Graph Neural Networks

CSE 849 Deep Learning
Spring 2025

Zijun Cui

Slides are revised based on Minji Yoon (CMU) - Guest lecture at 10707: Introduction to Deep Learning

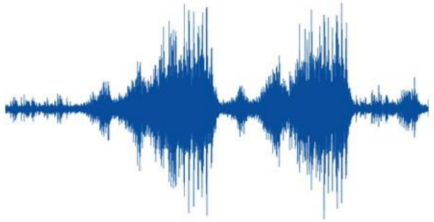
Grades are out

- ✓ Project 2
- ✓ Project 3
- ✓ HW2
  - Many students didn't consider $\mathrm{i} \neq j$.
  - This question is asking to match those shown in Slide 24, which contains both $\mathrm{i} = j$ and $\mathrm{i} \neq j$.
  - The question itself is a bit ambiguous since I use the notation $\frac{\partial u_i}{\partial z_i}$
  - To take this into account, we will update grades: instead of deducting 5 points, we will deduct 2 points.

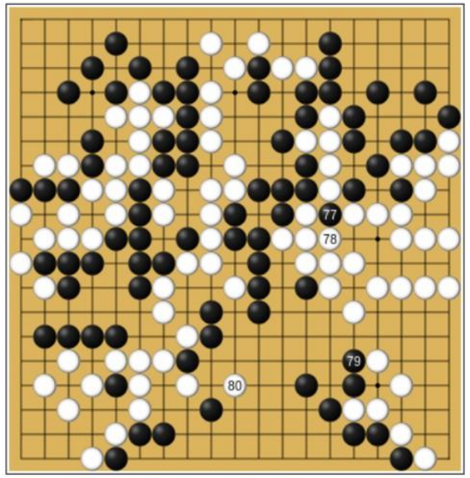
Traditional **Neural Networks**

IMAGENET

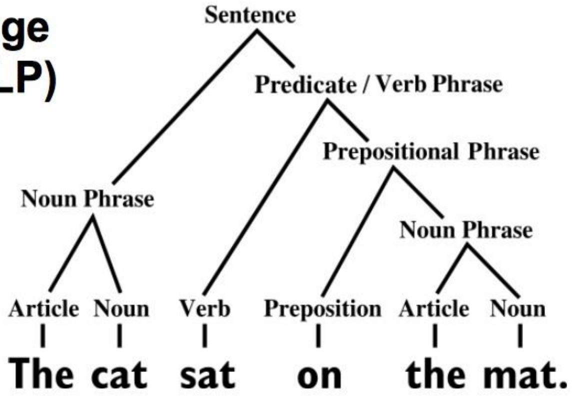
Speech data



Grid games



Natural language processing (NLP)



...

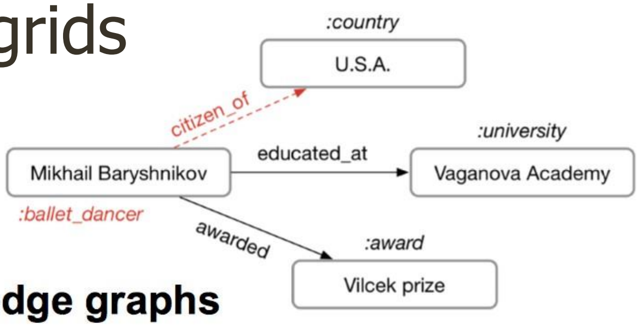
Deep neural nets that exploit:

- translation equivariance (weight sharing)
- hierarchical compositionality

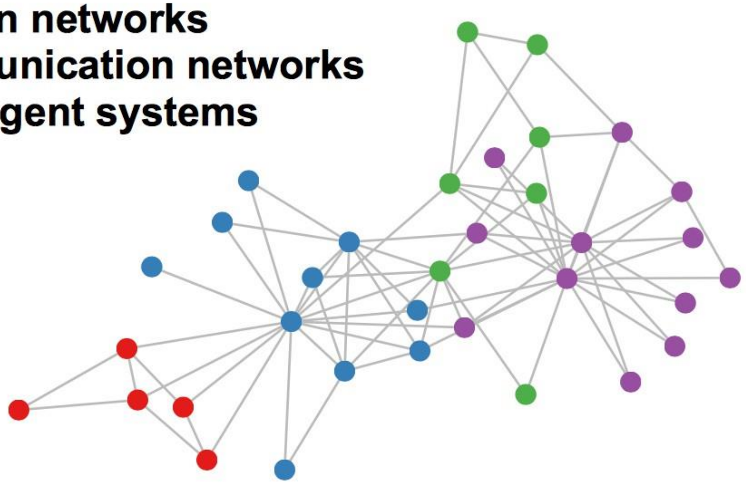
*slide from Thomas Kipf, **University of Amsterdam**

Graph: structured Data

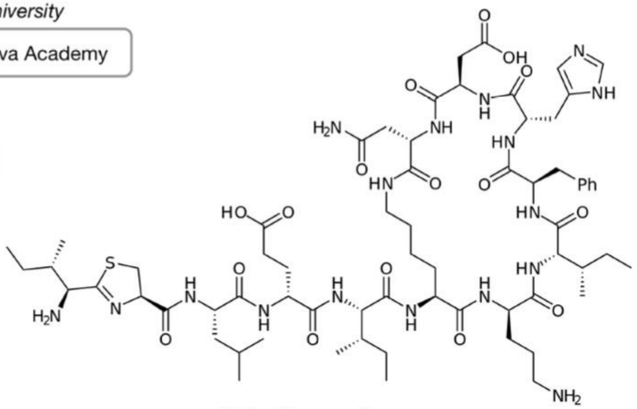
A lot of real-world data does not “live” on grids



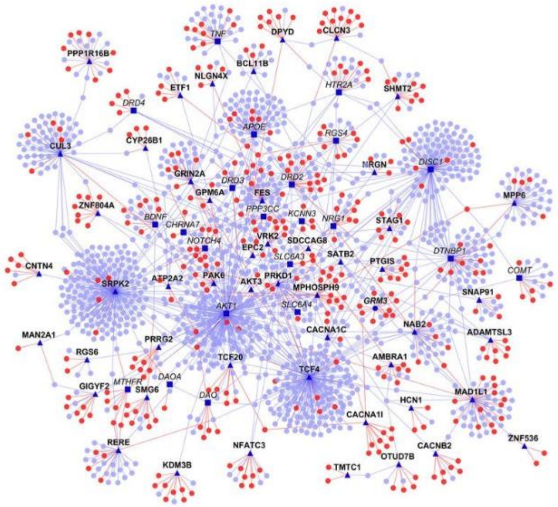
Social networks
Citation networks
Communication networks
Multi-agent systems



Knowledge graphs



Molecules

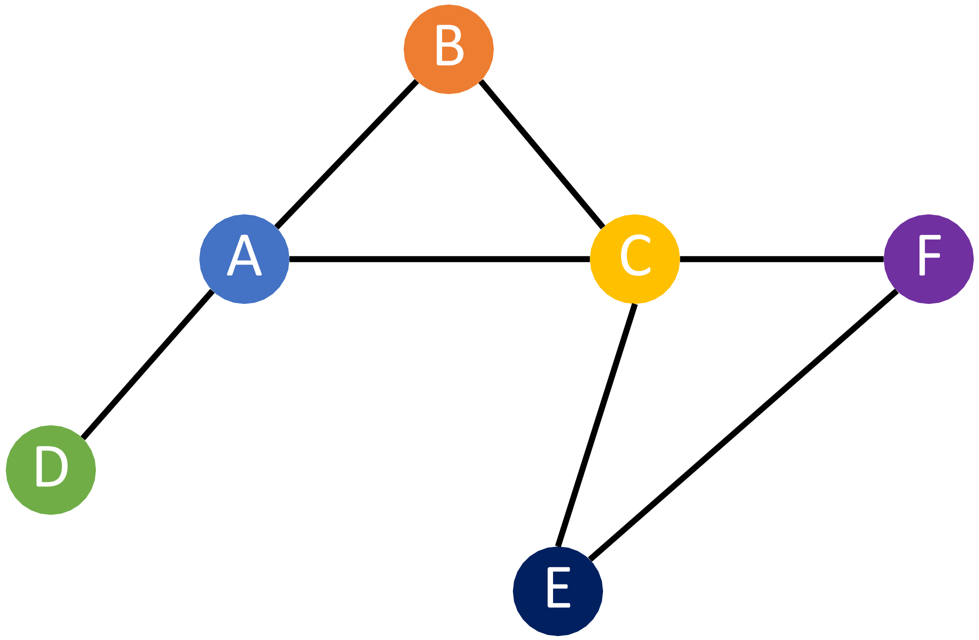


Protein interaction networks

Road maps

Standard **CNN** and **RNN** architectures don’t work on this data

What is a graph?



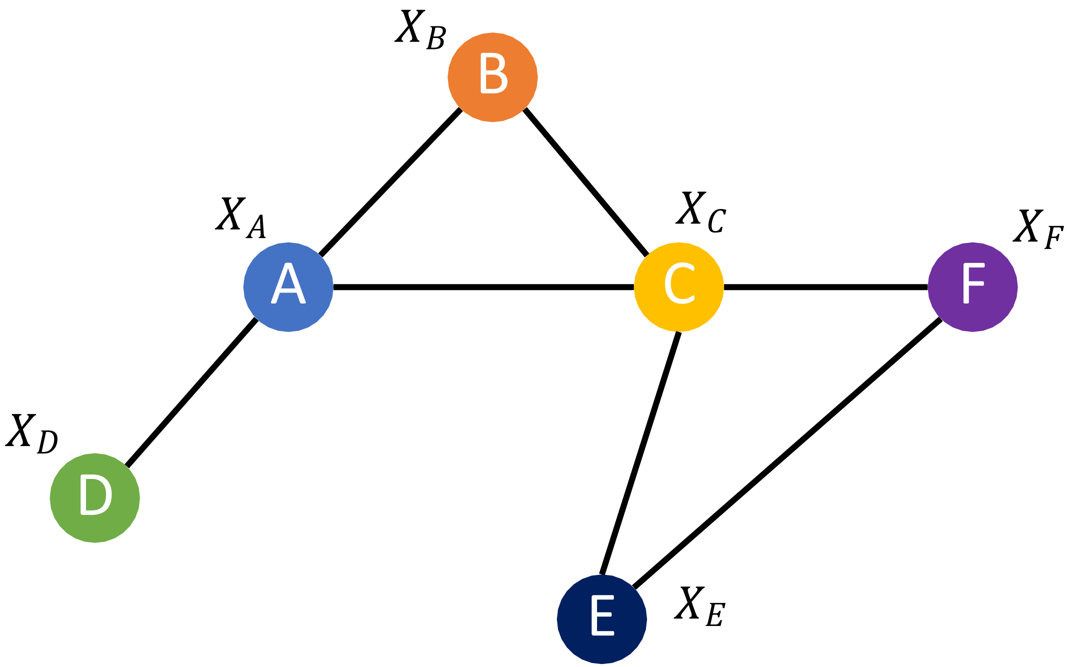
A graph is composed of

- **Nodes** (also called vertices)
- **Edges** connecting a pair of nodes

presented in an **adjacency matrix**

	A	B	C	D	E	F
A		1	1	1		
B	1		1			
C	1	1			1	1
D	1					
E			1			1
F			1		1	

What is a graph?



A graph is composed of

- **Nodes** (also called vertices)
- **Edges** connecting a pair of nodes

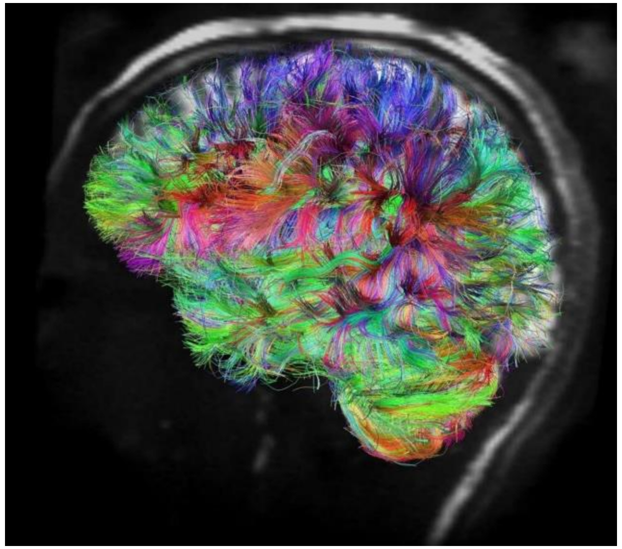
presented in an **adjacency matrix**

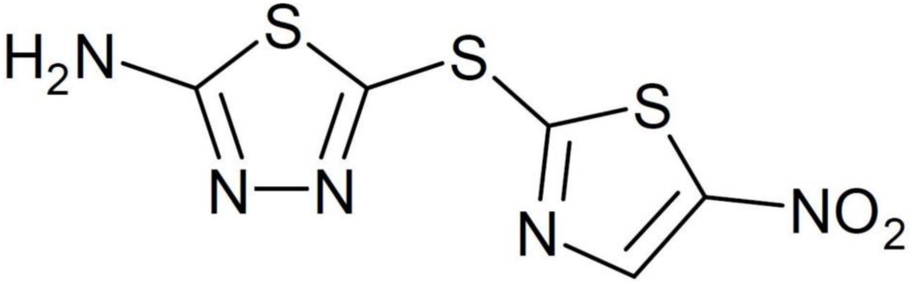
Nodes can have **feature vectors**

A	X_A
B	X_B
C	X_C
D	X_D
E	X_E
F	X_F

(slide: P. Veličković)

Graphs are everywhere





Graph Neural Networks have a large impact on…

DeepMind › Blog › Traffic prediction with advanced Graph Neural Networks

BLOG POST
RESEARCH

03 SEP 2020

Traffic prediction with advanced Graph Neural Networks

Food Discovery with Uber Eats: Using Graph Learning to Power Recommendations

Ankit Jain, Isaac Liu, Ankur Sarda, and Piero Molino

December 4, 2019

0

Pinterest Engineering

Aug 15, 2018 · 8 min read

PinSage: A new graph convolutional neural network for web-scale recommender systems

Ruining He | Pinterest engineer, Pinterest Labs

Web image search gets better with graph neural networks

ı to image search uses images returned by traditional search
des in a graph neural network through which similarity signals are
ieving improved ranking in cross-modal retrieval.

ıral Network

ER LABS Europe

FOLLOW

amazon | science

PUBLICATION

P-Companion: A principled framework for diversified complementary product recommendation

By Junheng Hao, Tong Zhao, Jin Li, Xin Luna Dong, Christos Faloutsos, Yizhou Sun, Wei Wang

2020

Graph Neural Networks have a large impact on...

GCN-RL Circuit Designer: Transferable Transistor Sizing with Graph Neural Networks and Reinforcement Learning

Hanrui Wang[1], Kuan Wang[1], Jiacheng Yang[1], Linxiao Shen[2], Nan Sun[2], Hae-Seung Lee[1], Song Han[1]

[1]Massachusetts Institute of Technology

[2]UT Austin

HAN LAB Hardware, AI and Neural-nets

TEXAS

npj | computational materials

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › npj computational materials › articles › article

Article | Open Access | Published: 03 June 2021

Benchmarking graph neural networks for materials chemistry

Victor Fung, Jiaxin Zhang, Eric Juarez & Bobby G. Sumpter

npj Computational Materials **7**, Article number: 84 (2021) | Cite this article

7807 Accesses | **7** Citations | **41** Altmetric | Metrics

The next big thing: the use of graph neural networks to discover particles

September 24, 2020 | Zack Savitsky

Share Tweet Email

Machine learning algorithms can beat the world's hardest video games in minutes and solve complex equations faster than the collective efforts of generations of physicists. But the conventional algorithms still struggle to pick out stop signs on a busy street.

Object identification continues to hamper the field of machine learning — especially when the pictures are multidimensional and complicated, like the ones particle detectors take of collisions in high-energy physics experiments. However, a new class of neural networks is helping these models boost their pattern recognition abilities, and the technology may soon be implemented in particle physics experiments to optimize data analysis.

nature View all journals Search Login

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › articles › article

Article | Published: 09 June 2021

A graph placement methodology for fast chip design

Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, William Hang, Emre Tuncer, Quoc V. Le, James Laudon, Richard Ho, Roger Carpenter & Jeff Dean

Graph Neural Networks have a large impact on...

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature > news > article

NEWS | 01 December 2021

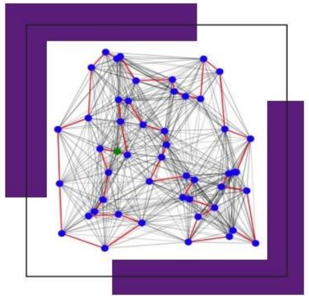
DeepMind's AI helps untangle the mathematics of knots

The machine-learning techniques could
sets.

ipam institute for pure & applied mathematics

Deep Learning and Combinatorial Optimization

February 22 - 25, 2021



Patterns

CellPress OPEN ACCESS

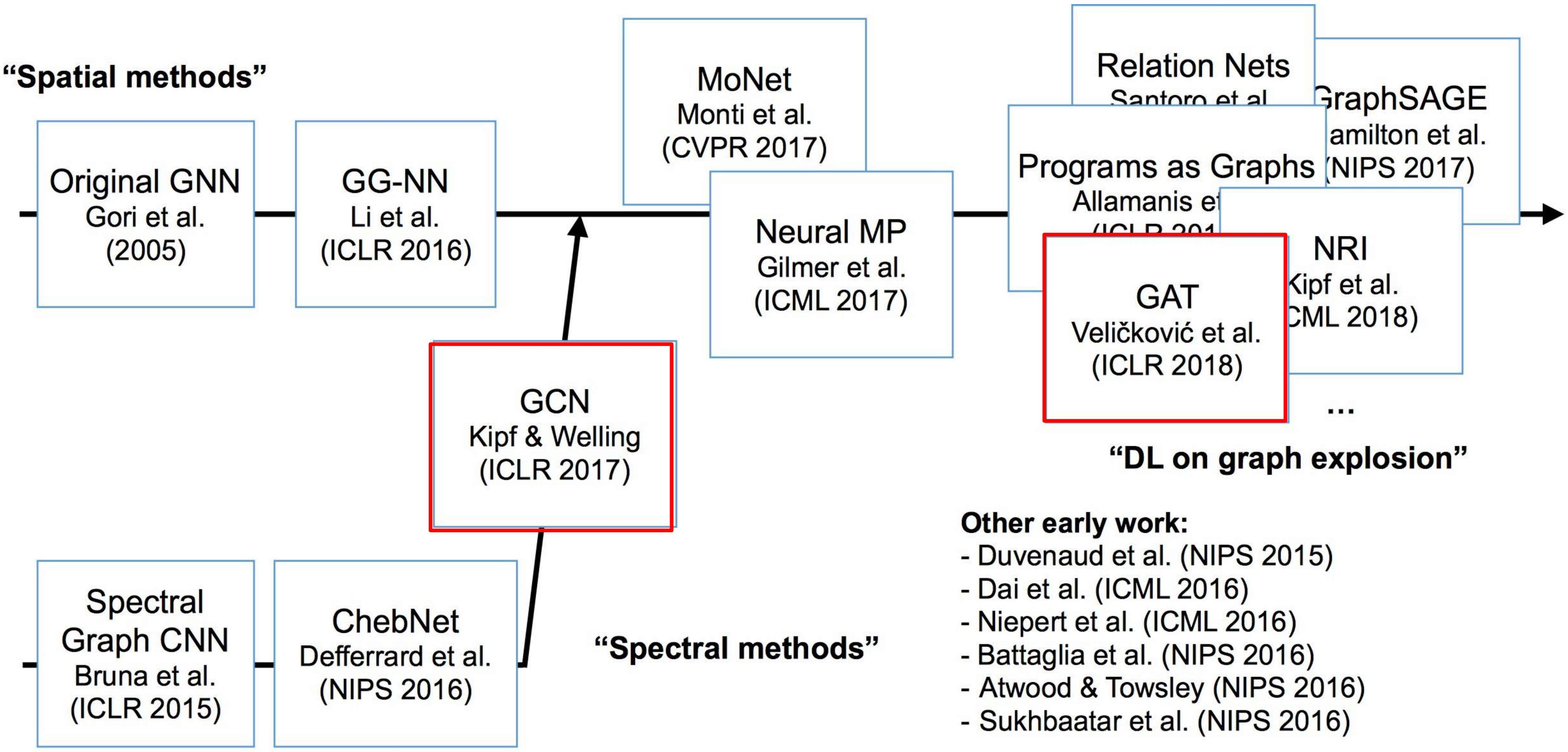
Opinion

Neural algorithmic reasoning

Petar Veličković[1,*] and Charles Blundell[1]
[1]DeepMind, London, Greater London, UK
*Correspondence: petarv@google.com
https://doi.org/10.1016/j.patter.2021.100273

We present neural algorithmic reasoning—the art of building neural networks that are able to execute algorithmic computation—and provide our opinion on its transformative potential for running classical algorithms on inputs previously considered inaccessible to them.

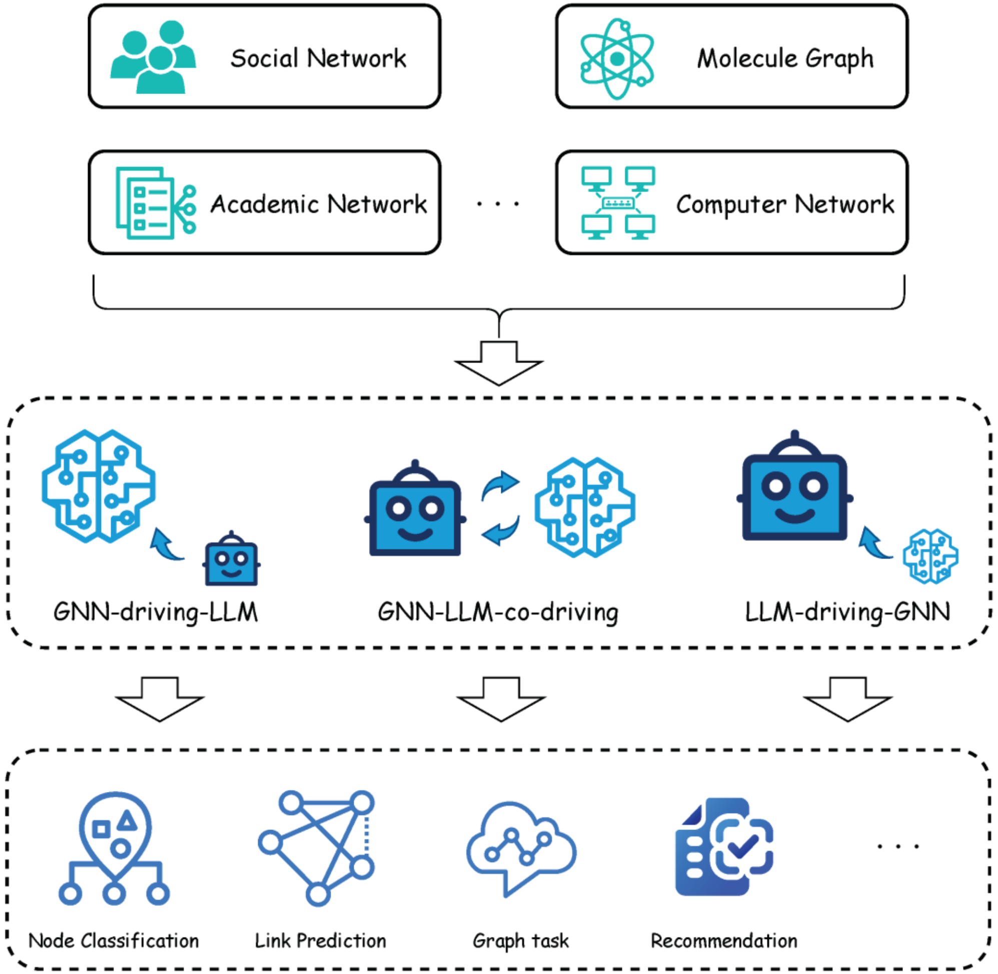
A **Brief History** of Graph Neural Nets

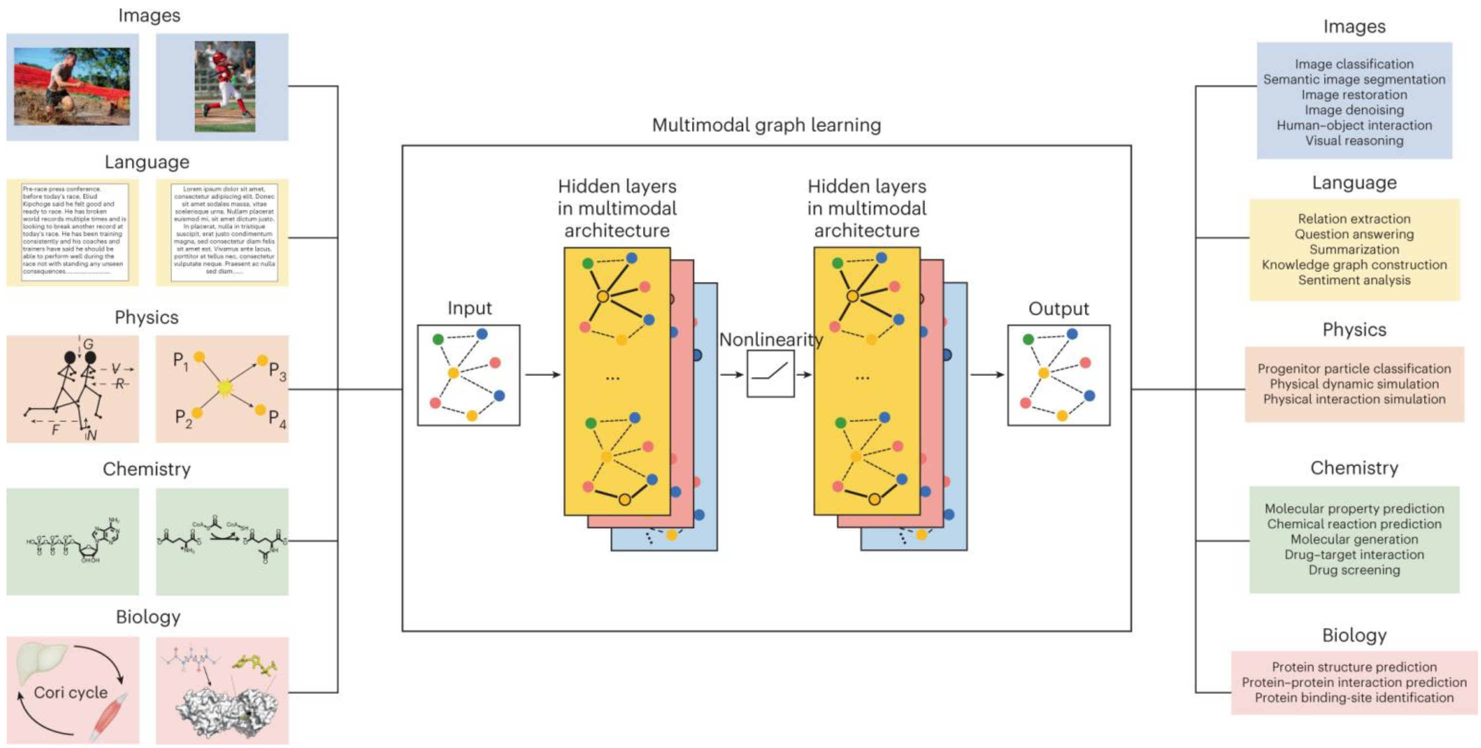


Other early work:
- Duvenaud et al. (NIPS 2015)
- Dai et al. (ICML 2016)
- Niepert et al. (ICML 2016)
- Battaglia et al. (NIPS 2016)
- Atwood & Towsley (NIPS 2016)
- Sukhbaatar et al. (NIPS 2016)

(slide inspired by Alexander Gaunt's talk on GNNs)

Remain a very hot research topic





Multimodal learning with graphs
https://www.nature.com/articles/s42256-023-00624-6

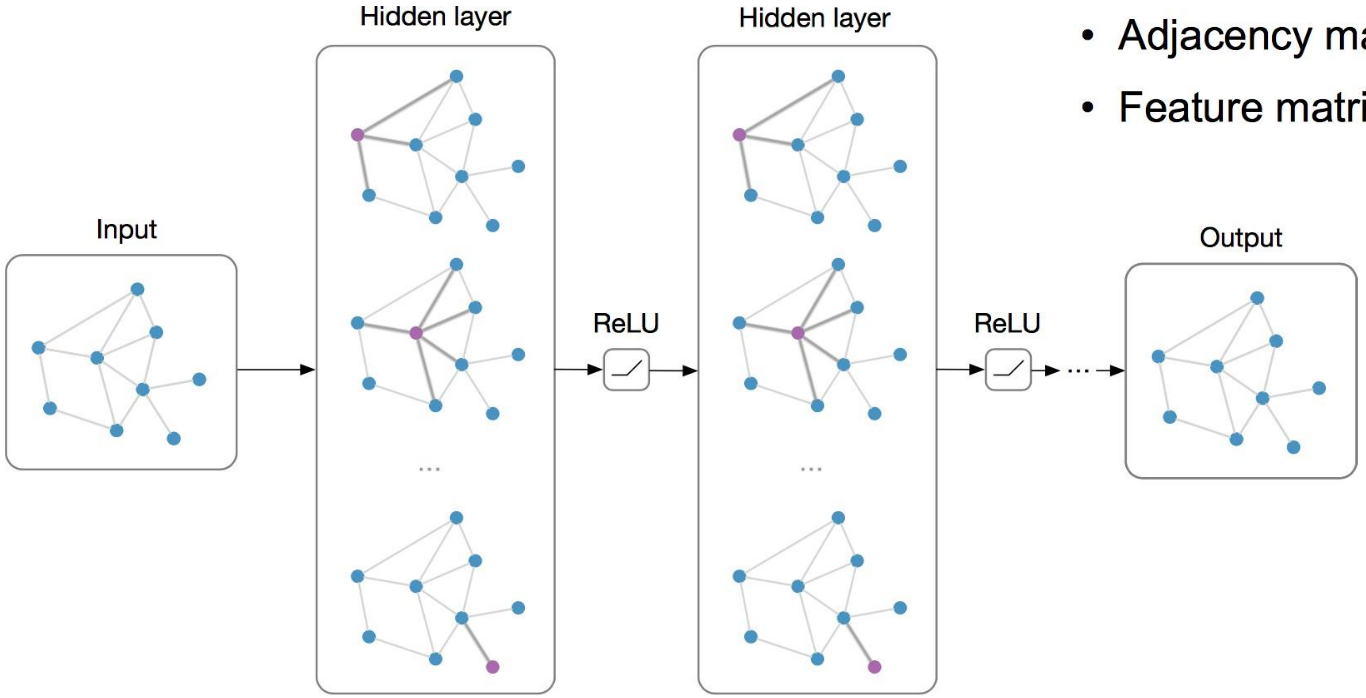
Large Language Models Meet Graph Neural Networks: A Perspective of Graph Mining
https://www.mdpi.com/2227-7390/13/7/1147

What is Graph Neural Network?

Graph Neural Networks (GNNs)

Notation: $\mathcal{G} = (\mathbf{A}, \mathbf{X})$

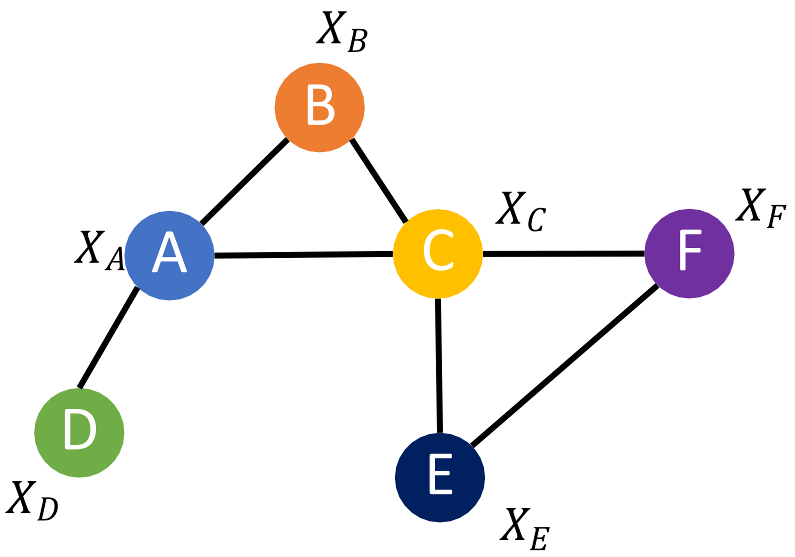
- Adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$
- Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$



Main Idea: Pass massages between pairs of nodes and agglomerate

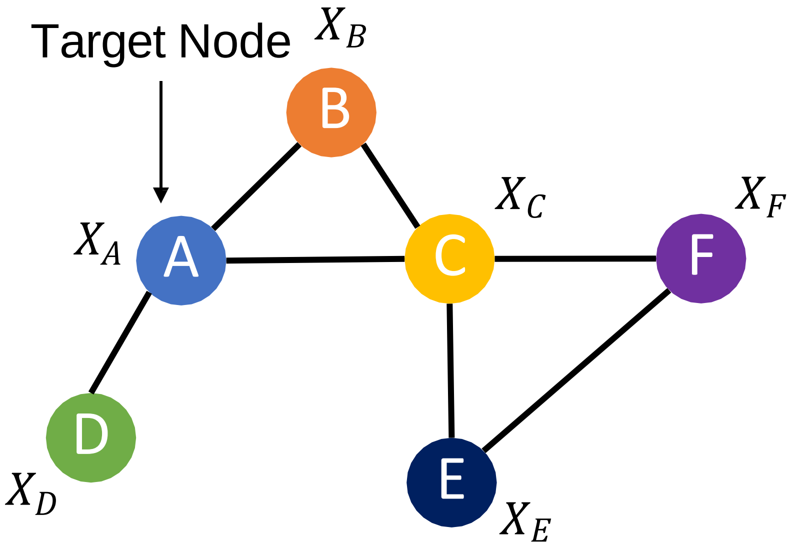
Pass massages between nodes to refine node (and possibly edge) representations

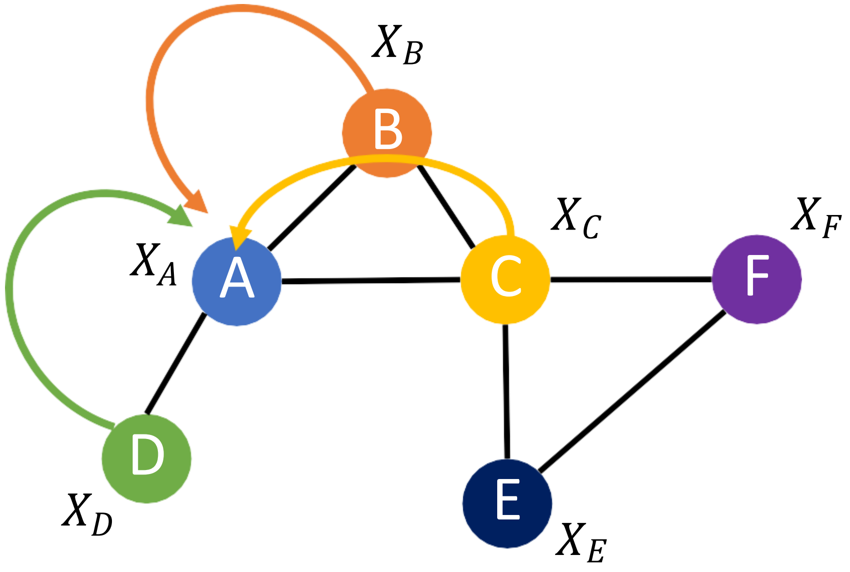
Problem definition



- **Given**
  - A graph
  - Node attributes
  - (part of nodes are labeled)
- **Find**
  - Node embeddings
- **Predict**
  - Labels for the remaining nodes (node classification task)

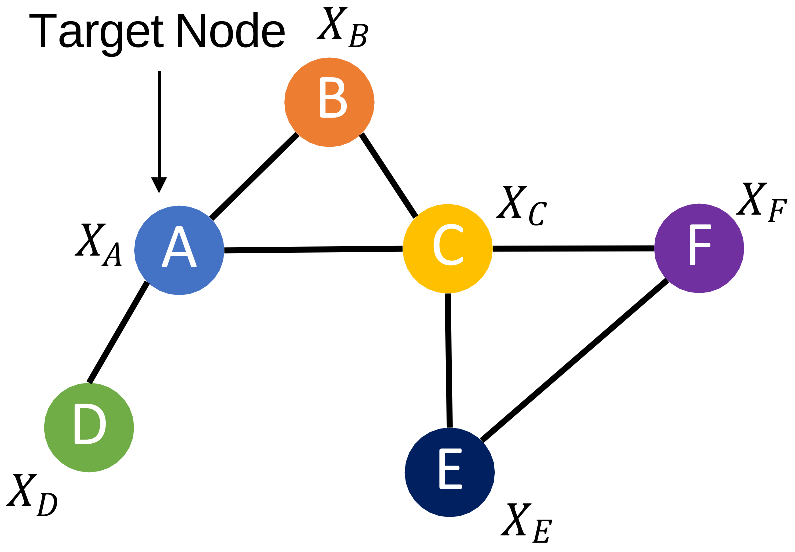
Graph Neural Networks

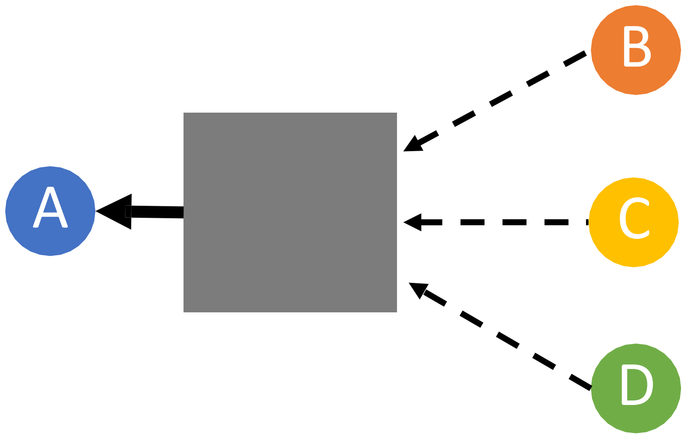




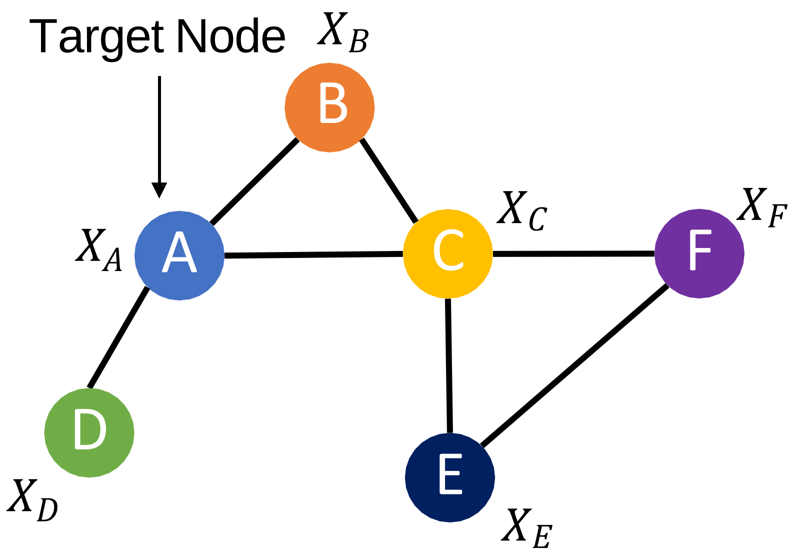
“Homophily: connected nodes are related/informative/similar”

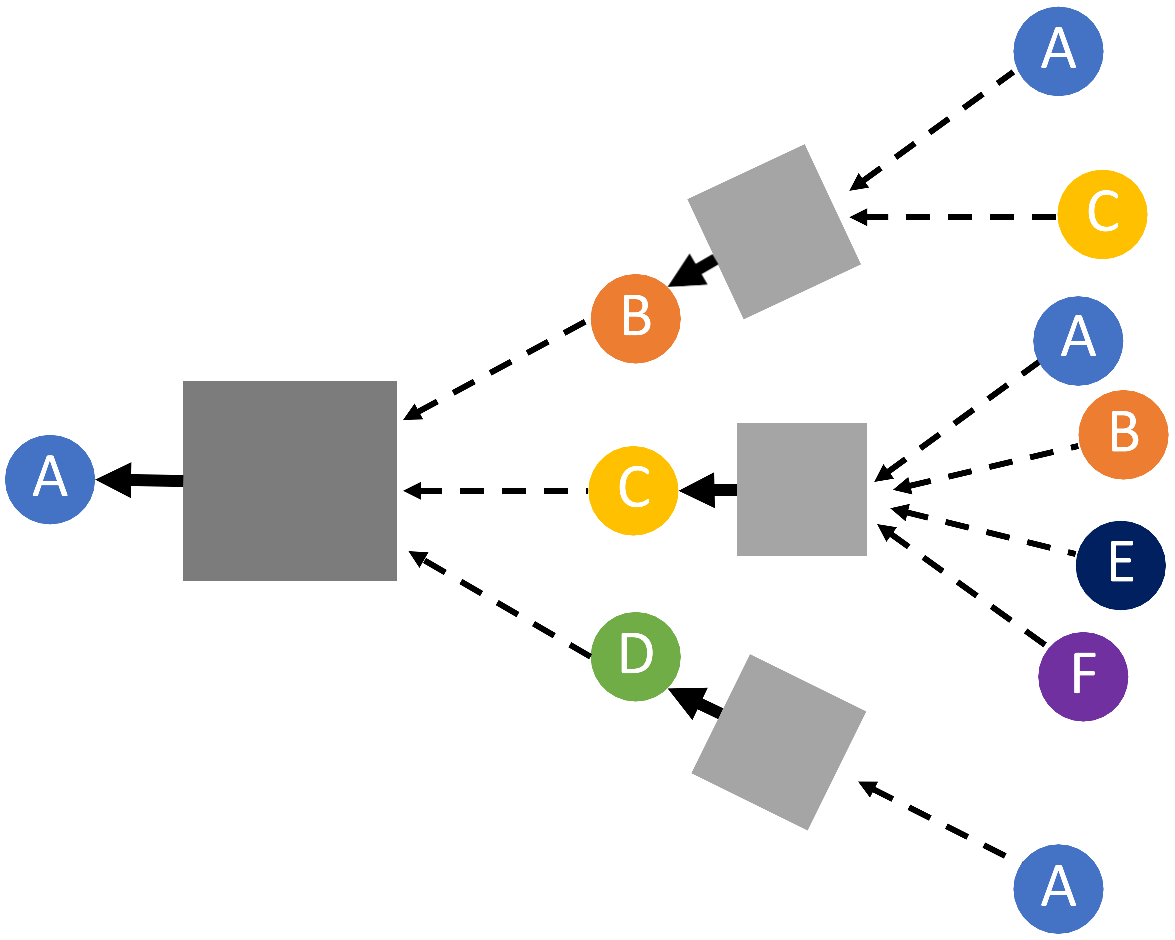
Graph Neural Networks



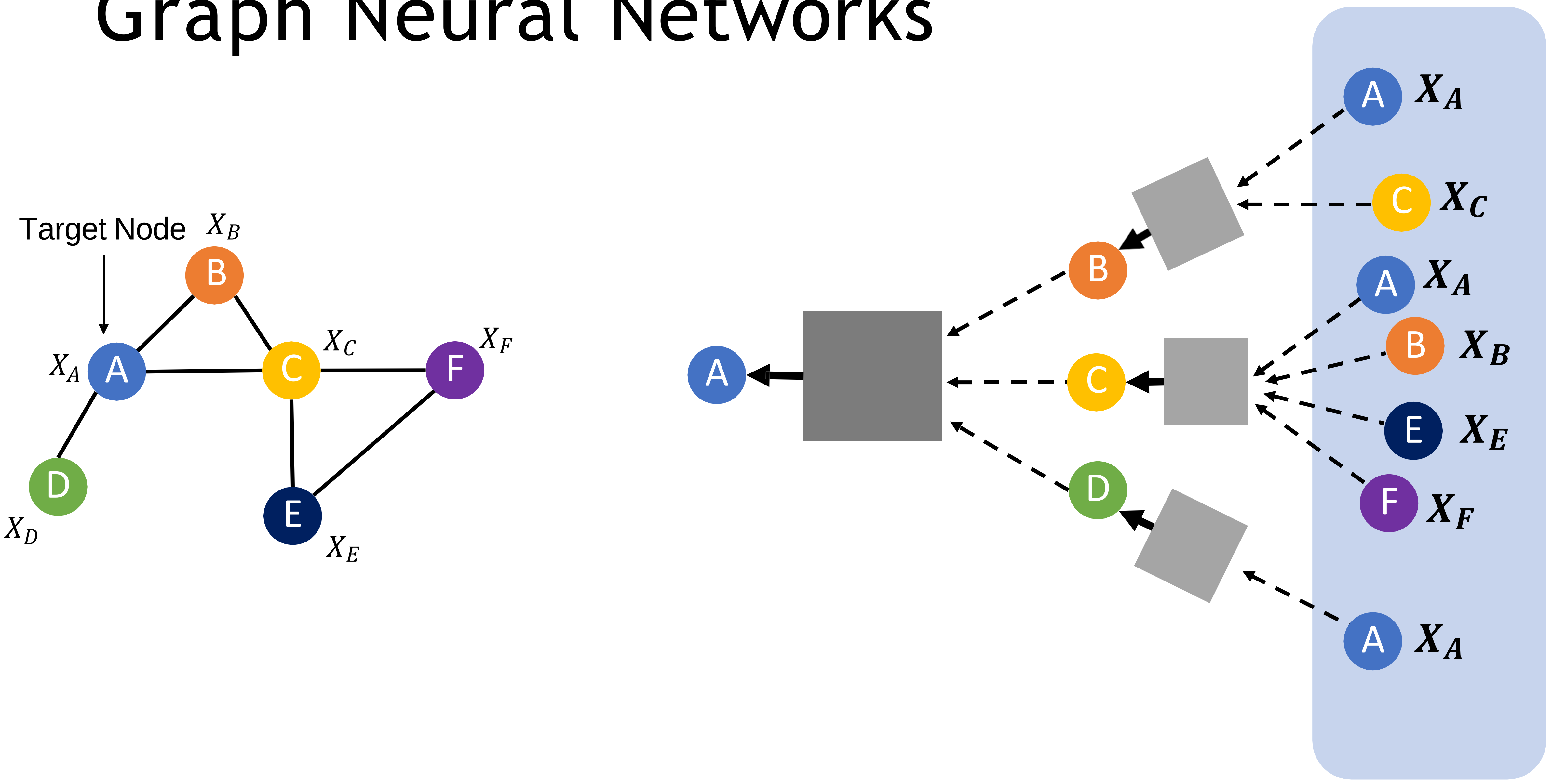


Graph Neural Networks

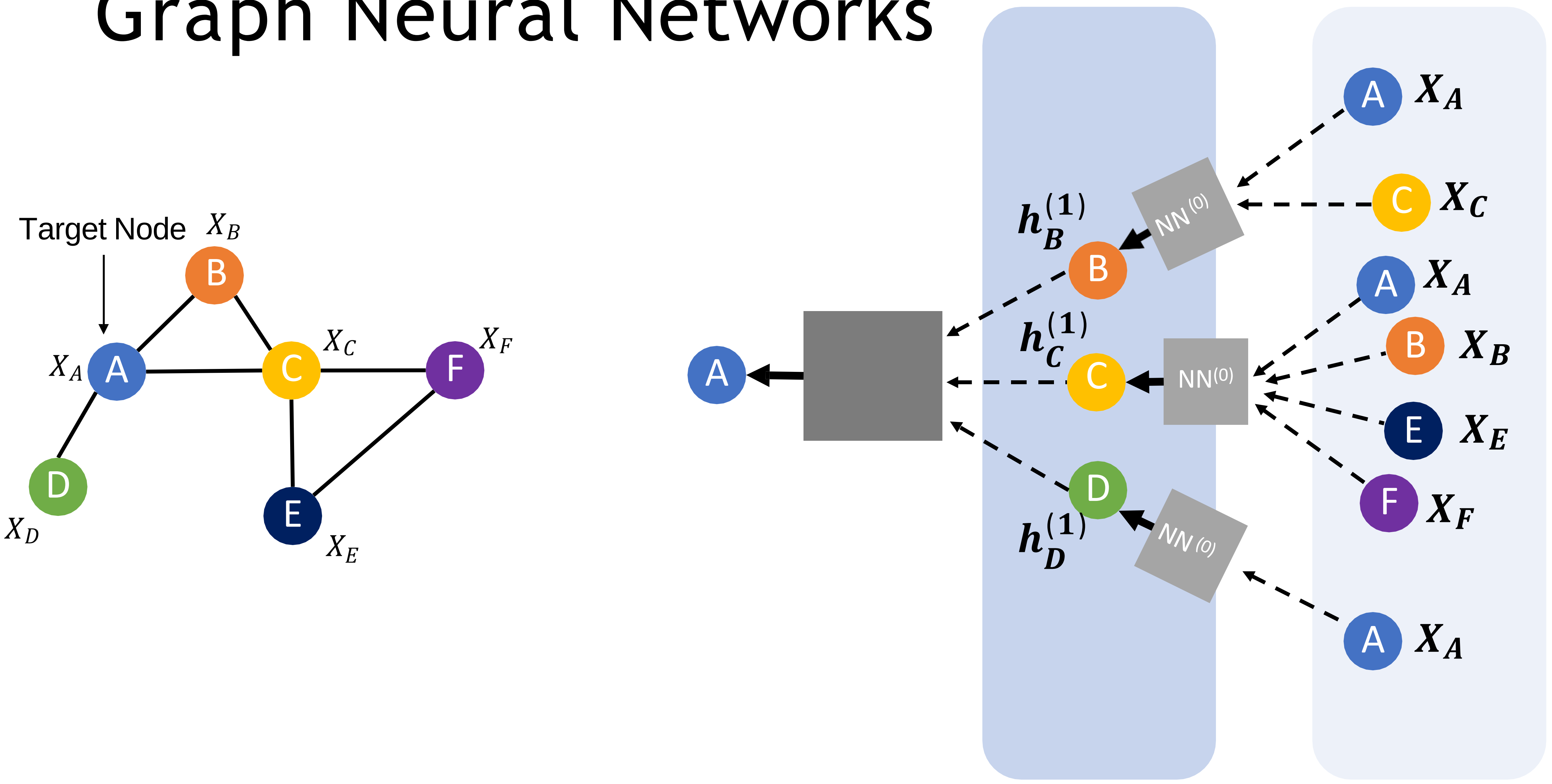




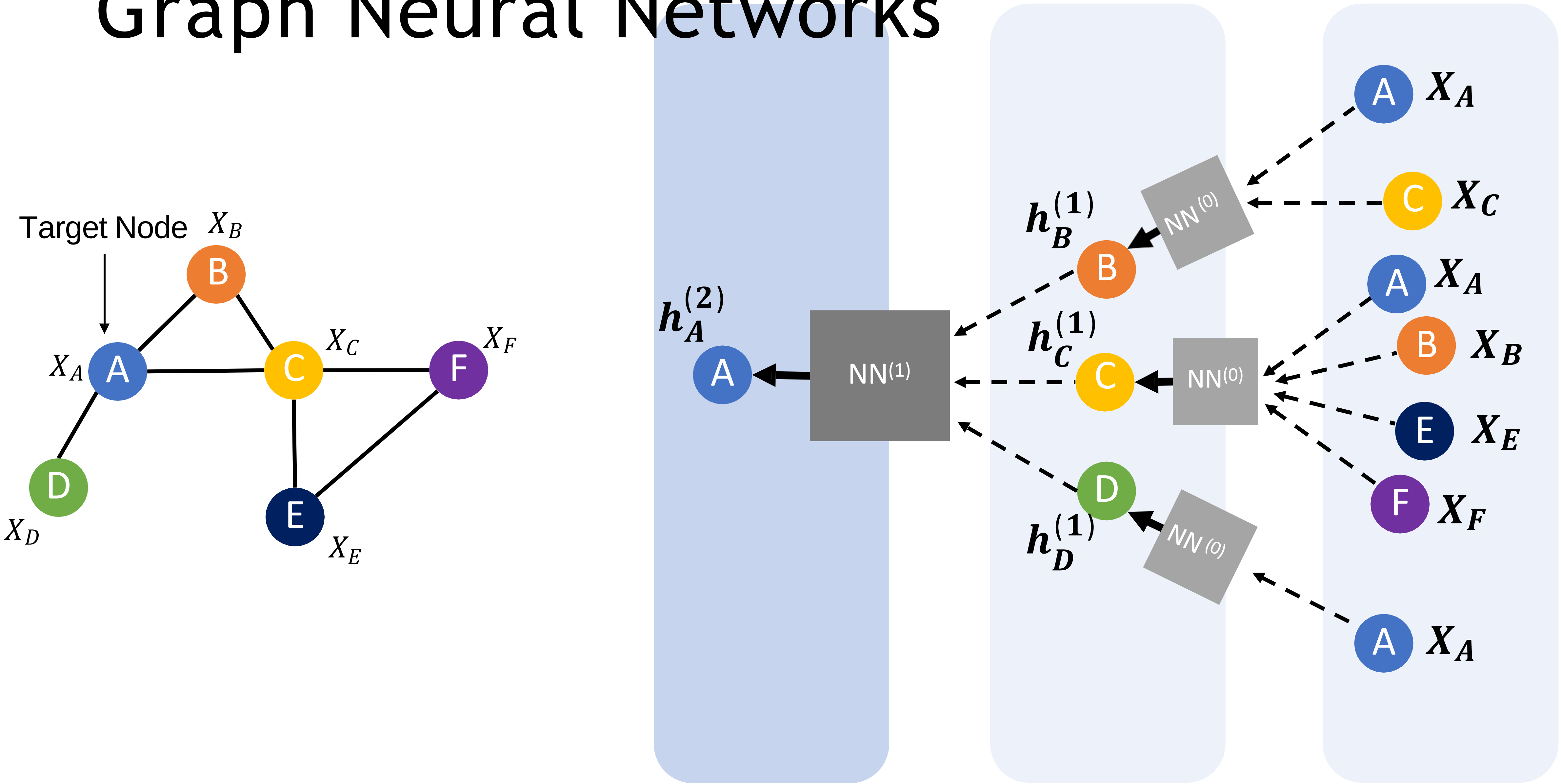
Graph Neural Networks



Graph Neural Networks



Graph Neural Networks

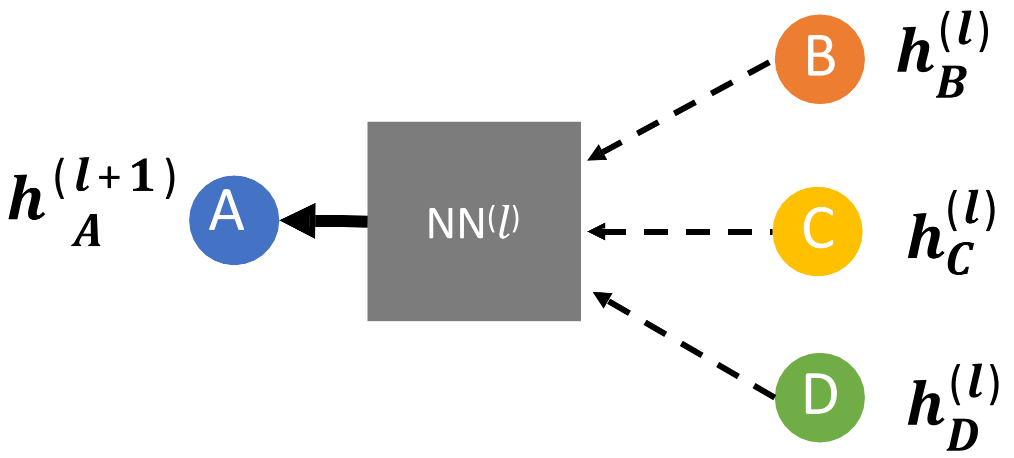


Graph Neural Networks

1. Aggregate messages from neighbors

$h_v^{(l)}$: node embedding of v at l-th layer
$\mathcal{N}(v)$: neighboring nodes of v
$\boldsymbol{f}^{(l)}$: aggregation function at l-th layer
$m_v^{(l)}$: message vector of v at l-th layer

$$
\begin{aligned}
m_A^{(l)} &= \boldsymbol{f}^{(l)}\left(h_A^{(l)}, \left\{h_u^{(l)} : u \in \mathcal{N}(A)\right\}\right) \\
&= \boldsymbol{f}^{(l)}\left(h_A^{(l)}, h_B^{(l)} h_C^{(l)} h_D^{(l)}\right)
\end{aligned}
$$



Neighbors of node A
$\mathcal{N}(A) = \{B, C, D\}$

Graph Neural Networks

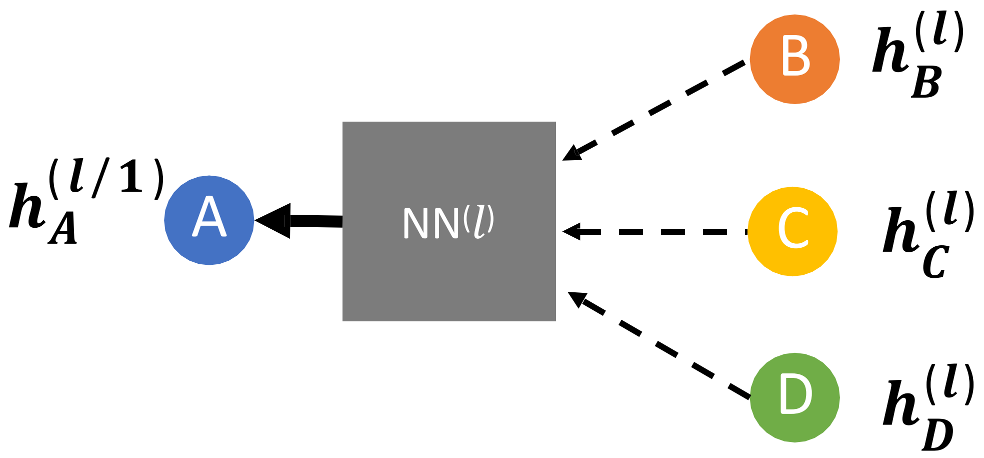
1. **Aggregate messages from neighbors**

$$m_A^{(l)} = \boldsymbol{f}^{(l)}\left(h_A^{(l)}, \left\{h_u^{(l)} : u \in \mathcal{N}(A)\right\}\right)$$
$$= \boldsymbol{f}^{(l)}\left(h_A^{(l)}, h_B^{(l)} h_C^{(l)} h_D^{(l)}\right)$$

2. **Transform** messages

$\boldsymbol{g}^{(l)}$: transformation function at l-th layer

$$h_A^{(l+1)} = \boldsymbol{g}^{(l)}(m_A^{(l)})$$



Neighbors of node A
$\mathcal{N}(A) = \{B, C, D\}$

Graph Neural Networks

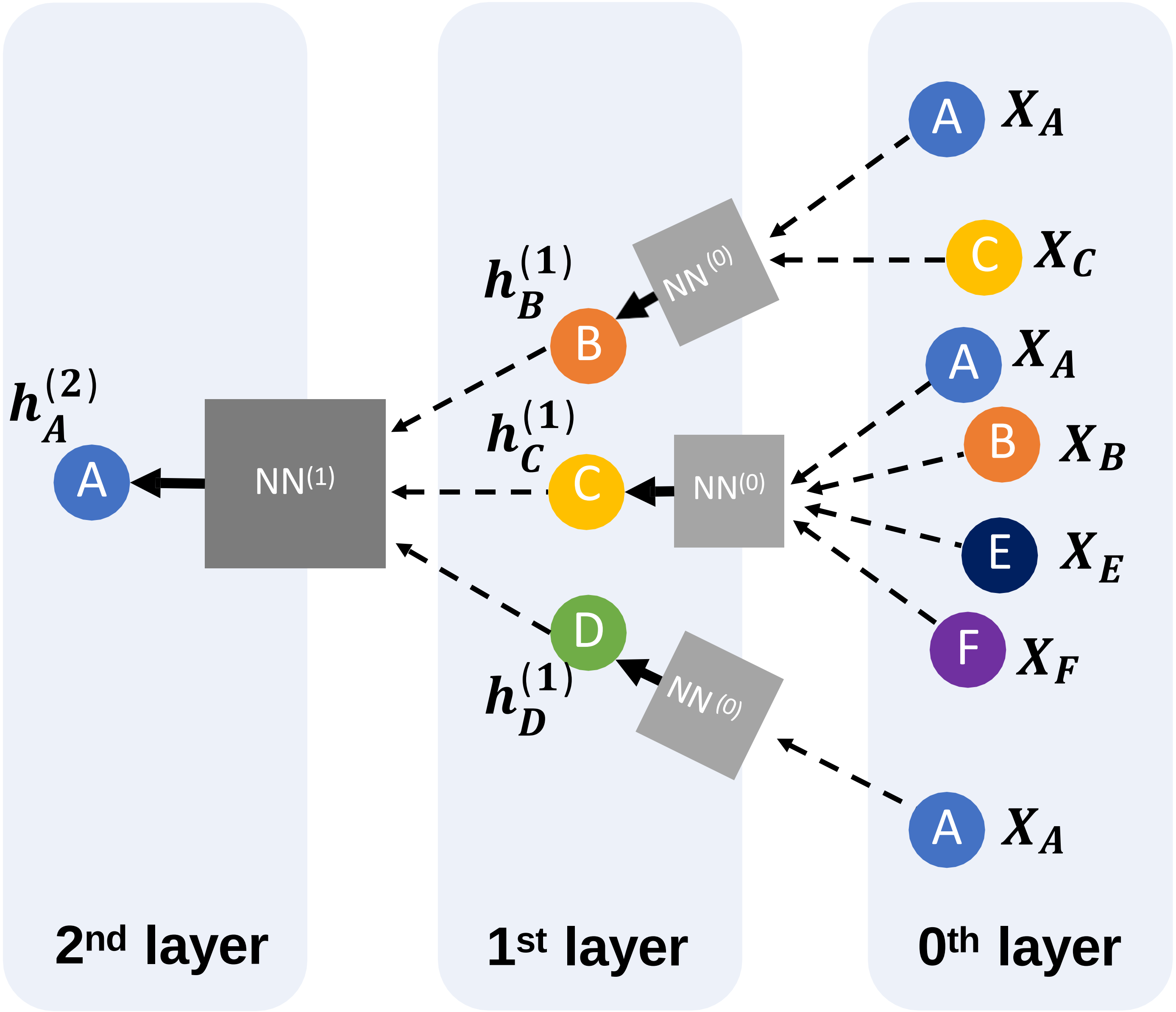
In each layer l ,
for each target node v :

1. Aggregate messages

$$m_v^{(l)} = \boldsymbol{f}^{(l)}\left(h_v^{(l)}, \left\{h_u^{(l)} : u \in \mathcal{N}(v)\right\}\right)$$

2. Transform messages

$$h_v^{(l+1)} = \boldsymbol{g}^{(l)}(m_v^{(l)})$$



Graph Neural Networks

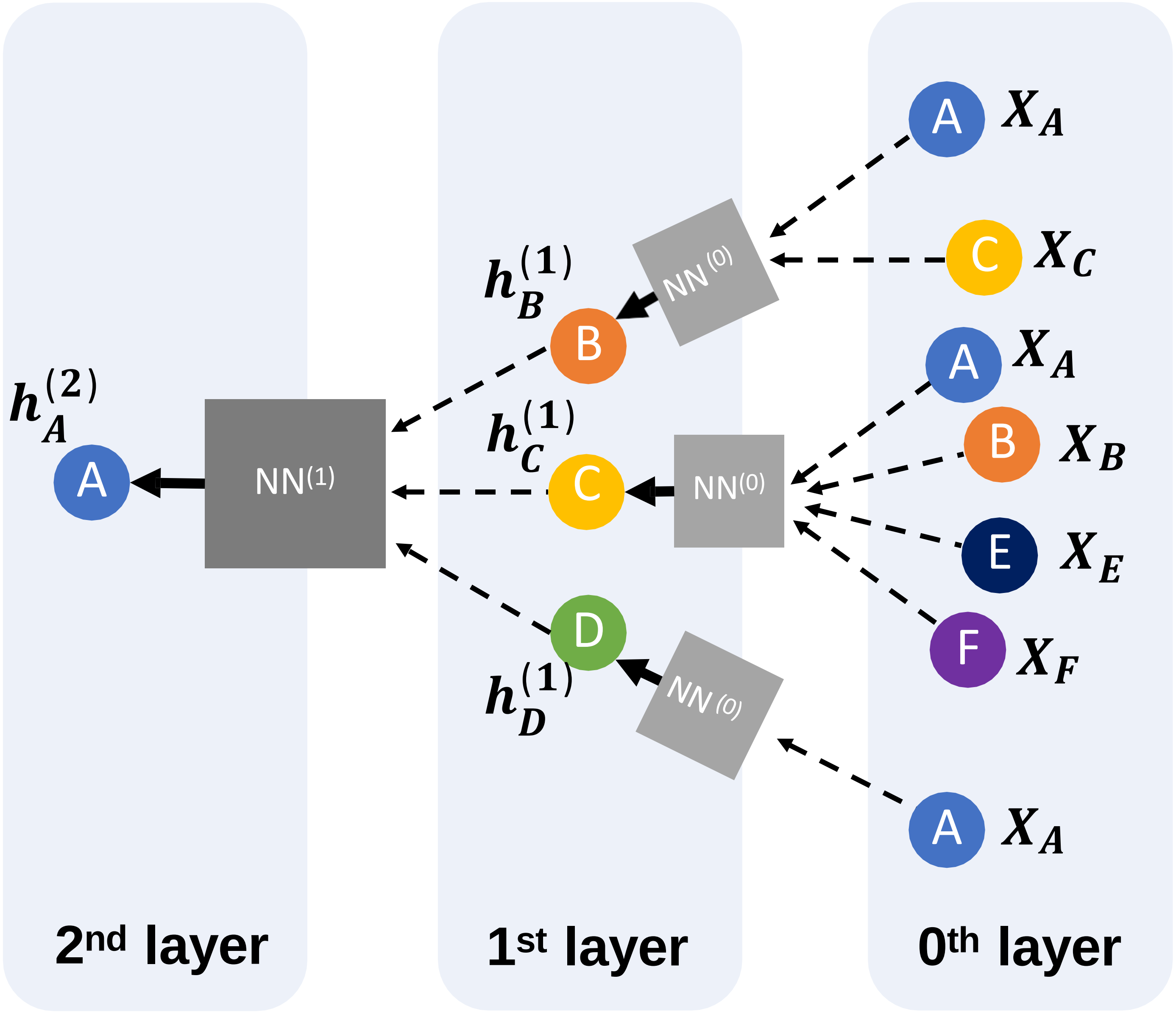
In each layer l ,
for each target node v :

1. Aggregate messages

$$m_v^{(l)} = \boldsymbol{f}^{(l)}\left(h_v^{(l)}, \left\{h_u^{(l)} : u \in \mathcal{N}(v)\right\}\right)$$

2. Transform messages

$$h_v^{(l+1)} = \boldsymbol{g}^{(l)}(m_v^{(l)})$$



Graph Neural Networks

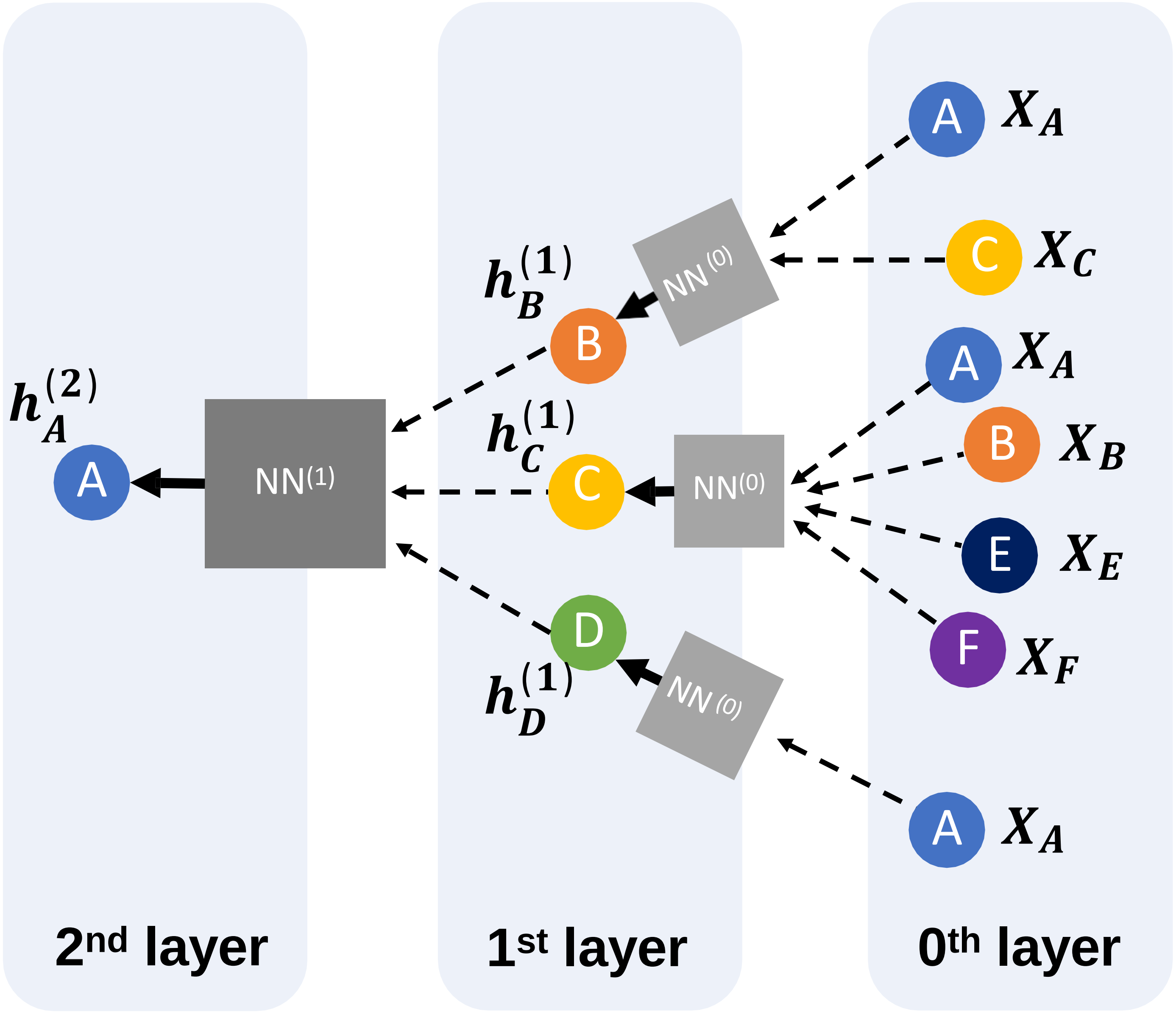
Graph Convolutional Networks[1]

1. Aggregate messages

$$m_v^{(l)} = \frac{1}{|\mathcal{N}(v) + 1|} \sum_{u \in \mathcal{N}(v) \cup \{v\}} h_u^{(l)}$$

2. Transform messages

$$h_v^{(l+1)} = \sigma(\boldsymbol{W}^{(l)} \circ m_v^{(l)})$$



[1] Kipf, Thomas N., et al. "Semi-supervised classification with graph convolutional networks."

Graph Neural Networks

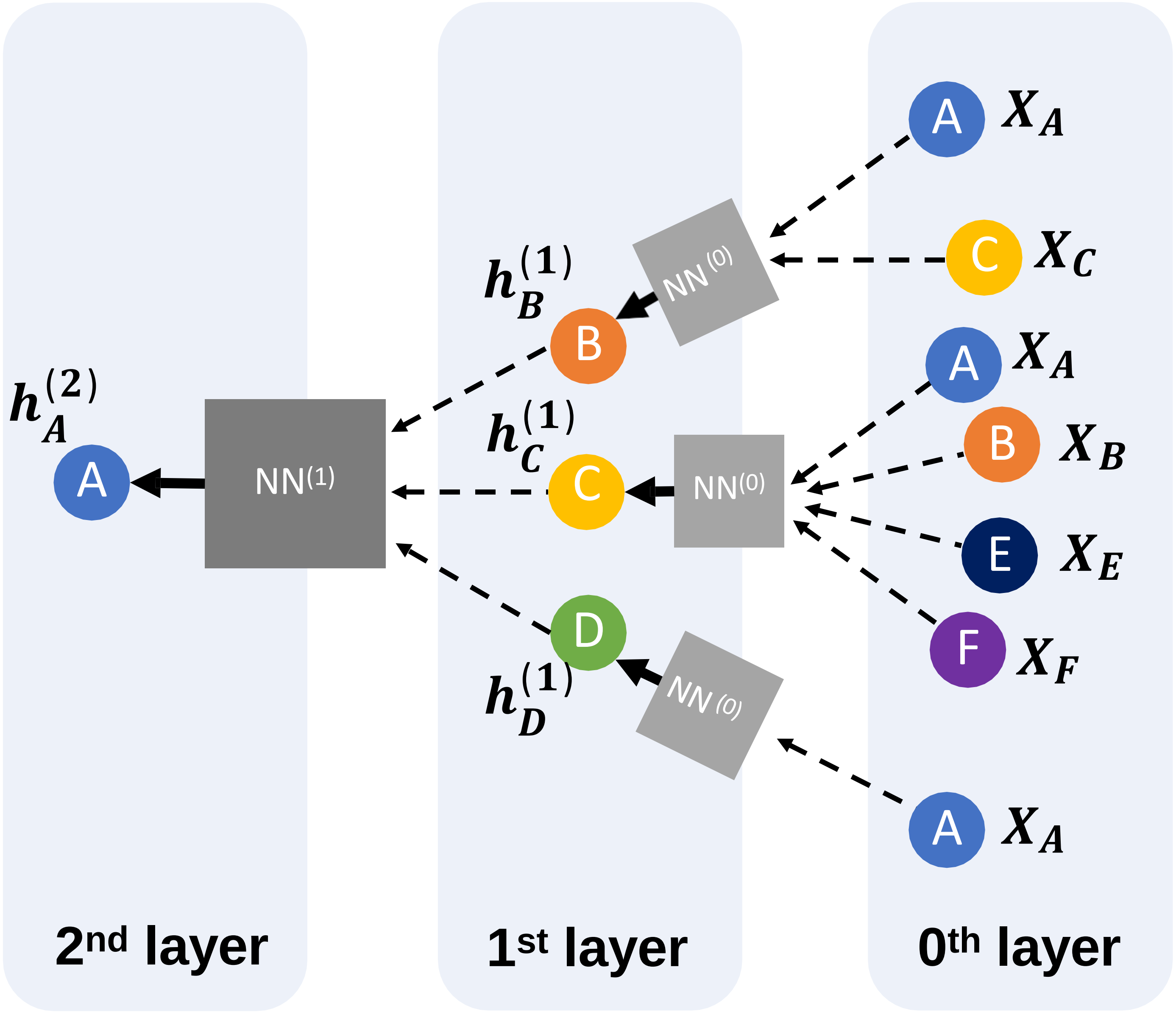
Graph Isomorphism Networks[2]

1. Aggregate messages

$$m_v^{(l)} = \sum_{u \in \mathcal{N}(v) \cup \{v\}} h_u^{(l)}$$

2. Transform messages

$$h_v^{(l+1)} = \sigma(\boldsymbol{W}^{(l)} \circ m_v^{(l)})$$



[2] Xu, Keyulu, et al. "How powerful are graph neural networks?."

Graph Neural Networks

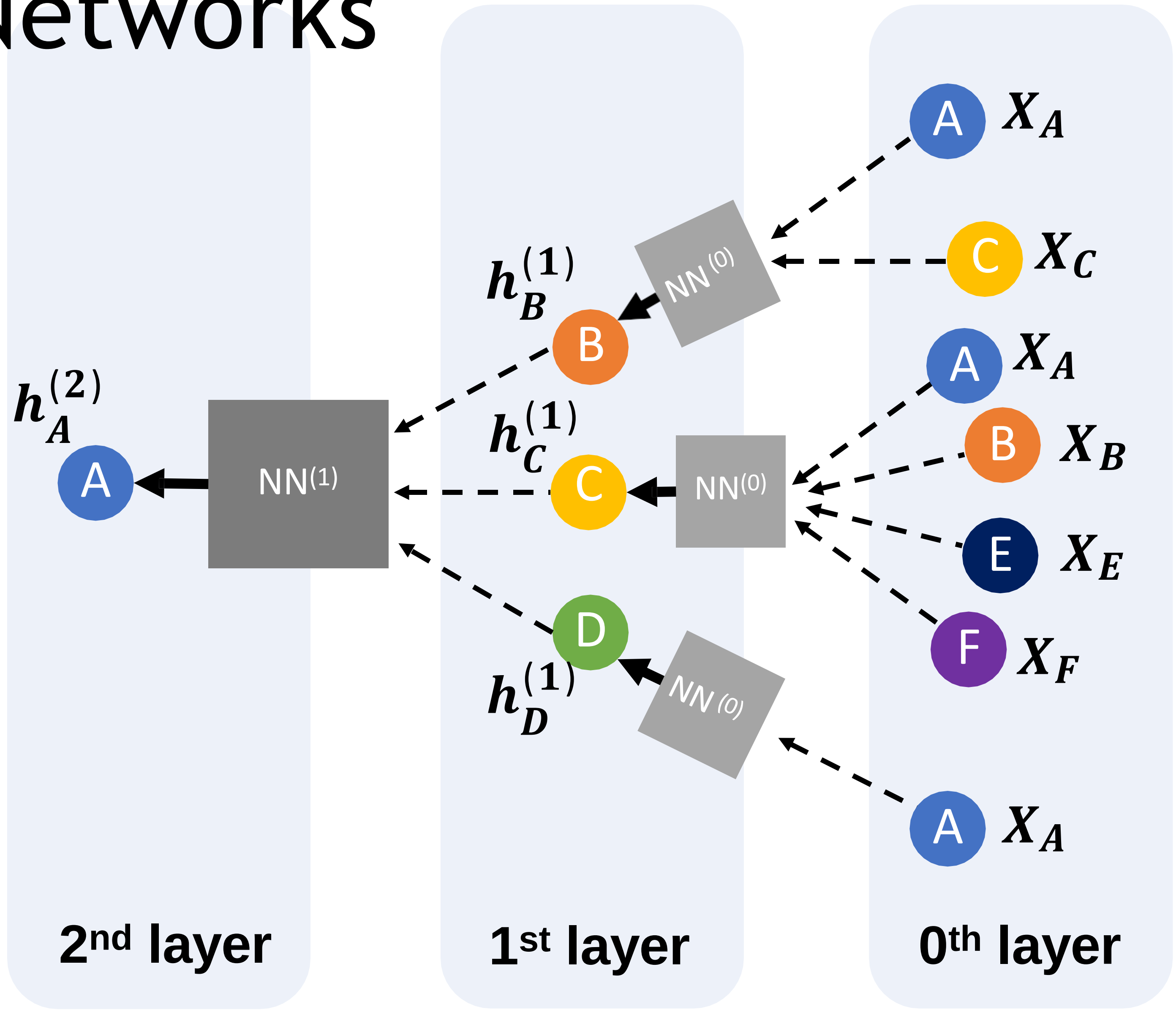
Simplified GCN[3]

1. Aggregate messages

$$m_v^{(l)} = \frac{1}{|\mathcal{N}(v) + 1|} \sum_{u \in \mathcal{N}(v) \cup \{v\}} h_u^{(l)}$$

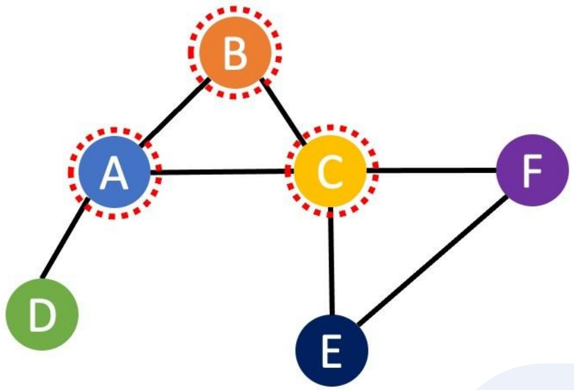
2. Transform messages

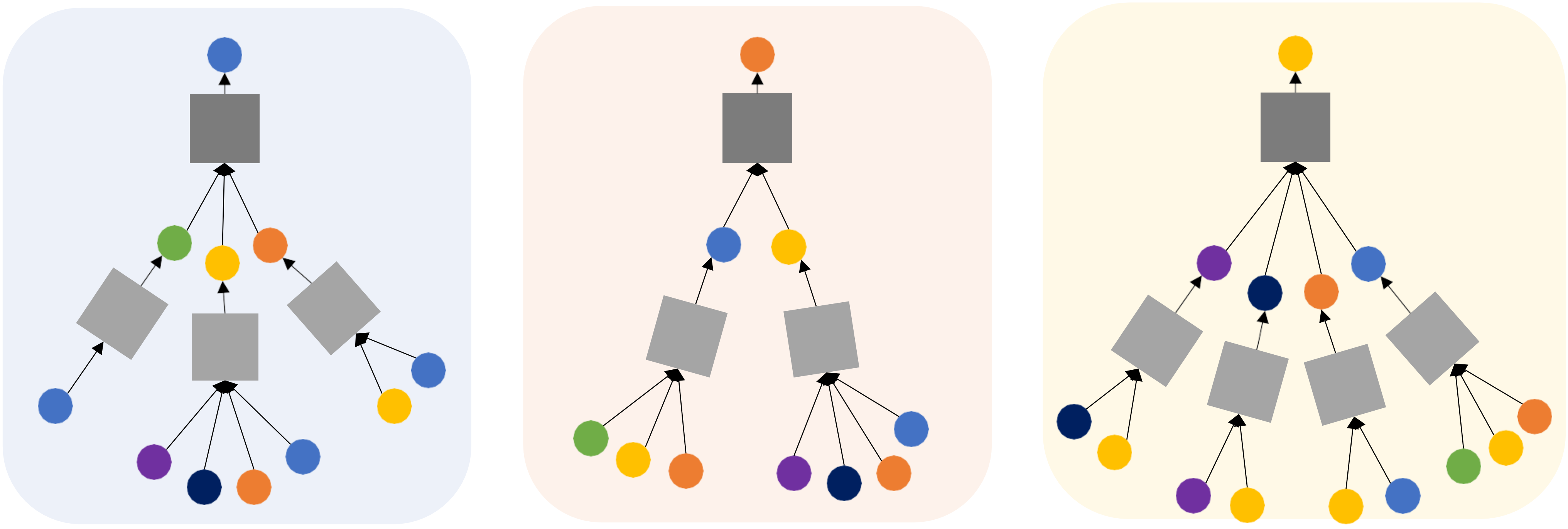
$$h_v^{(l+1)} = \boldsymbol{W}^{(l)} \circ m_v^{(l)}$$



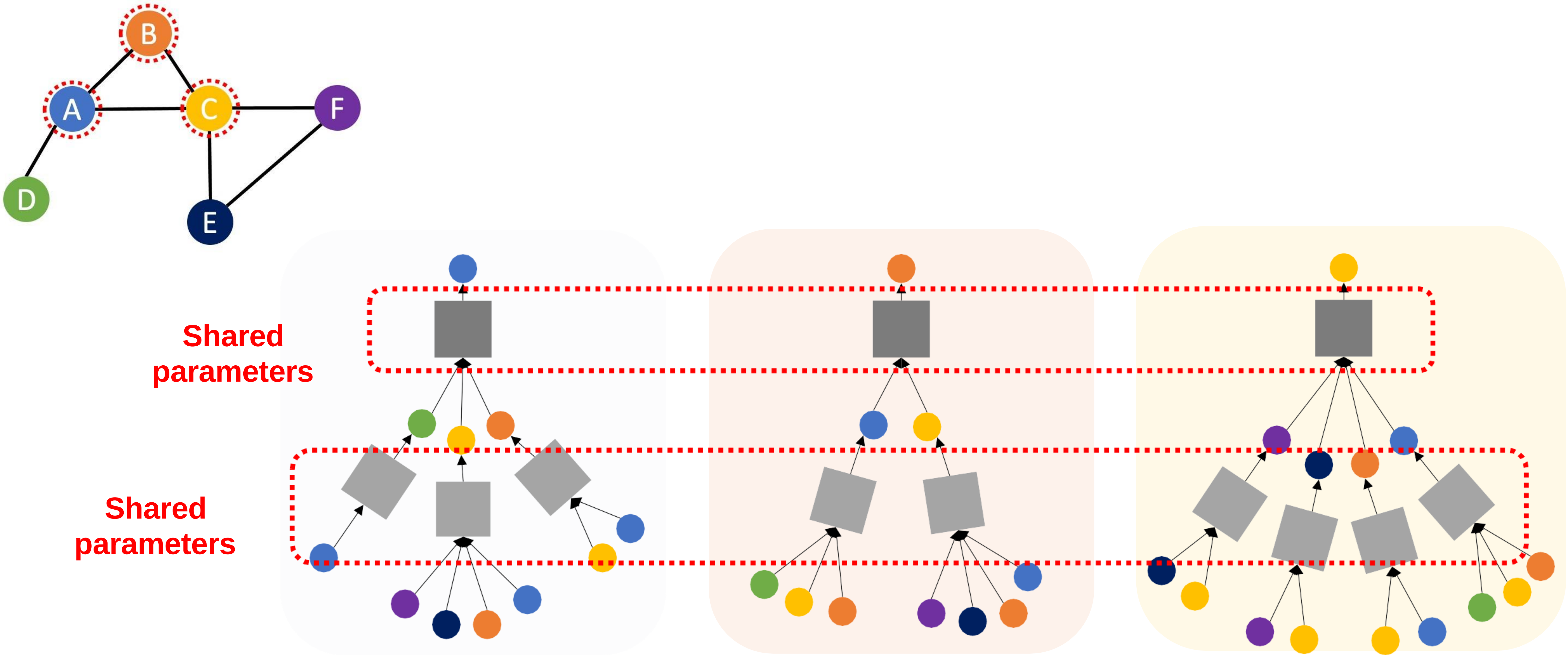
[3] Wu, Felix, et al. "Simplifying graph convolutional networks."

Computation graphs





Computation graphs



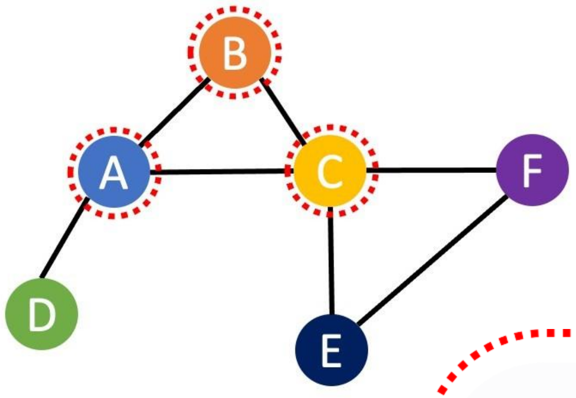
Parallel Computing

$$h_v^{(l)} = \sigma(\boldsymbol{W}^{(l)} \circ (\frac{1}{|\mathcal{N}(v)+1|} \sum_{u \in \mathcal{N}(v) \cup \{v\}} h_u^{(l-1)}))$$

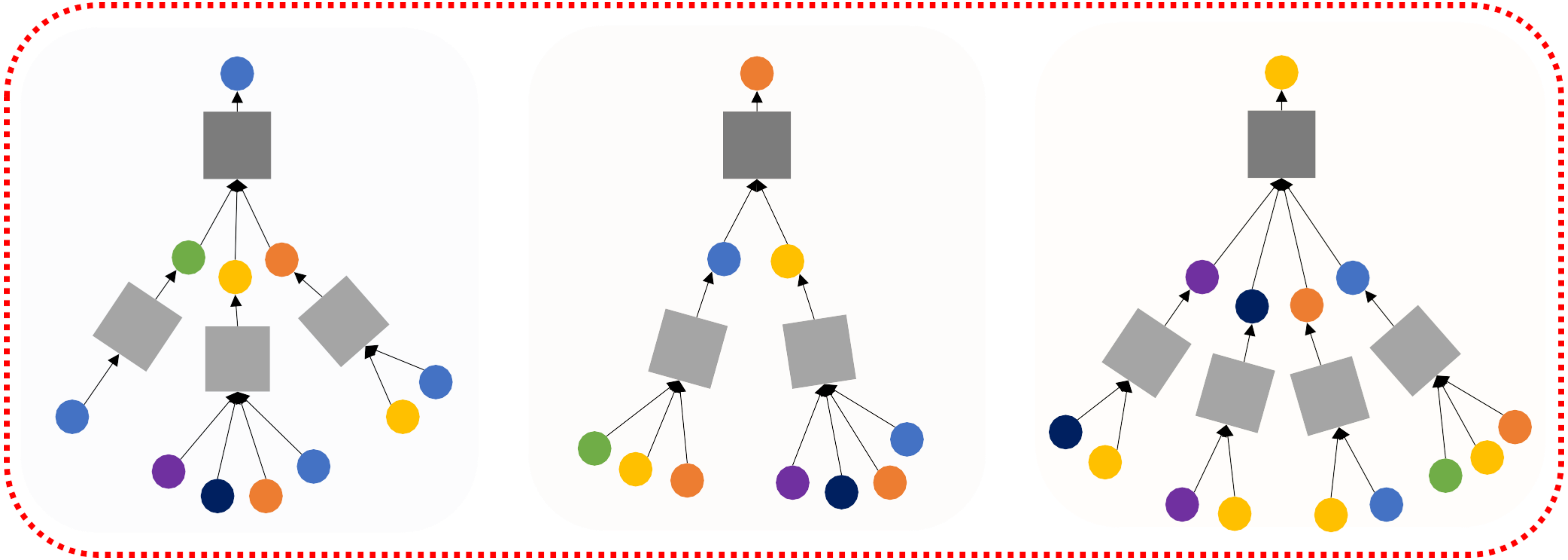
$$\boldsymbol{H}^{(l)} = \sigma(\boxed{\widetilde{(\boldsymbol{A}+\boldsymbol{I})}}\boldsymbol{H}^{(l-1)}\boxed{\boldsymbol{W}^{(l)}})$$

Fixed

Trainable

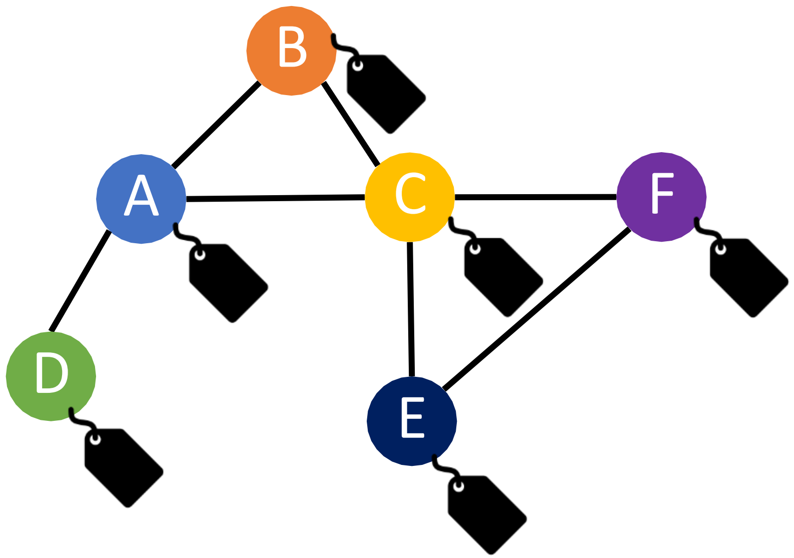


Parallel



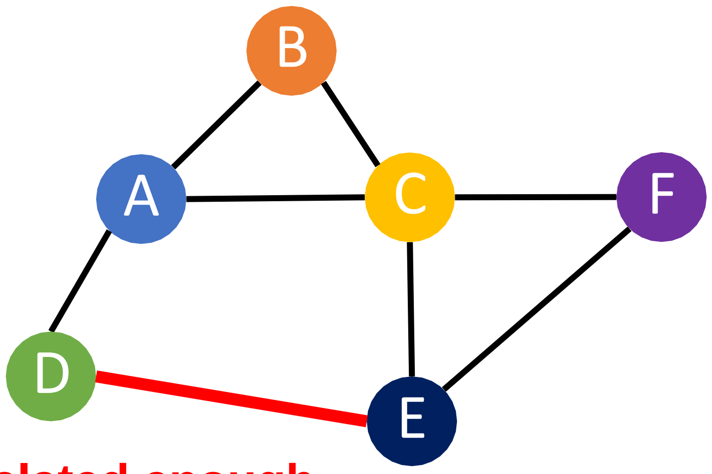
Downstream tasks

- Node-level prediction



Downstream tasks

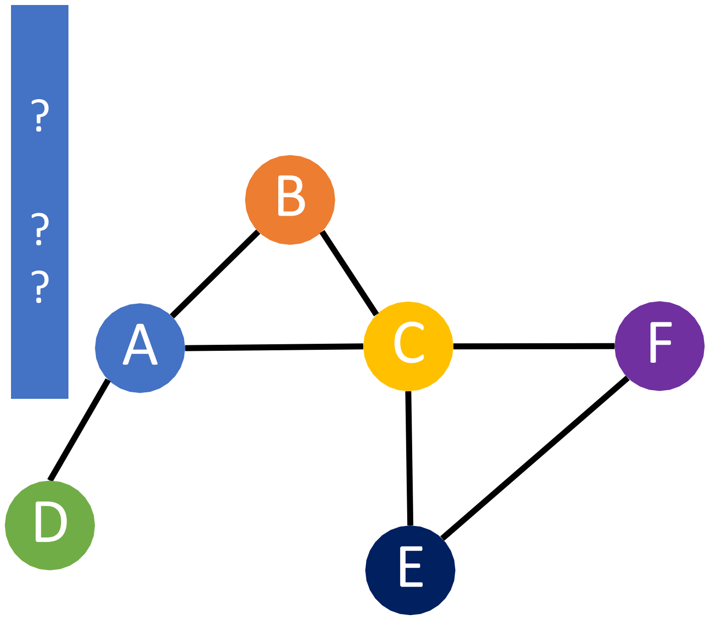
- Node-level prediction
- Edge-level prediction



D and E are related enough to be connected?

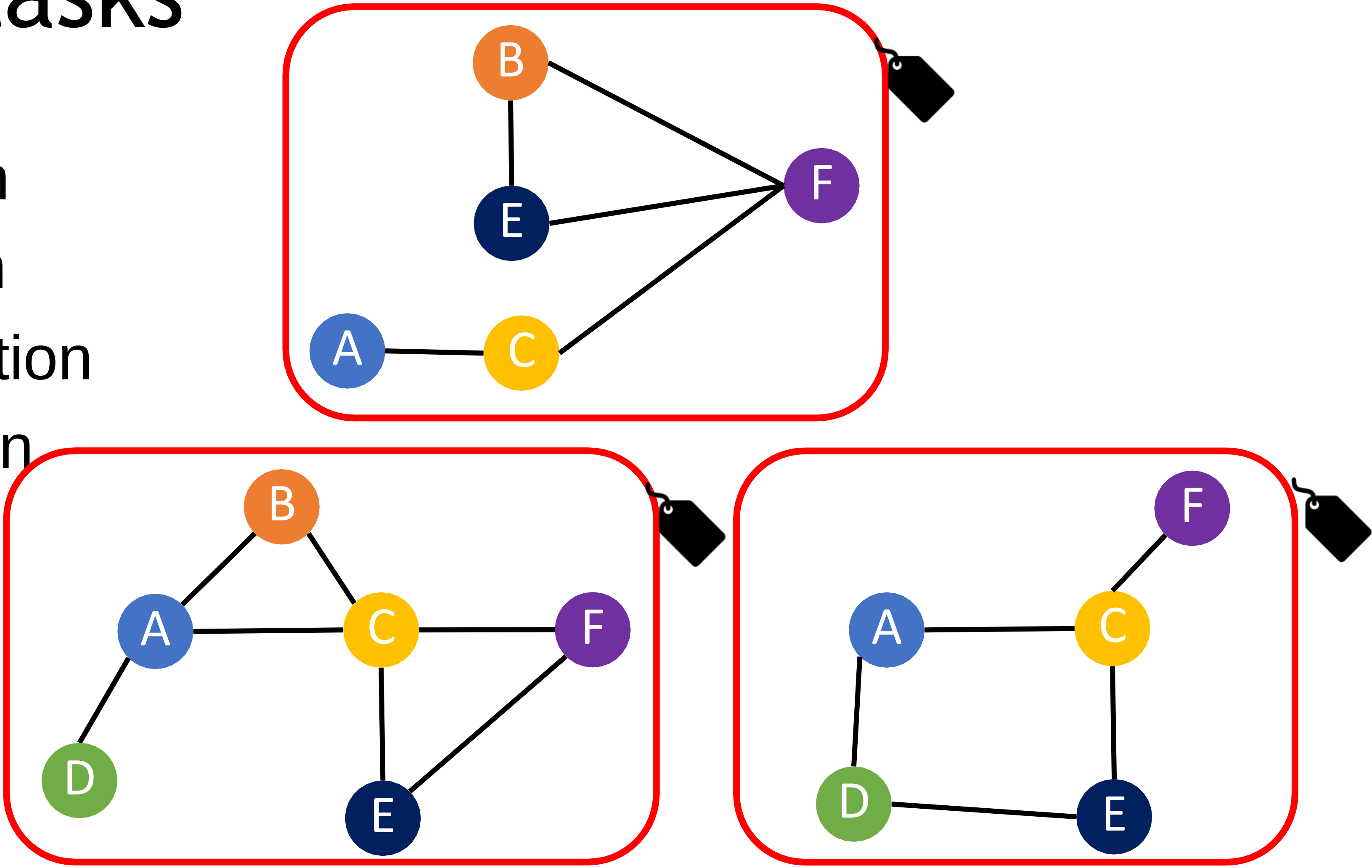
Downstream tasks

- Node-level prediction
- Edge-level prediction
- Attribute-level prediction



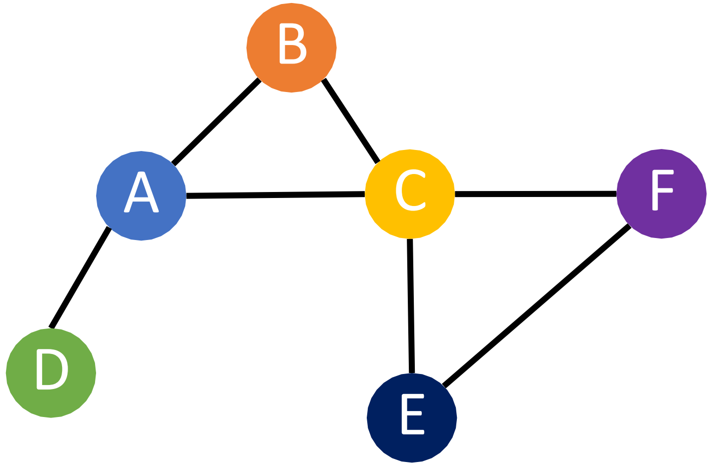
Downstream tasks

- Node-level prediction
- Edge-level prediction
- Attribute-level prediction
- Graph-level prediction

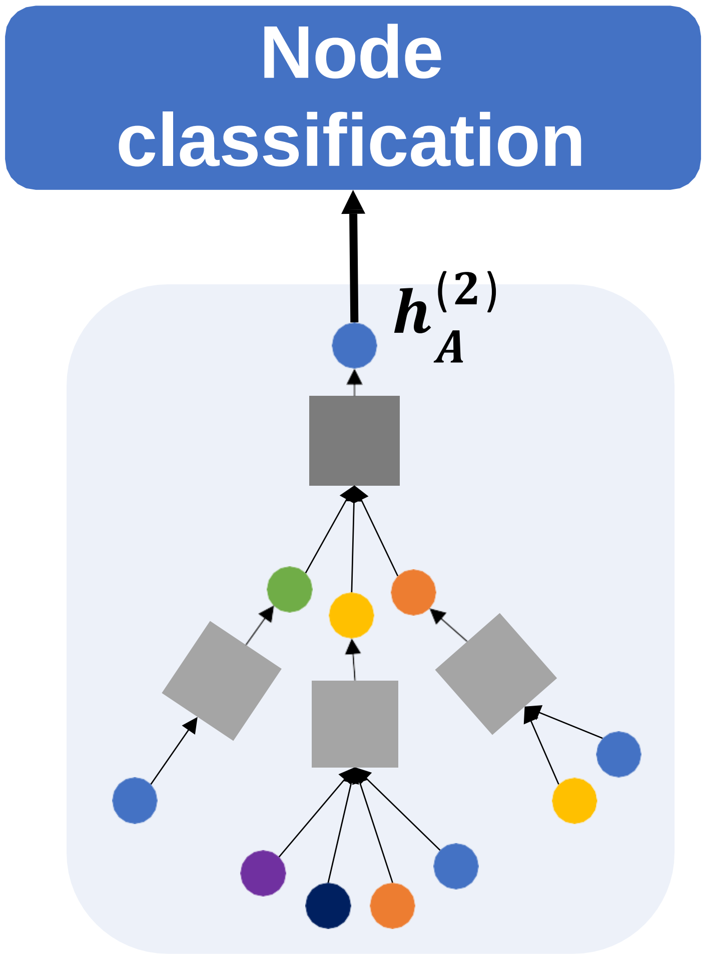


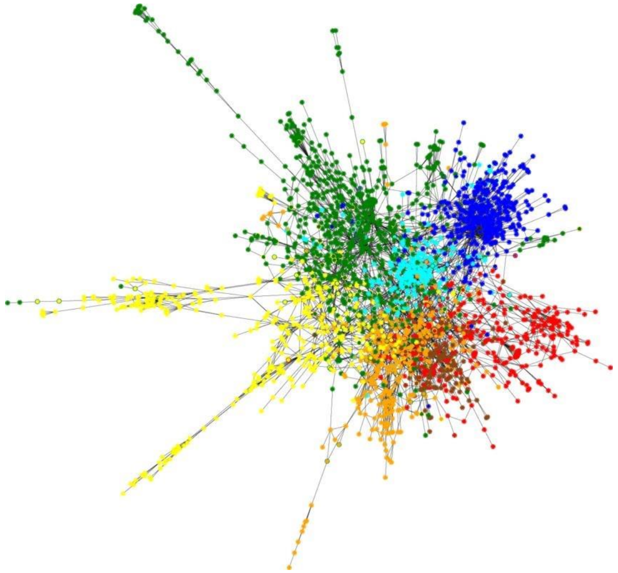
Downstream tasks

- **Node-level prediction**
- Edge-level prediction
- Attribute-level prediction
- **Graph-level prediction**



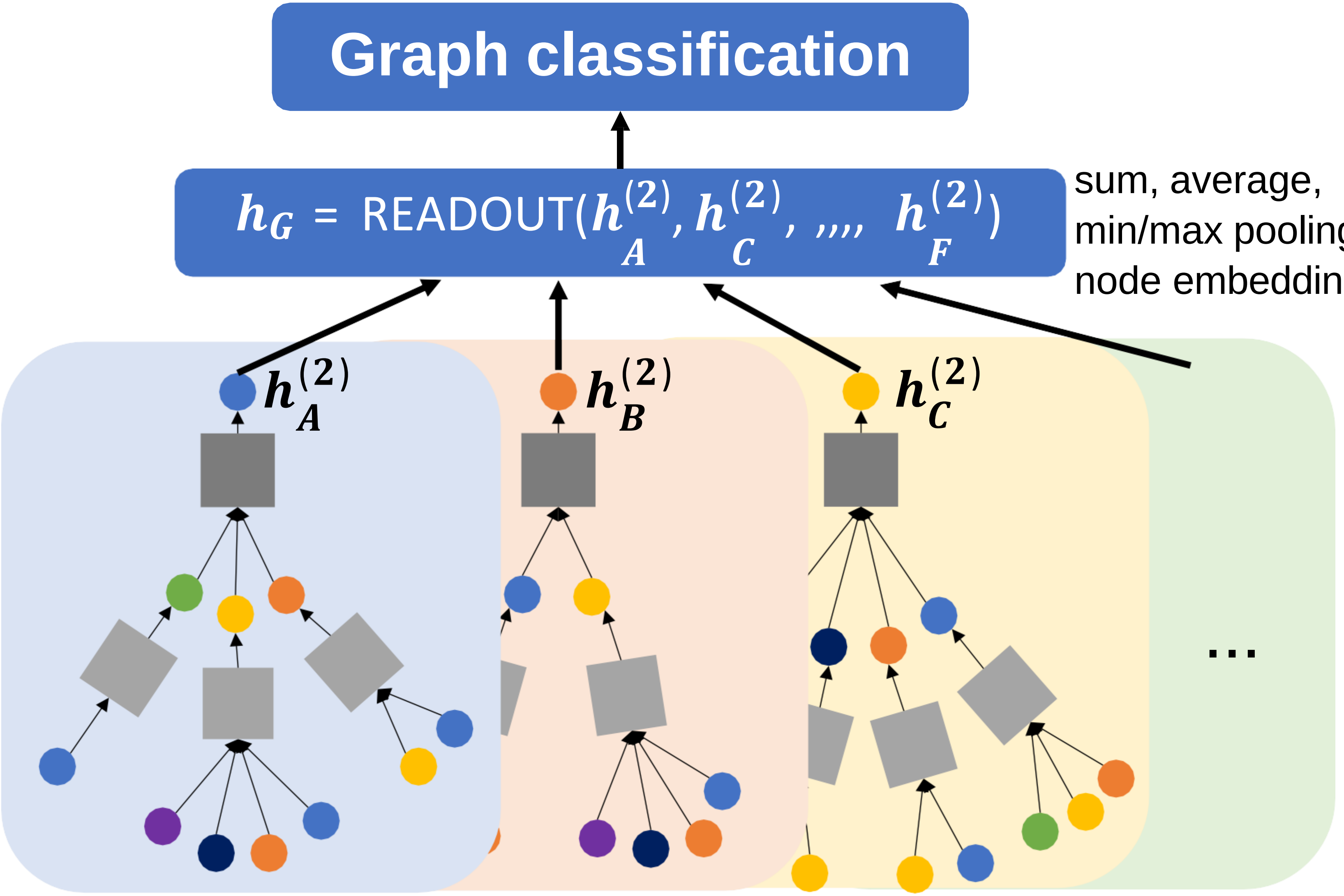
Node-level prediction tasks



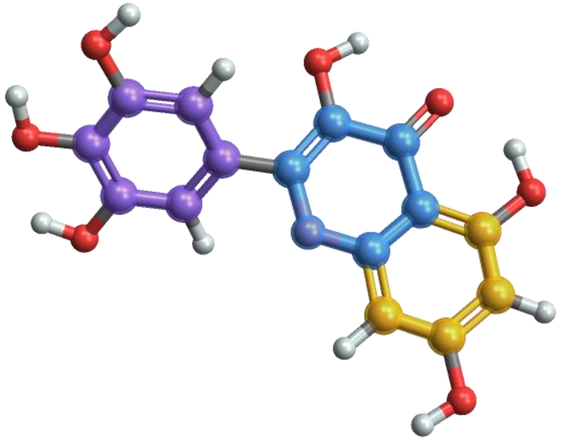


- Classify **papers** into topics on **citation networks**
- Cluster **posts** into subgroups on **Reddit networks**
- Classify **products** into categories on **Amazon co-purchase graphs**

Graph-level prediction tasks



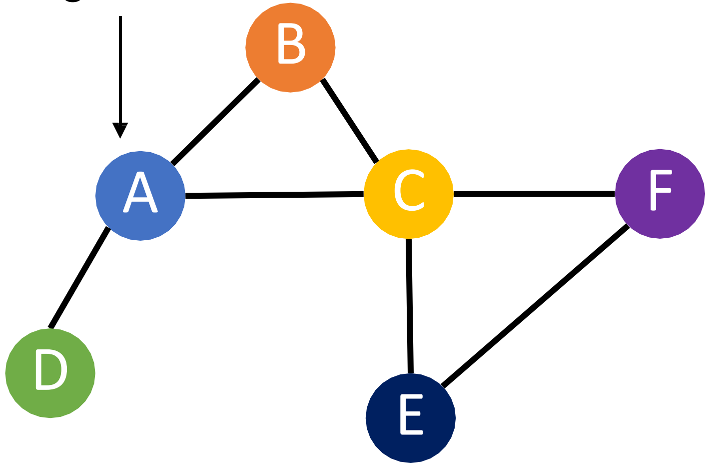
sum, average, min/max pooling of node embeddings

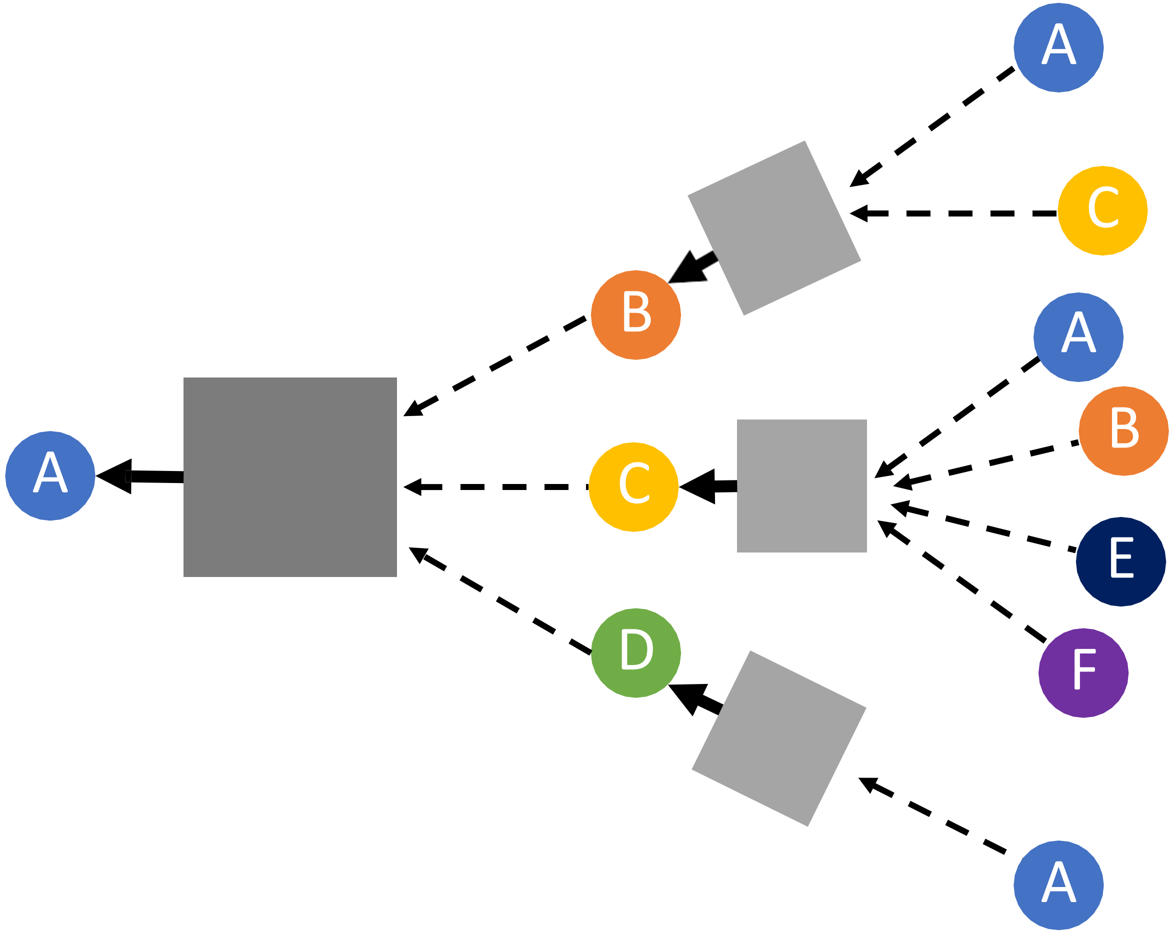


- Predict **properties of a molecule (graph)** where nodes are atoms and edges are chemical bonds

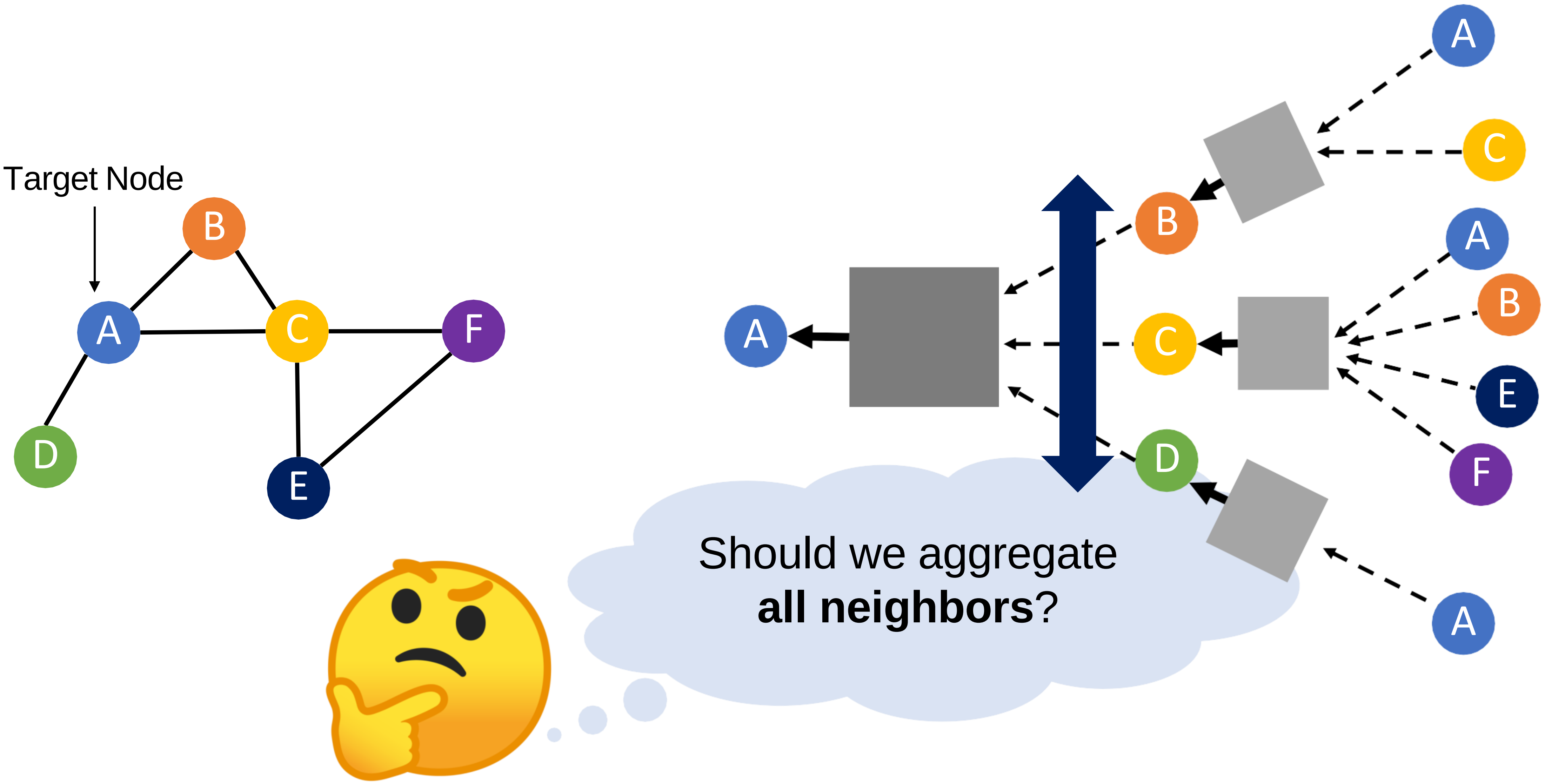
Graph Neural Networks

Target Node

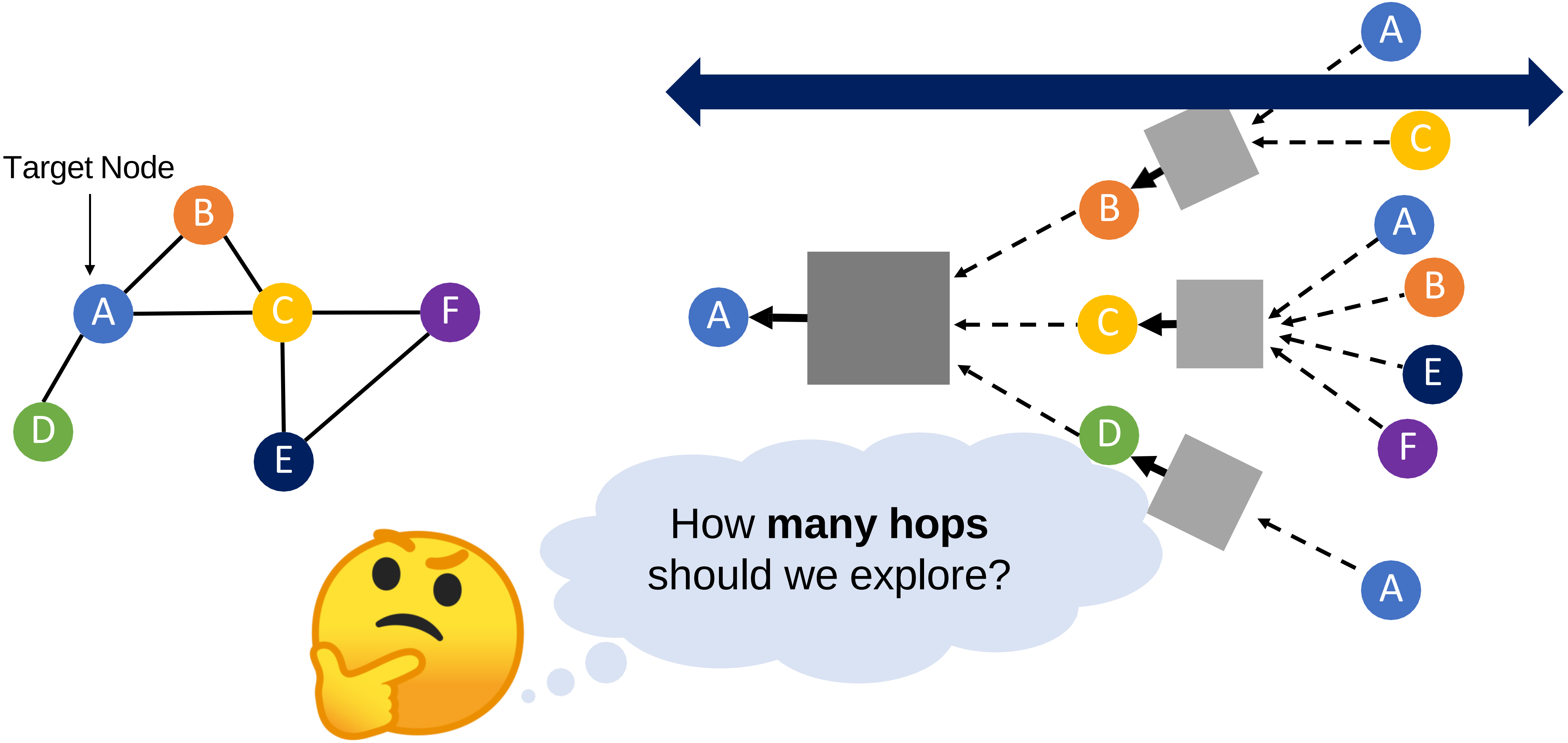




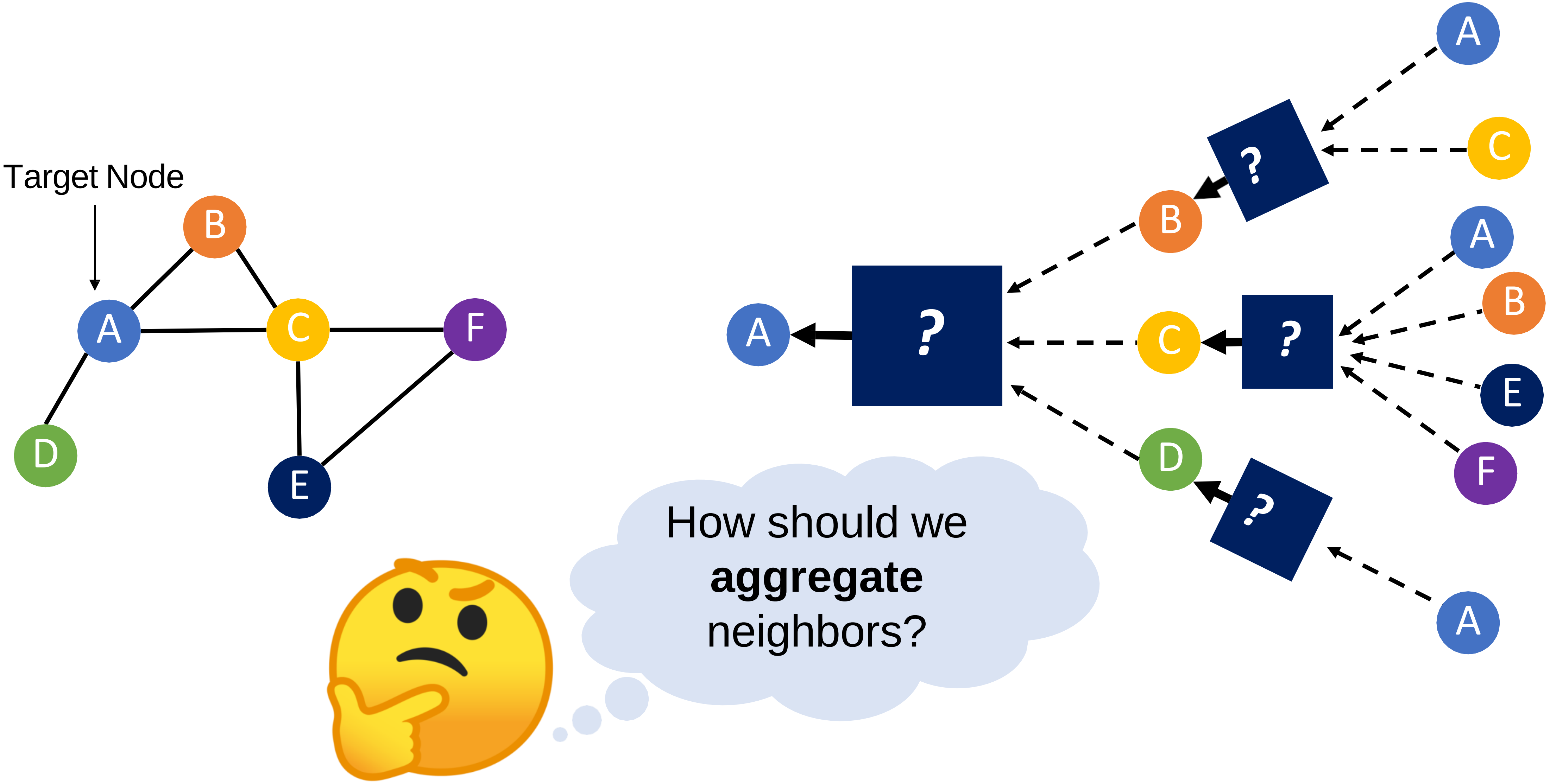
Graph Neural Networks - Width



Graph Neural Networks - Depth

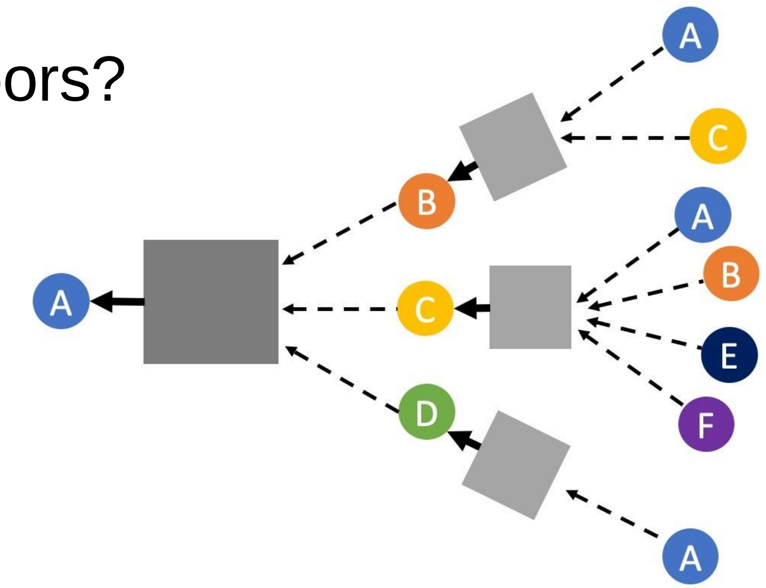


Graph Neural Networks - Aggregation



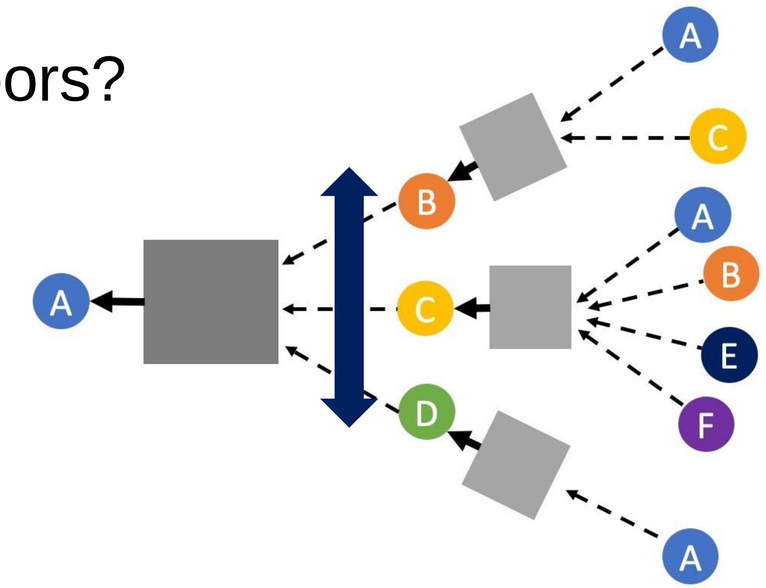
Graph Neural Network Architectures

- Width
  - Which neighbors should we aggregate messages from?
- Depth
  - How many hops should we check?
- Aggregation
  - How should we aggregate messages from neighbors?



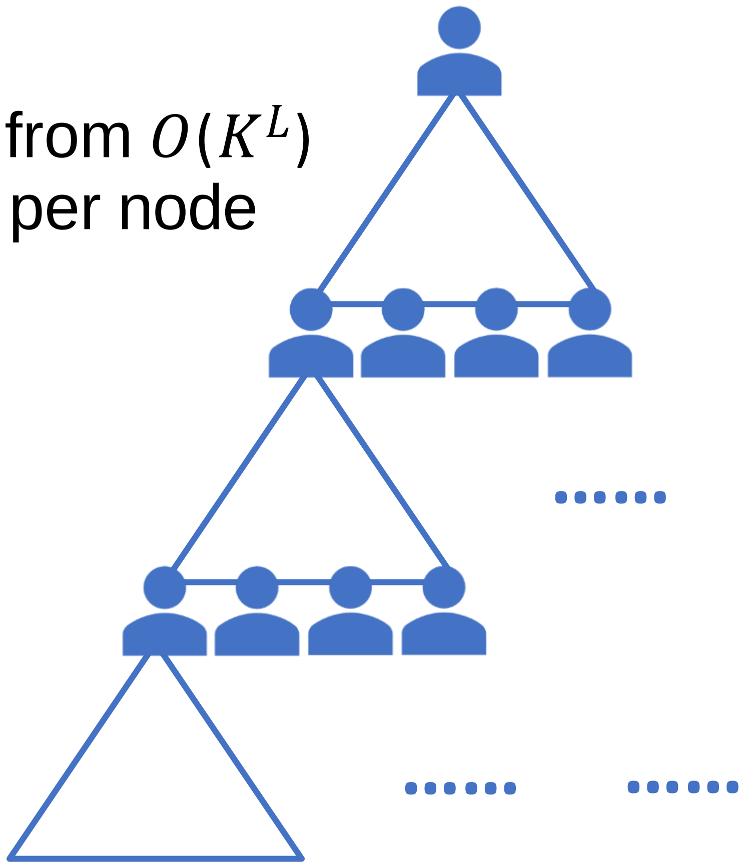
Graph Neural Network Architectures

- **Width**
  - **Which neighbors should we aggregate messages from?**
- Depth
  - How many hops should we check?
- Aggregation
  - How should we aggregate messages from neighbors?



Aggregation Width in GNNs

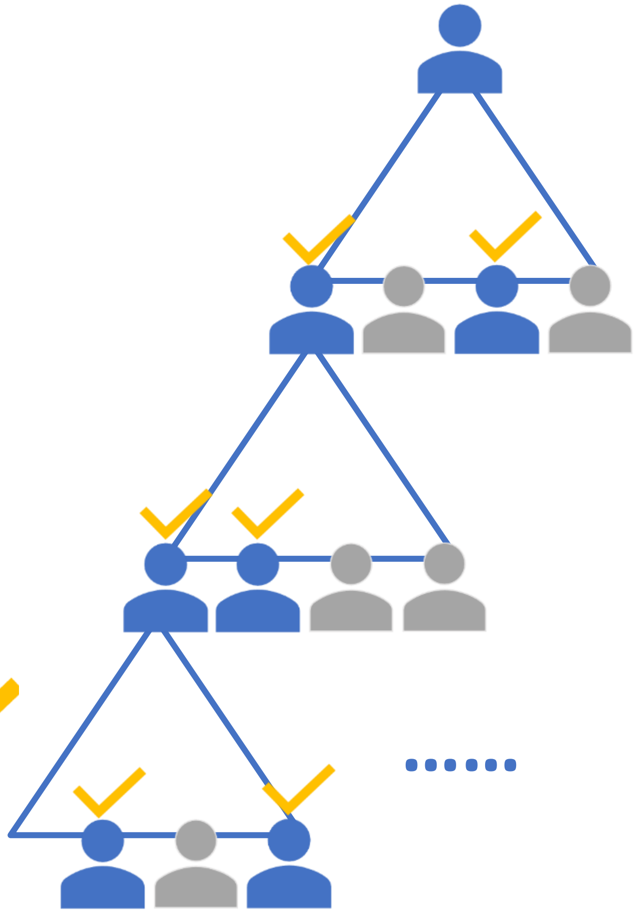
- If we aggregate all neighbors, GNNs have scalability issues
- Neighbor explosion
  - In L-layer GNNs, one node aggregates information from $O(K^L)$ nodes where K is the average number of neighbors per node



Aggregation Width in GNNs

- Limit the neighborhood expansion by **sampling** a fixed number of neighbors

Sample the neighbors ✓



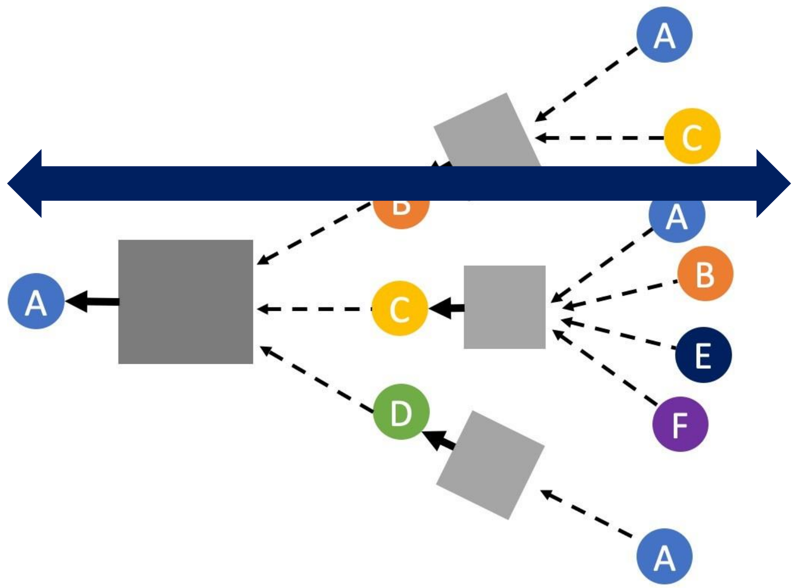
Aggregation Width in GNNs

- Random sampling
  - Assign **same** sampling probabilities to all neighbors
  - *GraphSage*[4]
- Importance sampling
  - Assign **different** sampling probabilities to all neighbors
  - *higher sampling probabilities to neighbors* who
    - **Minimize variance in sampling**
      - *FastGCN*[5], *LADIES*[6], *AS-GCN*[7], *GCN-BS*[8]
    - **Maximize GNN performance**
      - *PASS*[9]

4 Will Hamilton, et al. "Inductive representation learning on large graphs"
5 Jie Chen, et al. "Fastgcn: fast learning with graph convolutional networks via importance sampling"
6 Difan Zou, et al. "Layer-Dependent Importance Sampling for Training Deep and Large Graph Convolutional Networks"
7 Wenbing Huang, et al. "Adaptive sampling towards fast graph representation learning"
8 Ziqi Liu, et al. "Bandit Samplers for Training Graph Neural Networks"
9 Minji Yoon, et al. "Performance-Adaptive Sampling Strategy Towards Fast and Accurate Graph Neural Networks"

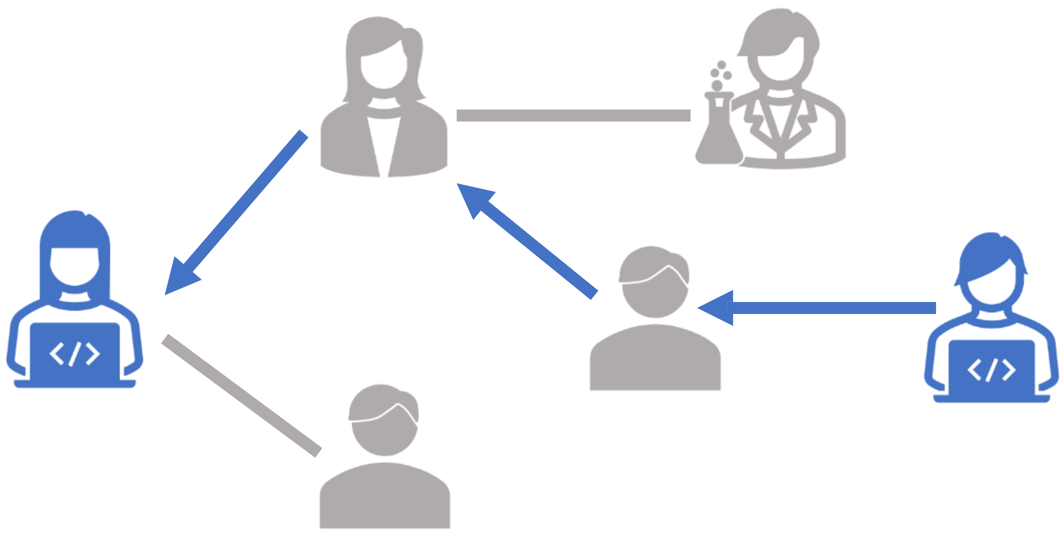
Graph Neural Network Architectures

- Width
  - Which neighbors should we aggregate messages from?
- **Depth**
  - **How many hops should we check?**
- Aggregation
  - How should we aggregate messages from neighbors?



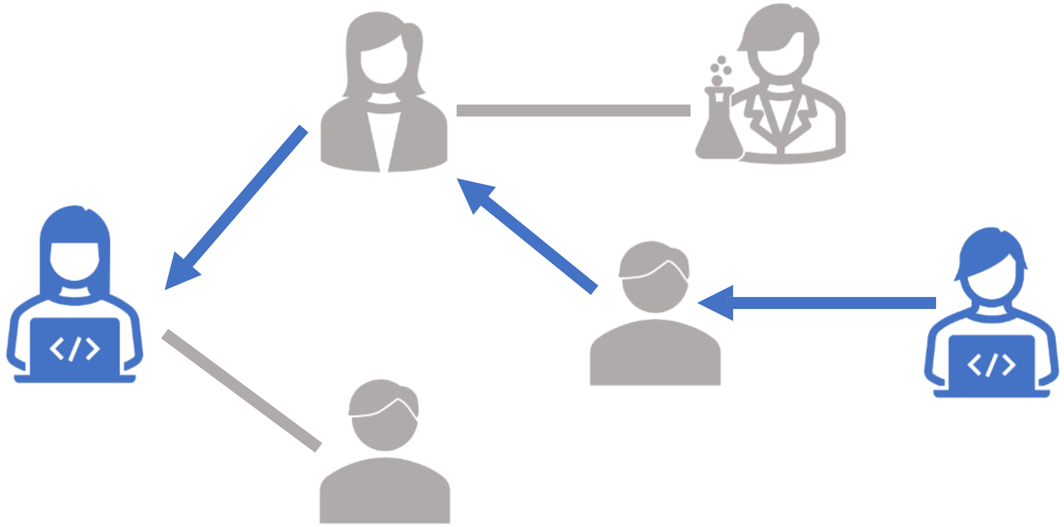
Aggregation Depth in GNNs

- Informative neighbors could be indirectly connected with a target node



Aggregation Depth in GNNs

- Informative neighbors could be indirectly connected with a target node
- Can’t we just look multiple hops away from the target node?



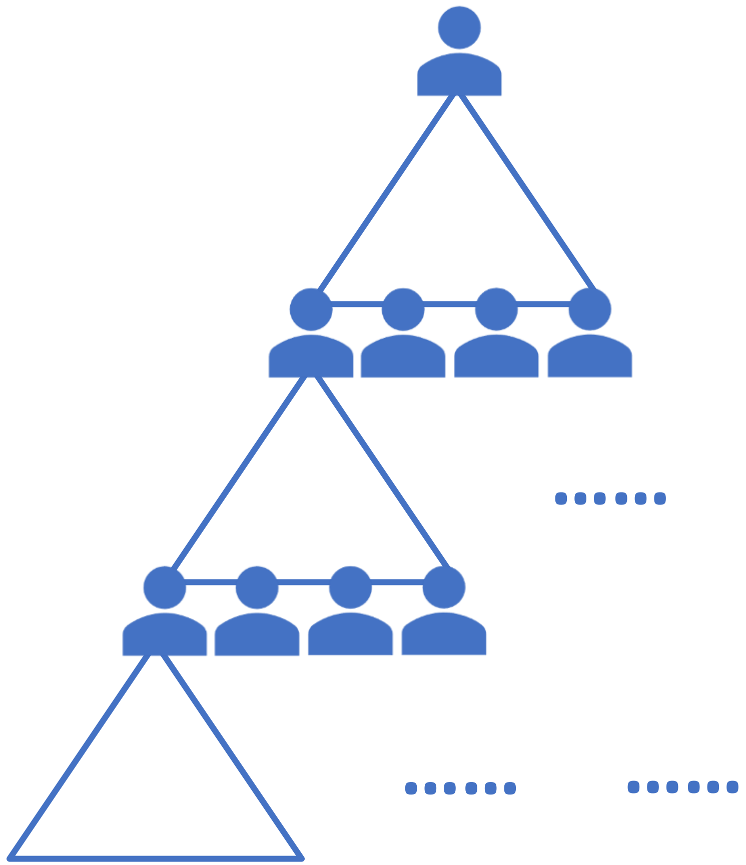
Aggregation Depth in GNNs

- 2-layer or 3-layer GNNs are commonly used in real worlds

Wasn't it Deeeep Learning?

Aggregation Depth in GNNs

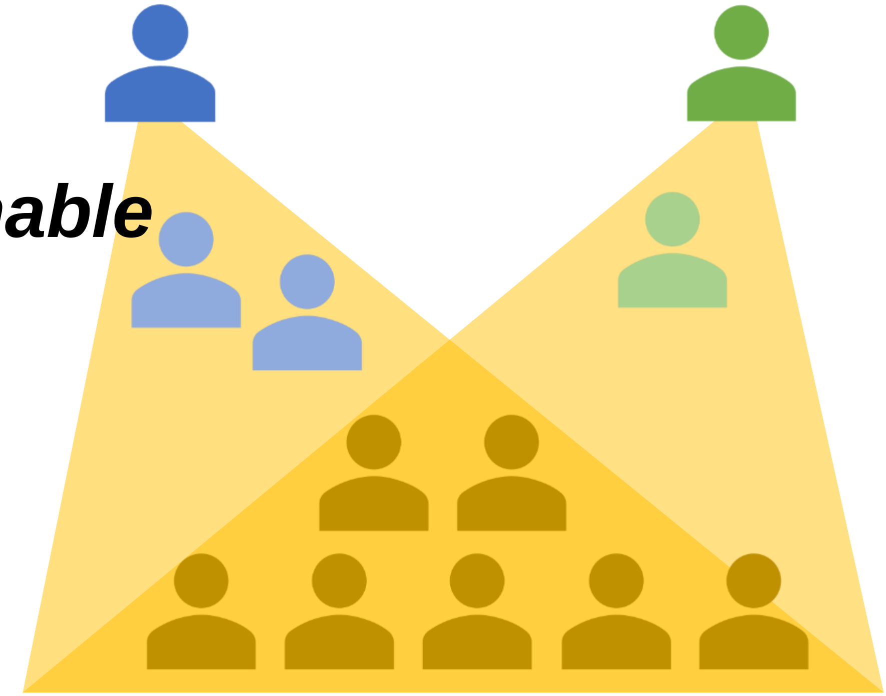
- When we increase the depth L more than this, GNNs face neighbor explosion $O(K^L)$
  - **Over-smoothing**
  - **Over-squashing**



Aggregation Depth in GNNs

Over-smoothing[10]

- When GNNs become deep, nodes share many neighbors
- Node embeddings become ***indistinguishable***

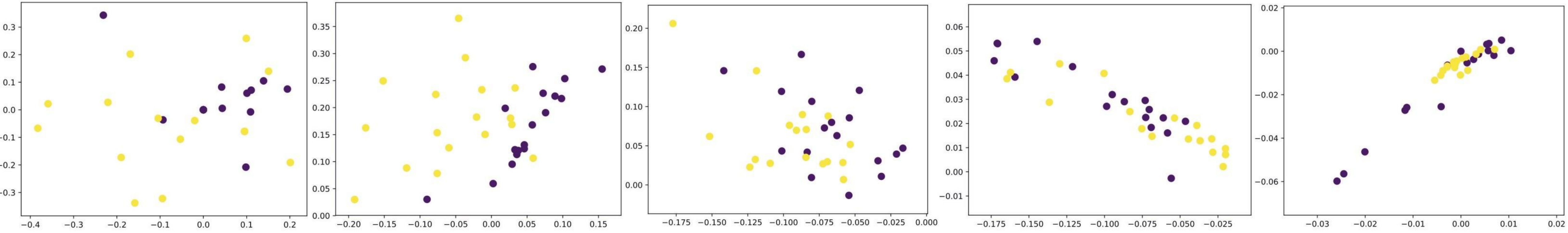


[10] Qimai Li, et al. "Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning"

Aggregation Depth in GNNs

Over-smoothing[10]

- Node embeddings of Zachary’s karate club network with GNNs



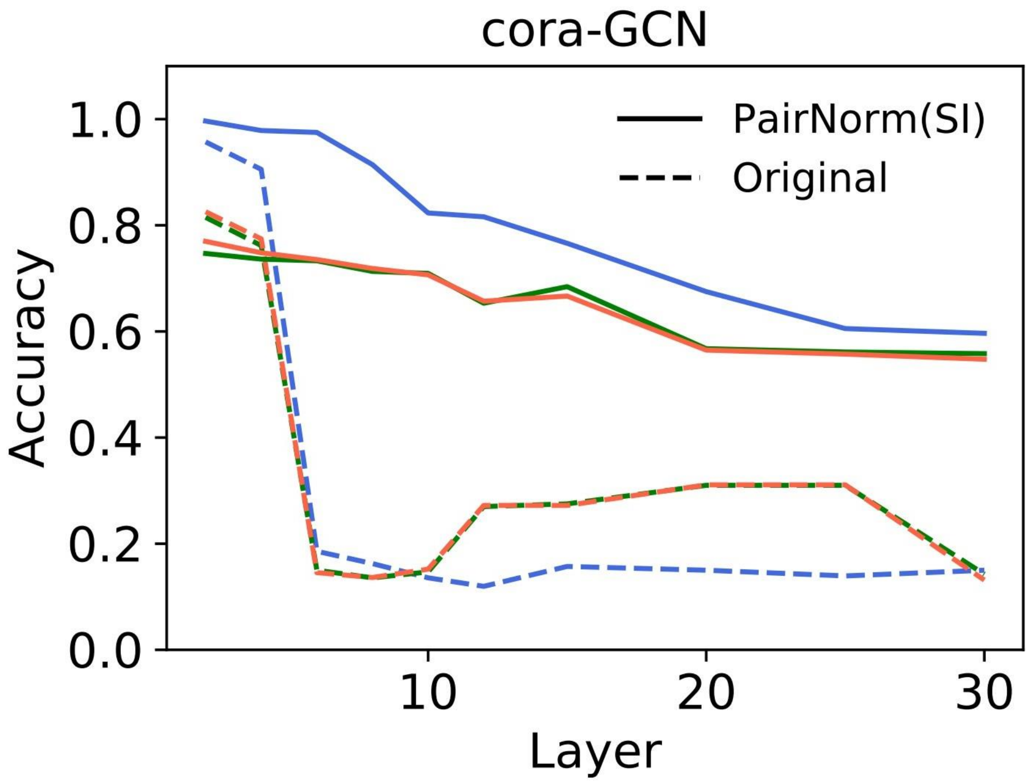
(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

[10] Qimai Li, et al. “Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning”

Aggregation Depth in GNNs

Mitigate over-smoothing

PairNorm[11]

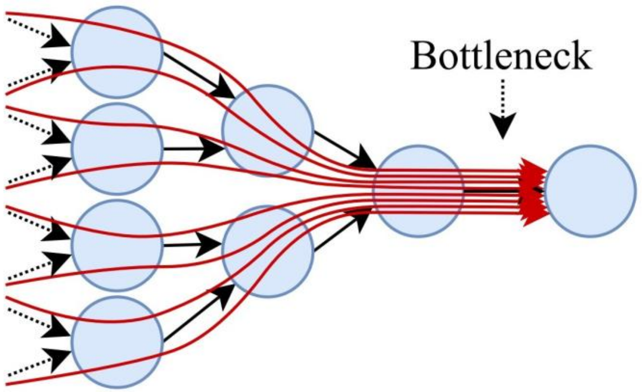


[11] Lingxiao Zhao, et al. “PAIRNORM: TACKLING OVERSMOOTHING IN GNNS”

Aggregation Depth in GNNs

Over-squashing[12]

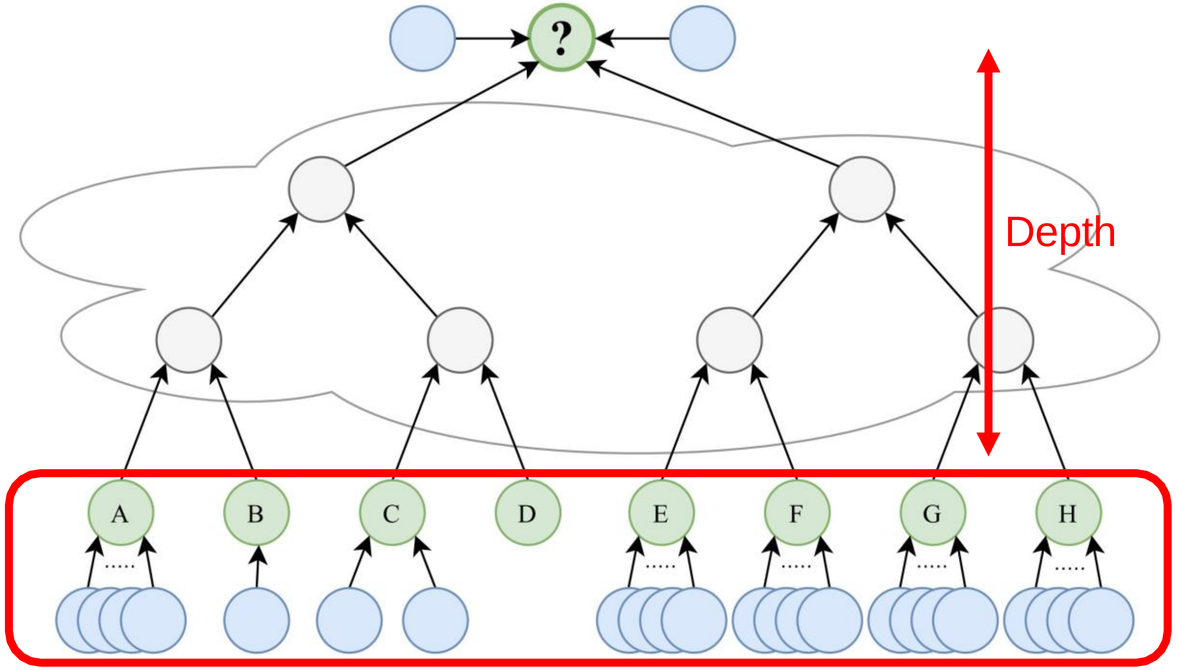
- A node’s exponentially-growing neighborhood is compressed into a fixed-size vector

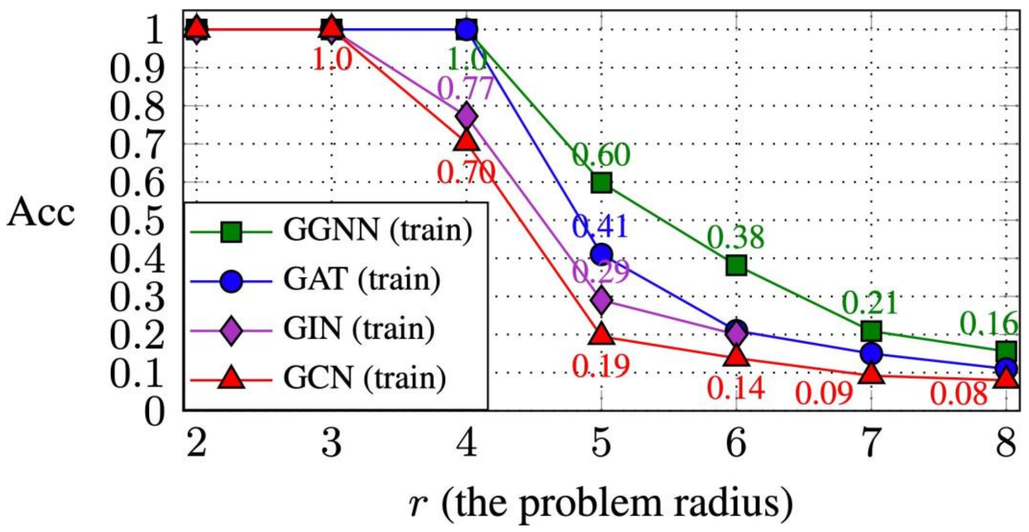


[12] Uri Alon, et al. “ON THE BOTTLENECK OF GRAPH NEURAL NETWORKS AND ITS PRACTICAL IMPLICATIONS”

Aggregation Depth in GNNs

Over-squashing[12]

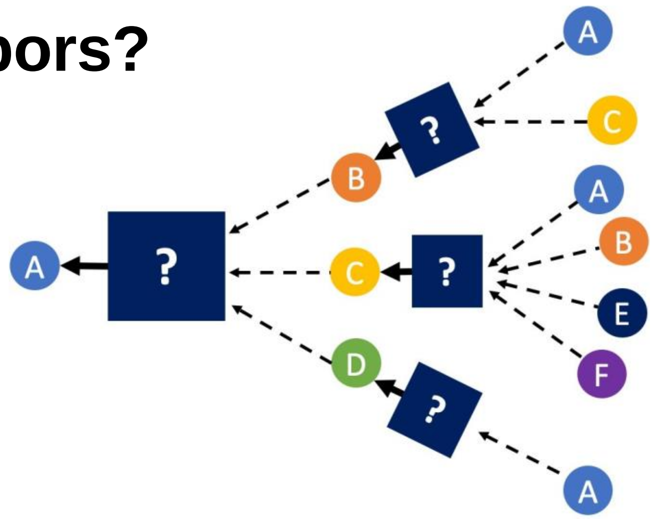




[12] Uri Alon, et al. "ON THE BOTTLENECK OF GRAPH NEURAL NETWORKS AND ITS PRACTICAL IMPLICATIONS"

Graph Neural Network Architectures

- Width
  - Which neighbors should we aggregate messages from?
- Depth
  - How many hops should we check?
- **Aggregation**
  - **How should we aggregate messages from neighbors?**



Aggregation strategy in GNNs

In each layer l :

Aggregate over neighbors

$$m_v^{(l-1)} = \boxed{\boldsymbol{f}^{(l)}}\left(h_v^{(l-1)}, \left\{h_u^{(l-1)} : u \in \mathcal{N}(v)\right\}\right)$$

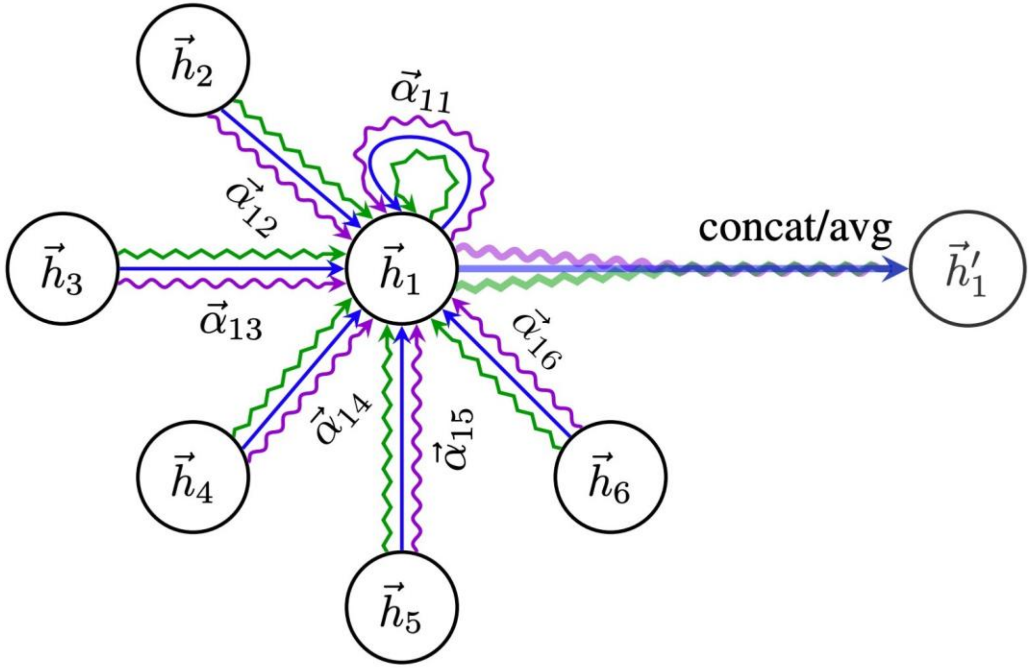
Transform messages

$$h_v^{(l)} = \boldsymbol{g}^{(l)}(m_v^{(l-1)})$$

Aggregation strategy in GNNs

- GAT[14]
  - Different weights to different nodes in a neighborhood
  - Multi-head attention

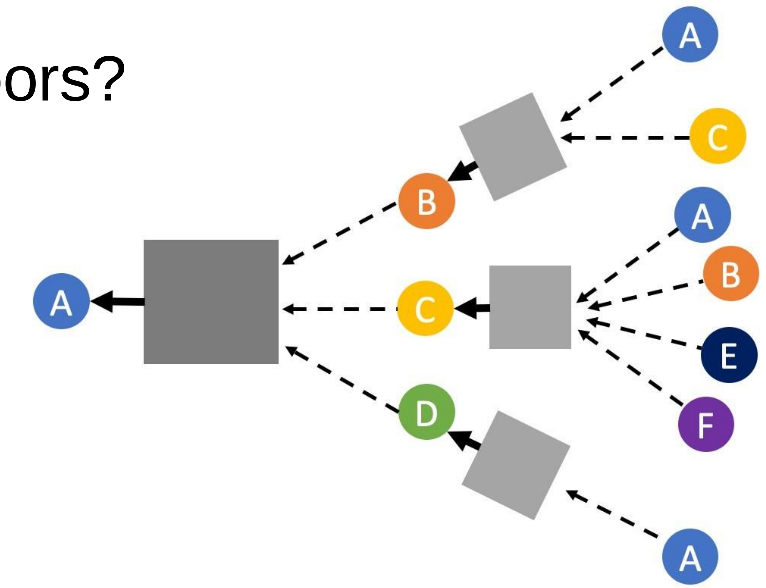
$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$



[14] Petar Veličković., et al. "GRAPH ATTENTION NETWORKS."

Graph Neural Network Architectures

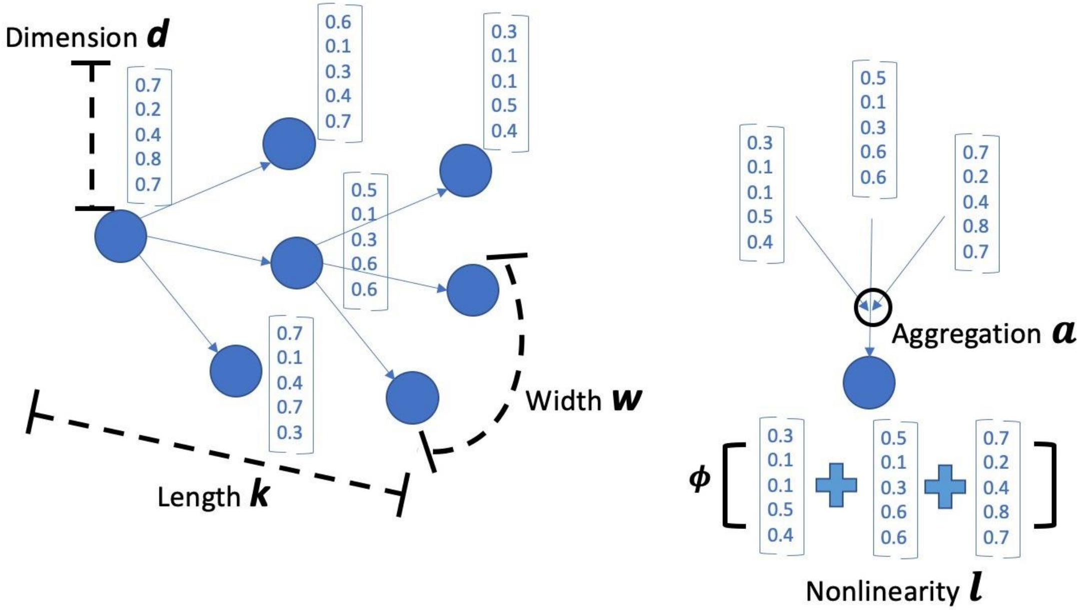
- Width
  - Which neighbors should we aggregate messages from?
- Depth
  - How many hops should we check?
- Aggregation
  - How should we aggregate messages from neighbors?



Neural Architecture Search for GNNs

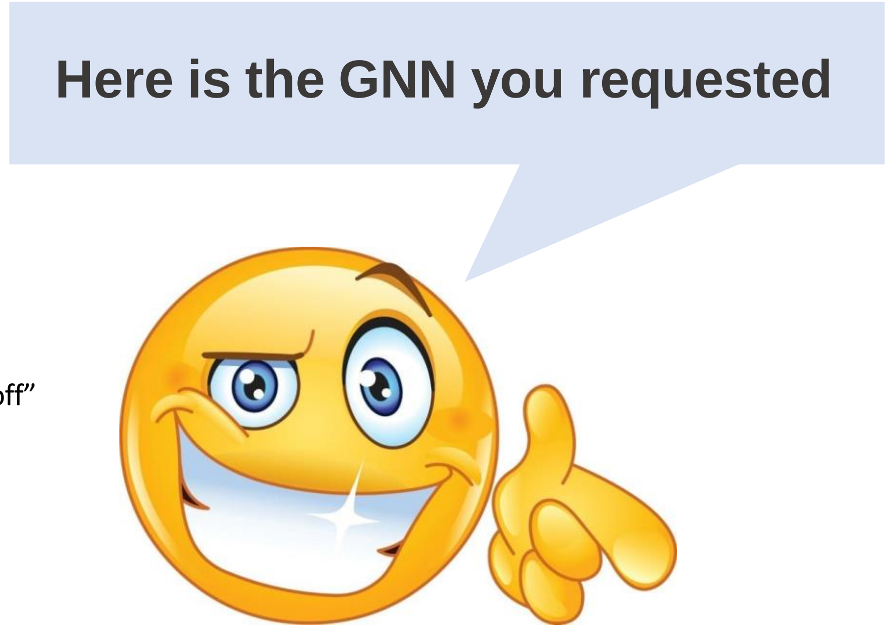
- Which *width, depth, and aggregation strategy* are proper for a given graph and task?

Width?
Depth?
Aggregation?



Neural Architecture Search for GNNs

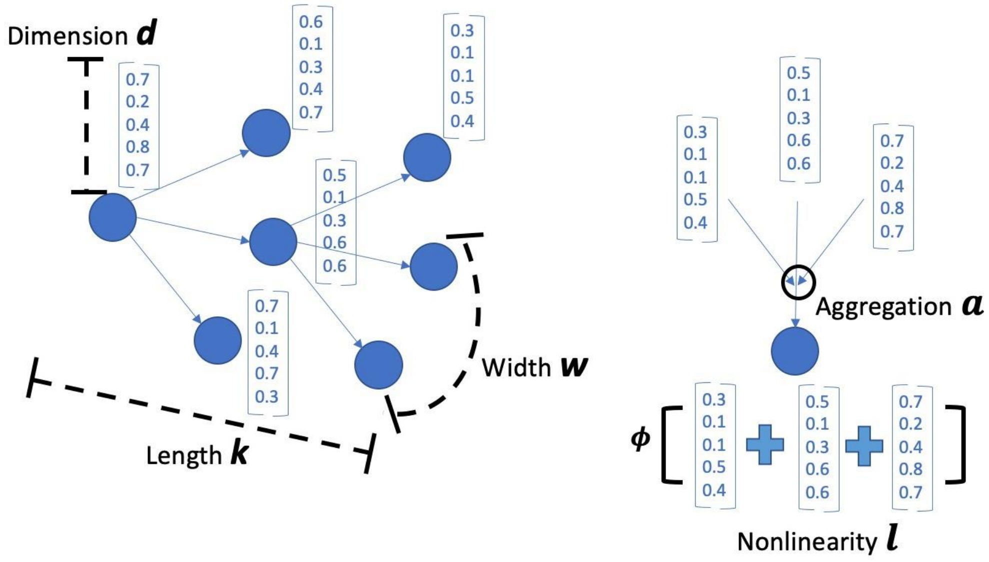
- Finding proper *width, depth, and aggregation strategy* for a given graph and task **automatically**[1,2,3]

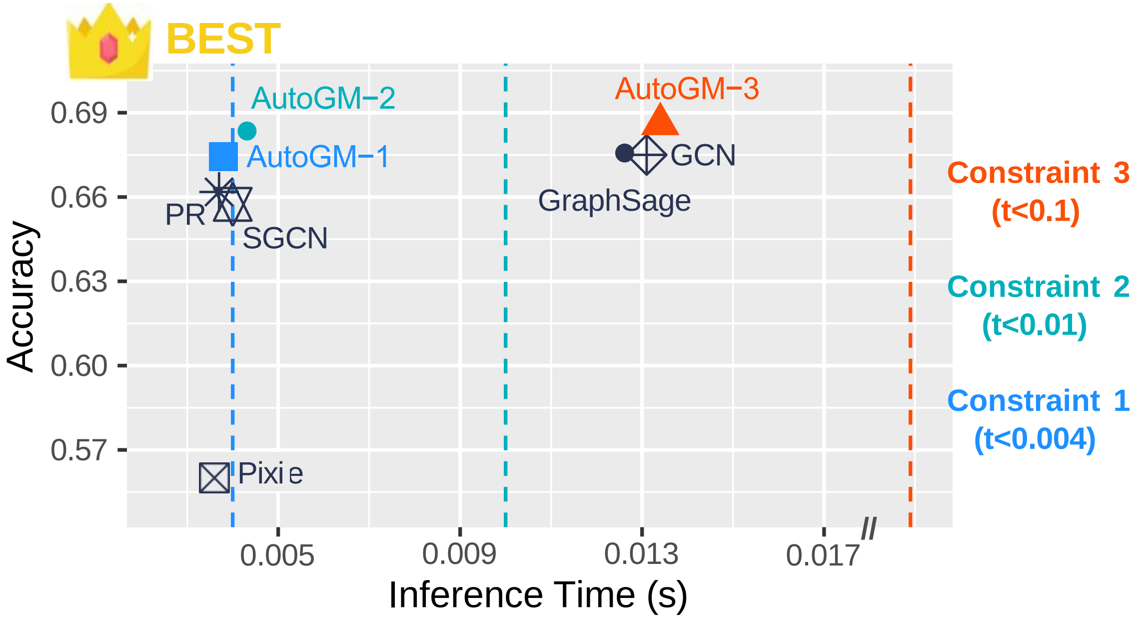


23 Minji Yoon., et al. "Autonomous Graph Mining Algorithm Search with Best Speed/Accuracy Trade-off"
24 Kaixiong Zhou, et al. “Auto-GNN: Neural Architecture Search of Graph Neural Networks”
25 Yang Gao, et al. “GraphNAS: Graph Neural Architecture Search with Reinforcement Learning”

Neural Architecture Search for GNNs

- AutoGM[23]





Step 1: define a hyperparameter space

Step 2: explore the space efficiently

[23] Minji Yoon., et al. "Autonomous Graph Mining Algorithm Search with Best Speed/Accuracy Trade-off"

Still many open problems..

- And many more chances to do groundbreaking research
- (ex) other graph formats
  - 3-dimensional graphs
  - Temporal graphs
  - ….

This Class

Part 1: Basic Concepts about Deep Learning

- Why Deep
- Forward and backward propagation
- Auto Diff, Pytorch
- Model training: convergence, learning rate, SGD, mini-batch, from Optimization aspect

Part 2: Popular Architectures

- Convolutional Neural Networks
- Recurrent Neural Networks
- LSTM, Seq-to-Seq, Attention, Transformer, LLMs
- GNNs

Part 3: Probabilistic deep learning, generative learning and beyond

- Generative learning, VAE, GAN, and Diffusion Models
- Bayesian deep learning
- Deep Probabilistic Graphical Models

This is our last lecture…

- I hope you enjoyed the class
  - Learn something you don’t know
  - Refresh something you already know
  - Spark your curiosity about something

- Keep exploring and Stay curious
  - What we talked about in class is just the beginning

- Thank you for your engagement
  - And of course, your questions!
  - Any comments are welcome in helping improving the class