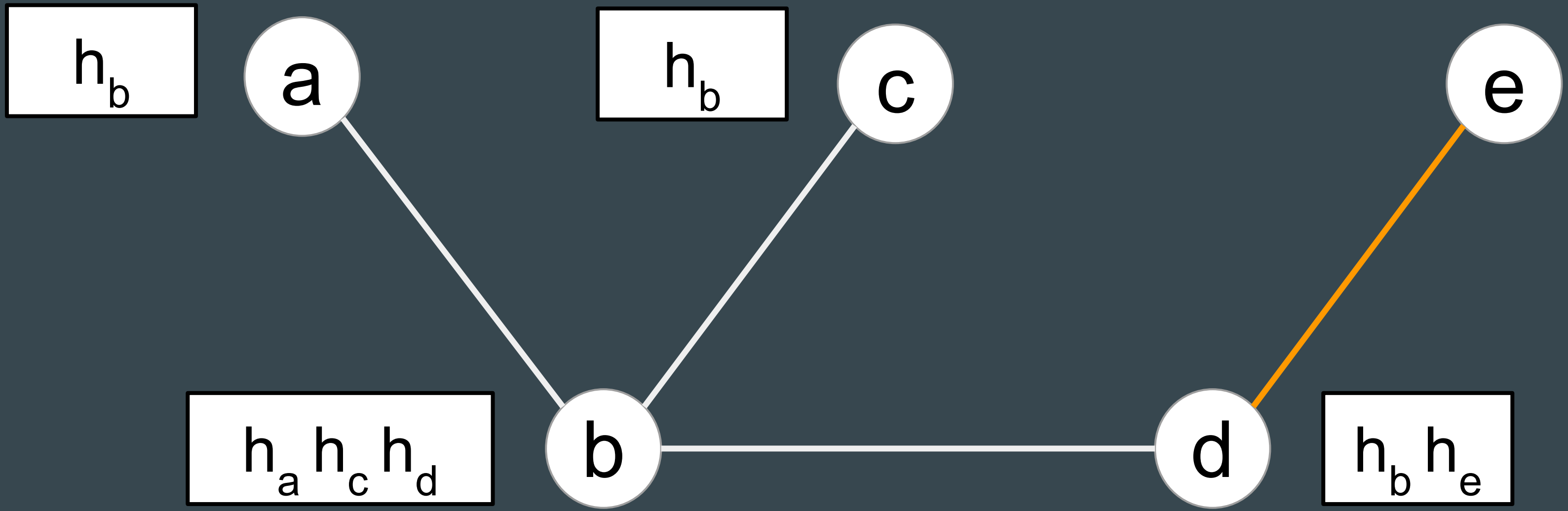
Message Passing Neural Network

Every node receives a message from its neighbour



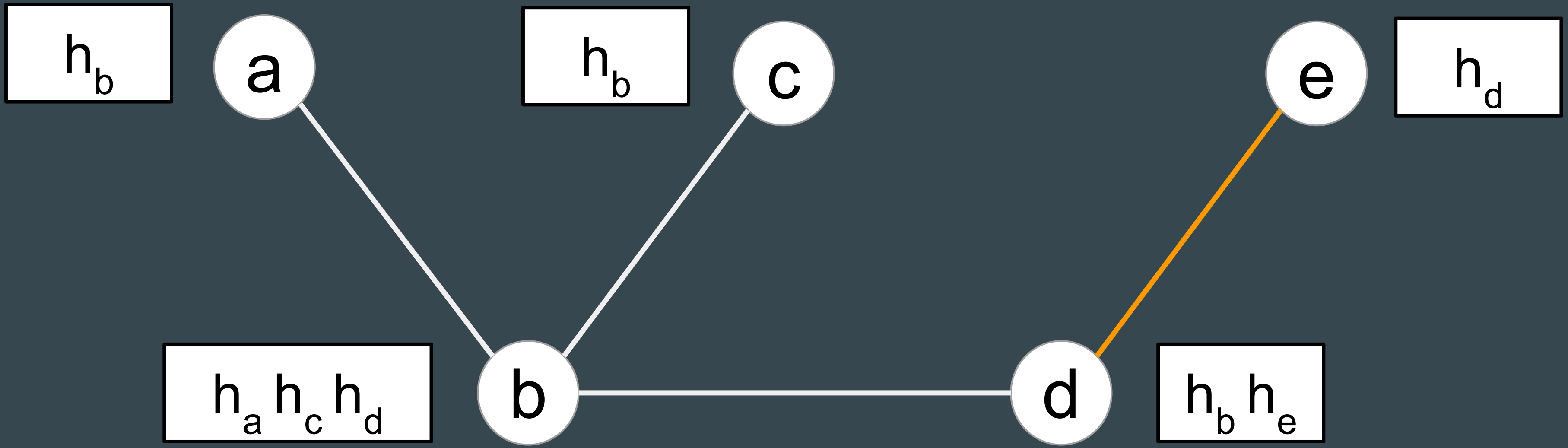
Message Passing Neural Network

Every node receives a message from its neighbour



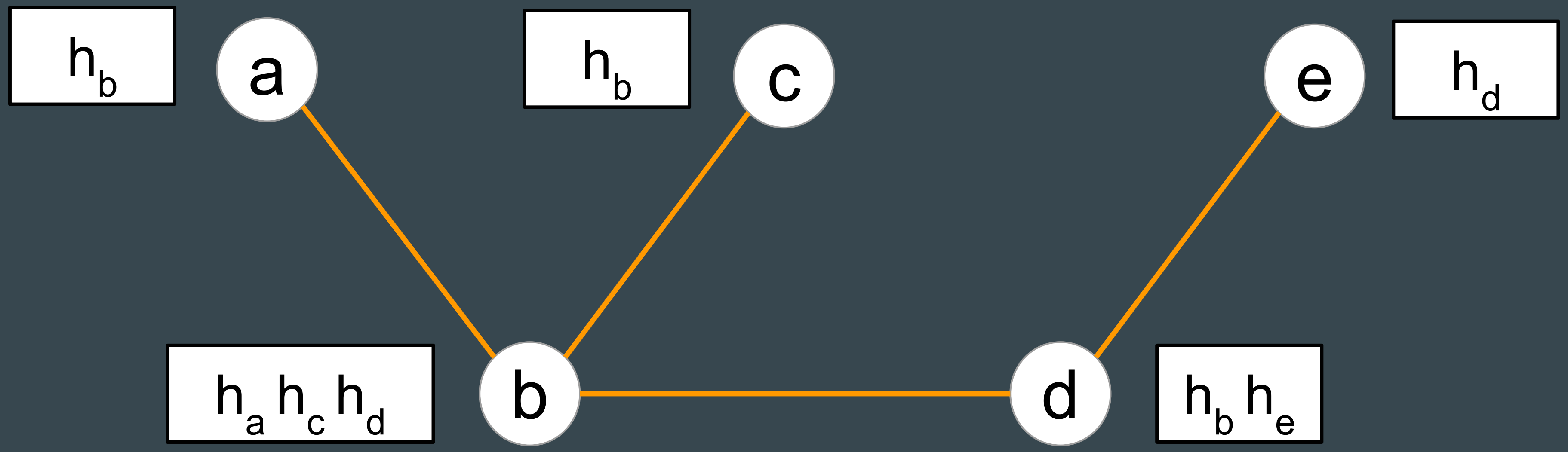
Message Passing Neural Network

Every node receives a message from its neighbour



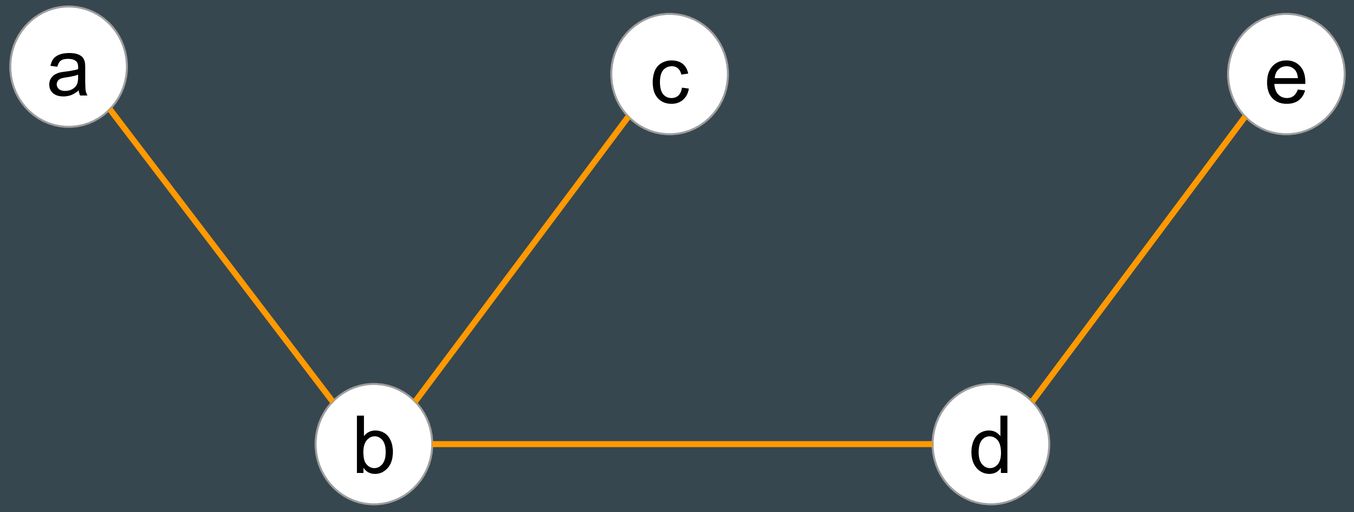
Message Passing Neural Network

Nodes updated their embedding using the messages



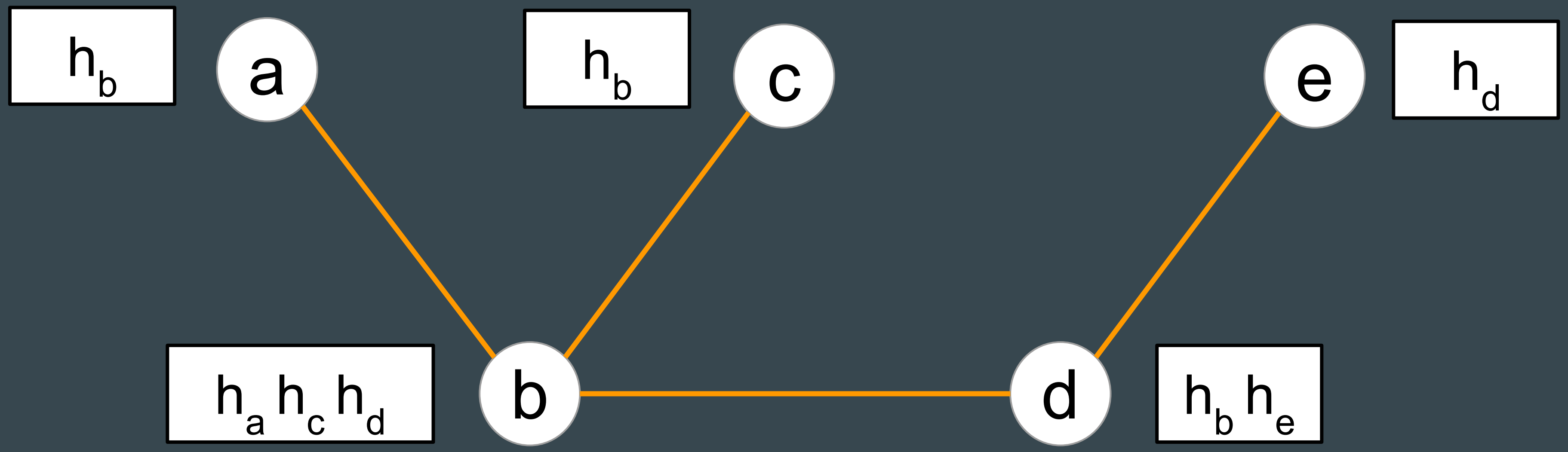
Message Passing Neural Network

Every node receives a message from its neighbour



Message Passing Neural Network

Nodes updated their embedding using the messages



Message Passing Phase

Every node receives a message from its neighbour

Message Passing Phase

Every node receives a message from its neighbour

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

Message Passing Phase

Every node receives a message from its neighbour

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

M could be *concatenation*, *element-wise product*, NN, etc

Message Passing Phase

Nodes updated their embedding using the messages

Message Passing Phase

Nodes updated their embedding using the messages

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

Message Passing Phase

Nodes updated their embedding using the messages

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

U could be *element-wise product*, NN, etc

Message Passing Phase

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

Message Passing Neural Network

Message Passing Phase

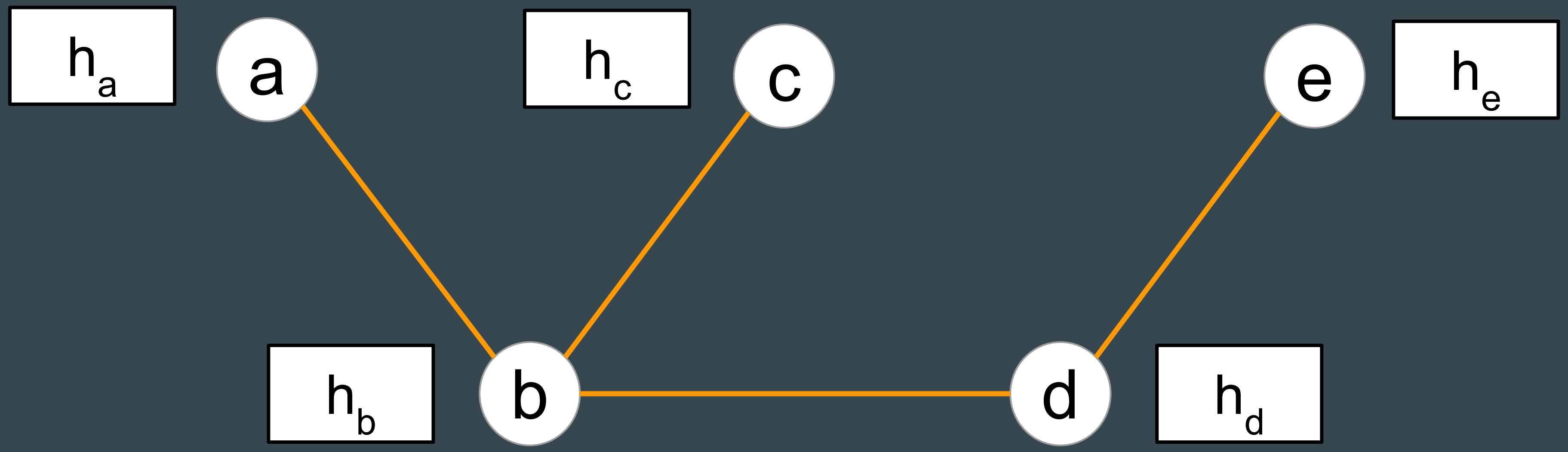
Read Phase

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

Read Phase

Combine embedding for all the nodes



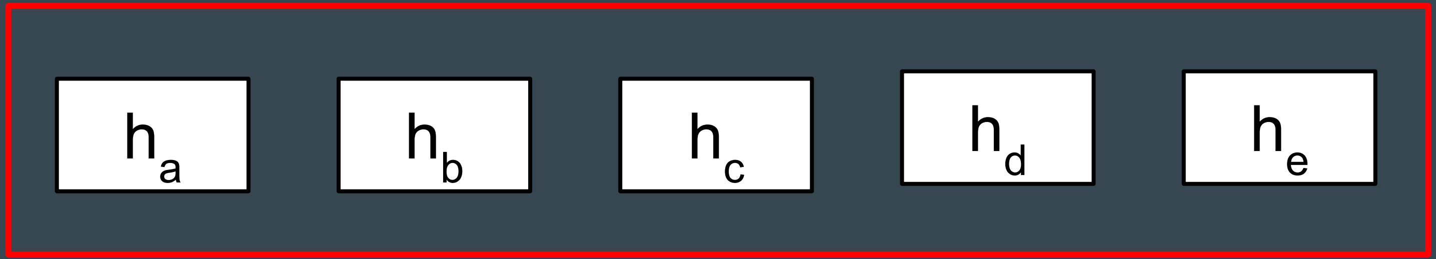
Read Phase

Combine embedding for all the nodes

$\boxed{h_a}$ $\boxed{h_b}$ $\boxed{h_c}$ $\boxed{h_d}$ $\boxed{h_e}$

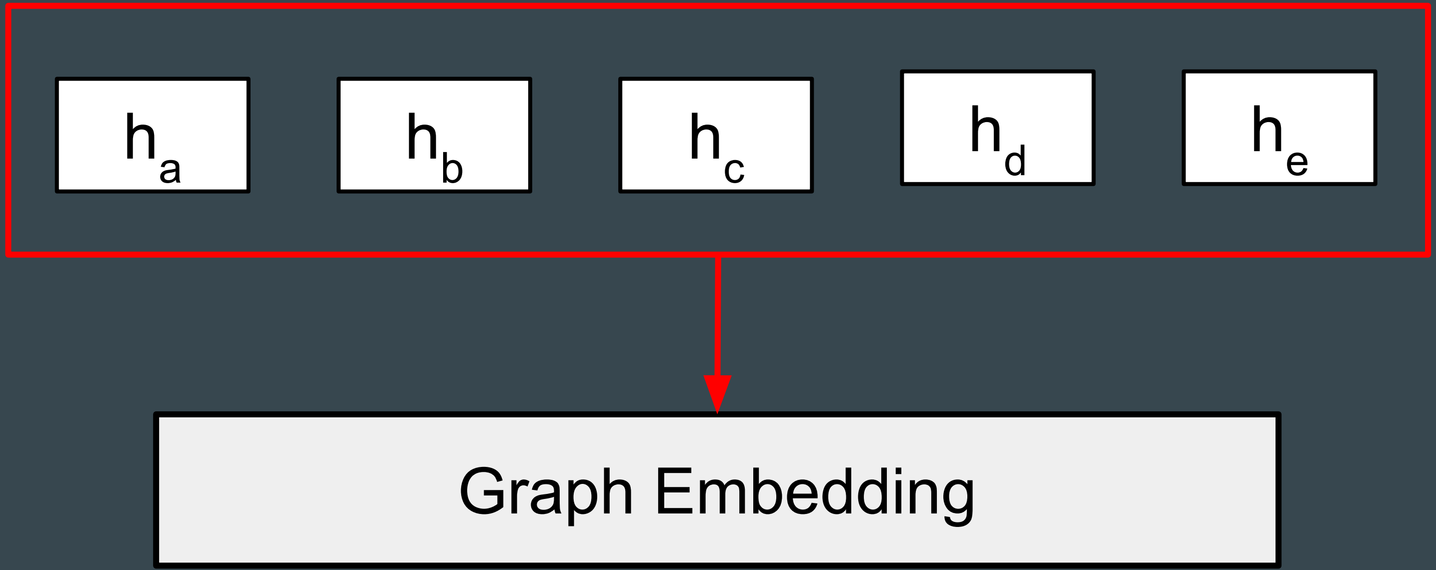
Read Phase

Combine embedding for all the nodes



Read Phase

Combine embedding for all the nodes



Read Phase

$$\hat{y} = R(\{h_v^T \mid v \in G\})$$

Read Phase

$$\hat{y} = R(\{h_v^T \mid v \in G\})$$

R could be any operation which is invariant to node permutation.

Message Passing Neural Network

Message Passing Phase

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

Read Phase

$$\hat{y} = R(\{h_v^T \mid v \in G\}).$$

Message Passing Neural Network

- M - Message Function
- U - (Vertex) Update Function
- R - Readout Function
- All are learnt differentiable functions

Convolutional Networks for Learning Molecular Fingerprints

Variants of Message Passing Neural Network

$$M(h_v, h_w, e_{vw}) = (h_w, e_{vw})$$

Convolutional Networks for Learning Molecular Fingerprints

Variants of Message Passing Neural Network

$$M(h_v, h_w, e_{vw}) = (h_w, e_{vw})$$

Convolutional Networks for Learning Molecular Fingerprints

Variants of Message Passing Neural Network

$$M(h_v, h_w, e_{vw}) = (h_w, e_{vw})$$

$$U_t(h_v^t, m_v^{t+1}) = \sigma(H_t^{\deg(v)} m_v^{t+1})$$

Convolutional Networks for Learning Molecular Fingerprints

Variants of Message Passing Neural Network

$$M(h_v, h_w, e_{vw}) = (h_w, e_{vw})$$

$$U_t(h_v^t, m_v^{t+1}) = \sigma(H_t^{\deg(v)} m_v^{t+1})$$

Convolutional Networks for Learning Molecular Fingerprints

Variants of Message Passing Neural Network

$$M(h_v, h_w, e_{vw}) = (h_w, e_{vw})$$

$$U_t(h_v^t, m_v^{t+1}) = \sigma(H_t^{\deg(v)} m_v^{t+1})$$

$$R = f\left(\sum_{v,t} \mathrm{softmax}(W_t h_v^t)\right)$$

Gated Graph Neural Networks

Variants of Message Passing Neural Network

$$M_t(h_v^t, h_w^t, e_{vw}) = A_{e_{vw}} h_w$$

Gated Graph Neural Networks
Variants of Message Passing Neural Network

$$M_t(h_v^t, h_w^t, e_{vw}) = A_{e_{vw}} h_w$$

Gated Graph Neural Networks

Variants of Message Passing Neural Network

$$M_t(h_v^t, h_w^t, e_{vw}) = A_{e_{vw}} h_w$$

$$U_t = \mathrm{GRU}(h_v^t, m_v^{t+1})$$

Gated Graph Neural Networks

Variants of Message Passing Neural Network

$$M_t(h_v^t, h_w^t, e_{vw}) = A_{e_{vw}} h_w$$

$$U_t = \mathrm{GRU}(h_v^t, m_v^{t+1})$$

$$R = \sum_{v \in V} \sigma\left(i(h_v^{(T)}, h_v^0)\right) \odot \left(j(h_v^{(T)})\right)$$

Interaction Networks

Variants of Message Passing Neural Network

$$M(h_v, h_w, e_{vw}) = f(h_v, h_w, e_{vw}))$$

$$U(h_v, x_v, m_v = g(h_v, x_v, m_v))$$

$$R = f(\sum_{v \in G} h_v^T)$$

Molecular Graph Convolutions

Variants of Message Passing Neural Network

$$\mathrm{M}(\mathrm{h}_v^t, h_w^t, e_{vw}^t) = e_{vw}^t$$

Molecular Graph Convolutions

Variants of Message Passing Neural Network

$$\mathrm{M}(\mathrm{h}_v^t, h_w^t, e_{vw}^t) = e_{vw}^t$$

Molecular Graph Convolutions

Variants of Message Passing Neural Network

$$\mathrm{M}(\mathrm{h}_v^t, h_w^t, e_{vw}^t) = e_{vw}^t$$

$$\mathrm{U}_t(h_v^t, m_v^{t+1}) = \alpha(W_1(\alpha(W_0 h_v^t), m_v^{t+1}))$$

$$\mathrm{e}_{vw}^{t+1} = \alpha(W_4(\alpha(W_2, e_{vw}^t), \alpha(W_3(h_v^t, h_w^t))))$$

Deep Tensor Neural Networks

Variants of Message Passing Neural Network

$$M_t = \tanh\left(W^{fc}((W^{cf}h_w^t + b_1) \odot (W^{df}e_{vw} + b_2))\right)$$

$$U_t(h_v^t, m_v^{t+1}) = h_v^t + m_v^{t+1}$$

$$R = \sum_v \mathrm{NN}(h_v^T)$$

Graph Convolutional Network

Variants of Message Passing Neural Network

$$M_t(h_v^t, h_w^t) = C_{vw}^t h_w^t$$

Graph Convolutional Network

Variants of Message Passing Neural Network

$$M_t(h_v^t, h_w^t) = C_{vw}^t h_w^t$$

$$U_t(h_v^t, m_v^{t+1}) = \sigma(m_v^{t+1})$$

Graph Convolutional Network

Variants of Message Passing Neural Network

$$M_t(h_v^t, h_w^t) = c_{vw} h_w^t$$

Graph Convolutional Network

Variants of Message Passing Neural Network

$$M_t(h_v^t, h_w^t) = c_{vw} h_w^t$$

$$U_v^t(h_v^t, m_v^{t+1}) = \mathrm{ReLU}(W^t m_v^{t+1})$$

Graph Convolutional Network

Variants of Message Passing Neural Network

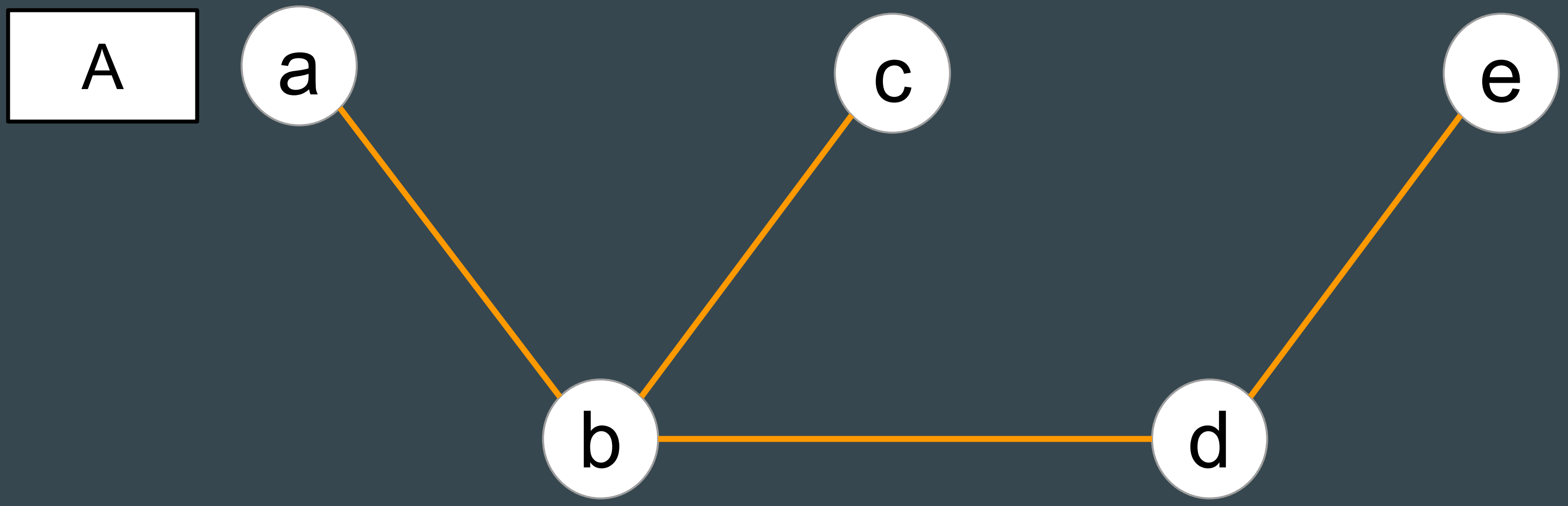
$$M_t(h_v^t, h_w^t) = c_{vw} h_w^t$$

$$U_v^t(h_v^t, m_v^{t+1}) = \mathrm{ReLU}(W^t m_v^{t+1})$$

$$c_{vw} = \left(\deg(v)\deg(w)\right)^{-1/2} A_{vw}$$

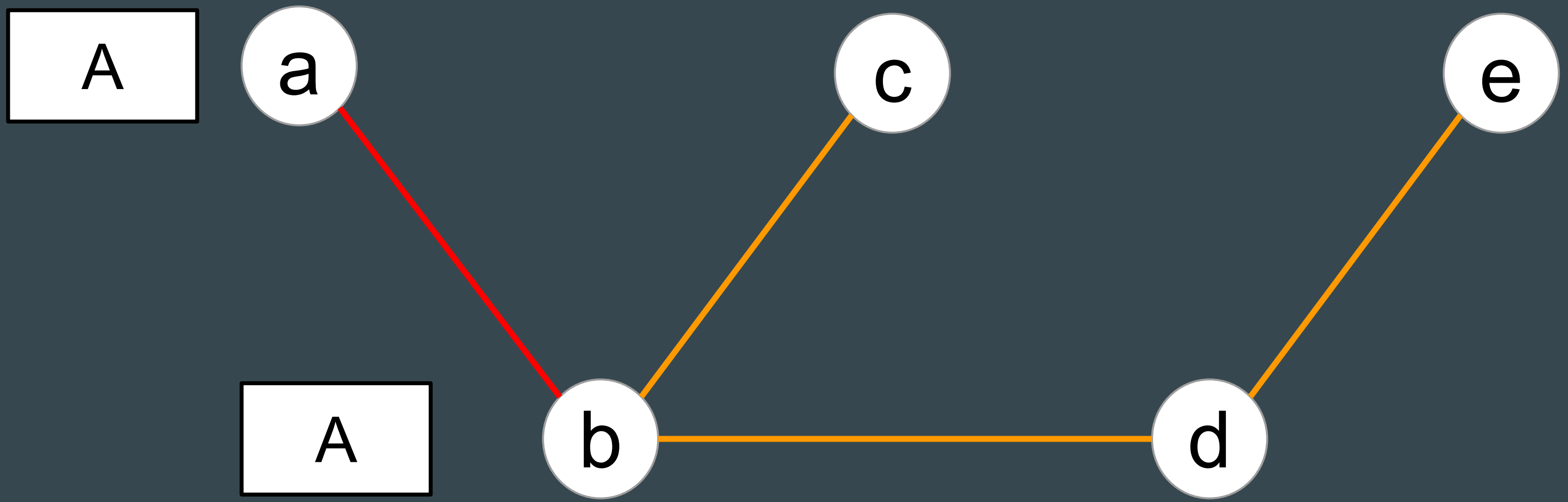
Message Passing Neural Network

Trajectory of a Message



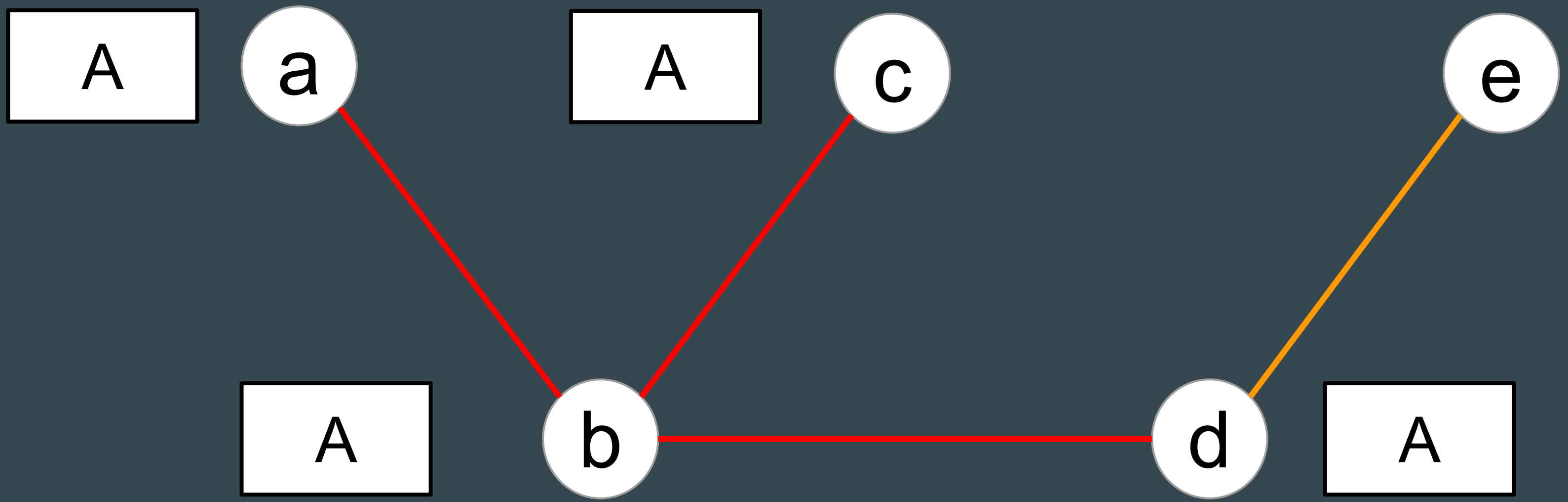
Message Passing Neural Network

Trajectory of a Message



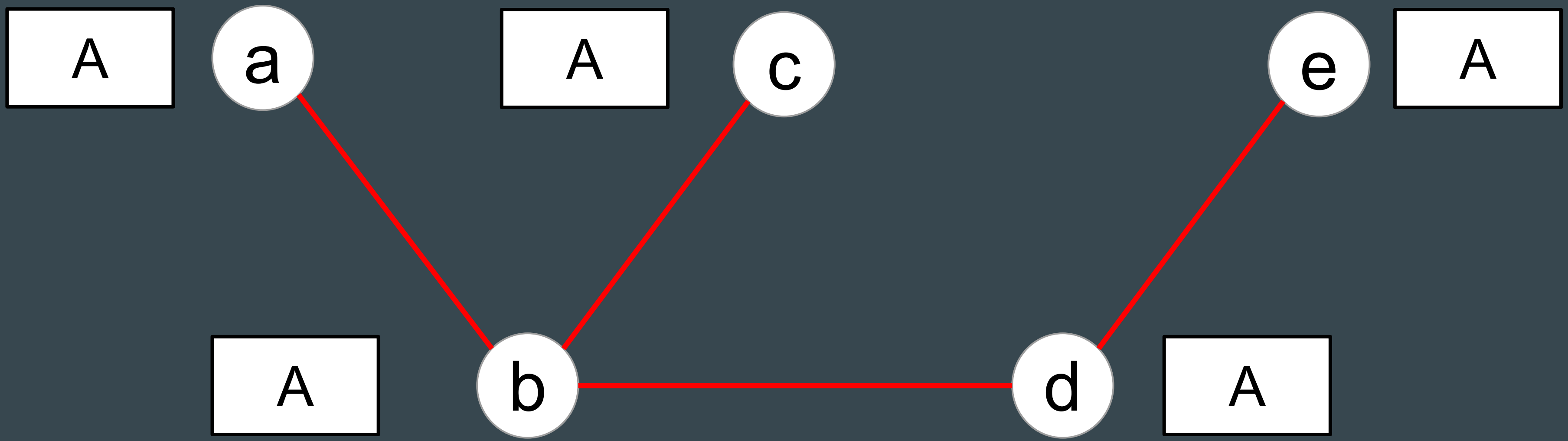
Message Passing Neural Network

Trajectory of a Message



Message Passing Neural Network

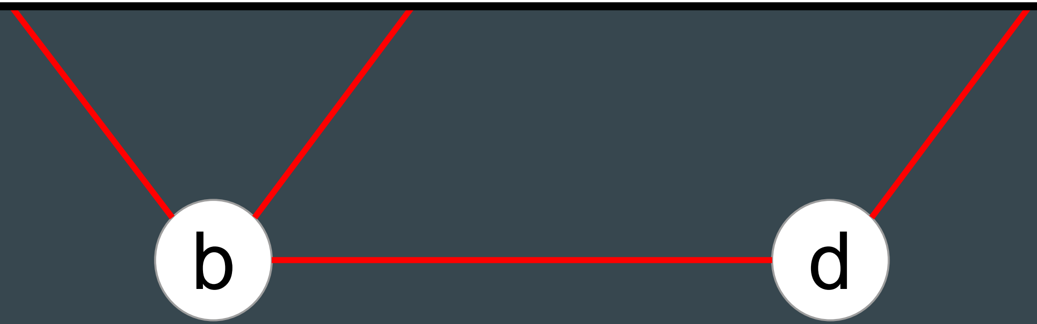
Trajectory of a Message



Message Passing Neural Network

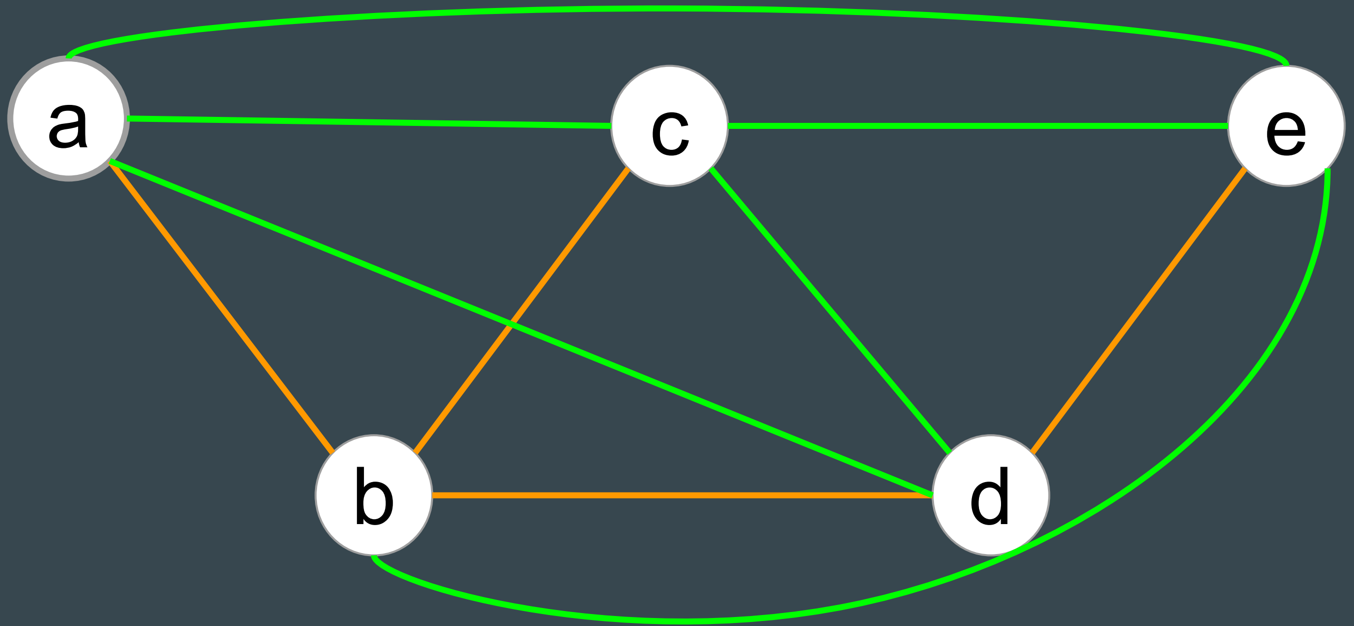
Trajectory of a Message

Message could take a while to reach other nodes!



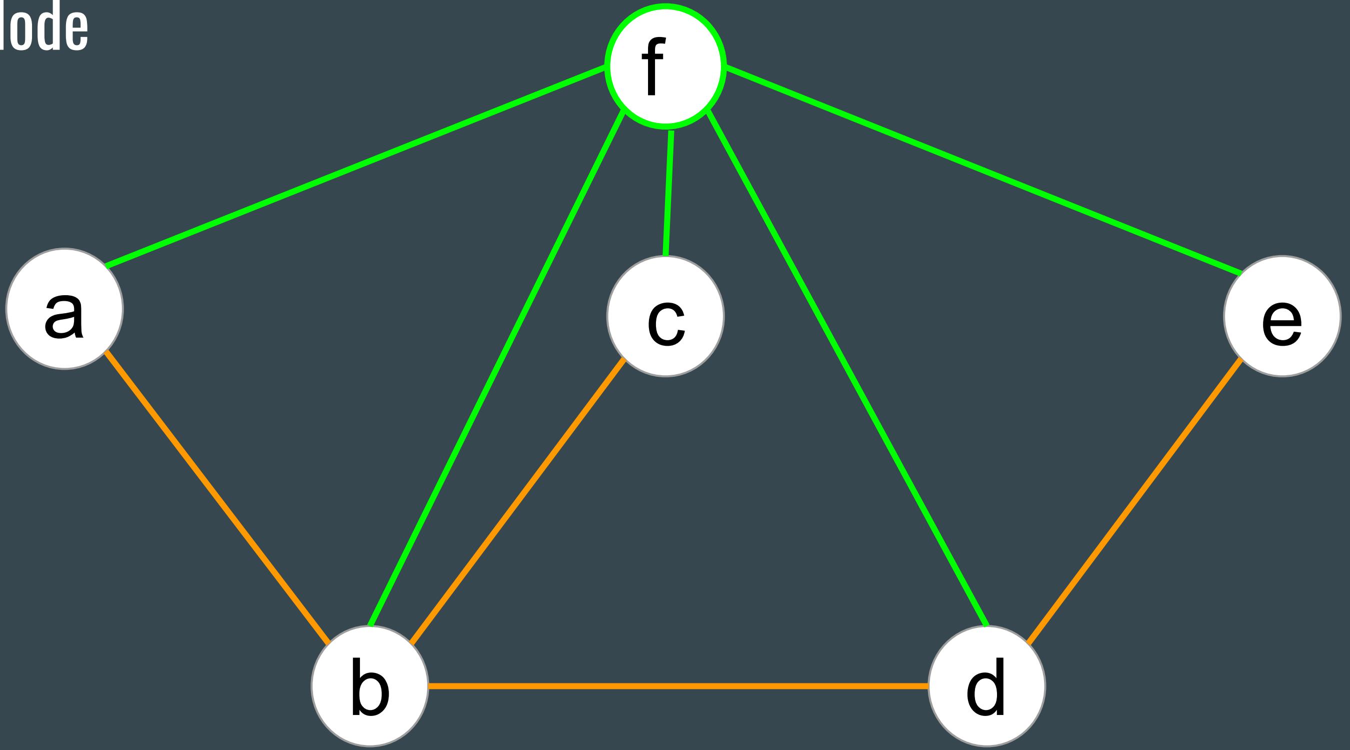
Message Passing Neural Network

Virtual Edges



Message Passing Neural Network

Virtual Node



References

1. Neural Message Passing for Quantum Chemistry Gilmer et al (2017)
2. Convolutional Networks for Learning Molecular Fingerprints, Duvenaud et al. (2015)
3. Gated Graph Neural Networks (GG-NN), Li et al. (2016)
4. Interaction Networks, Battaglia et al. (2016)
5. Molecular Graph Convolutions, Kearnes et al. (2016)
6. Deep Tensor Neural Networks, Schutt et al. ¨ (2017)
7. Semi-Supervised Classification with Graph Convolutional Networks. Kipf et al. (2016)

Thank You