

Graph Neural Networks

Wenxi Wang

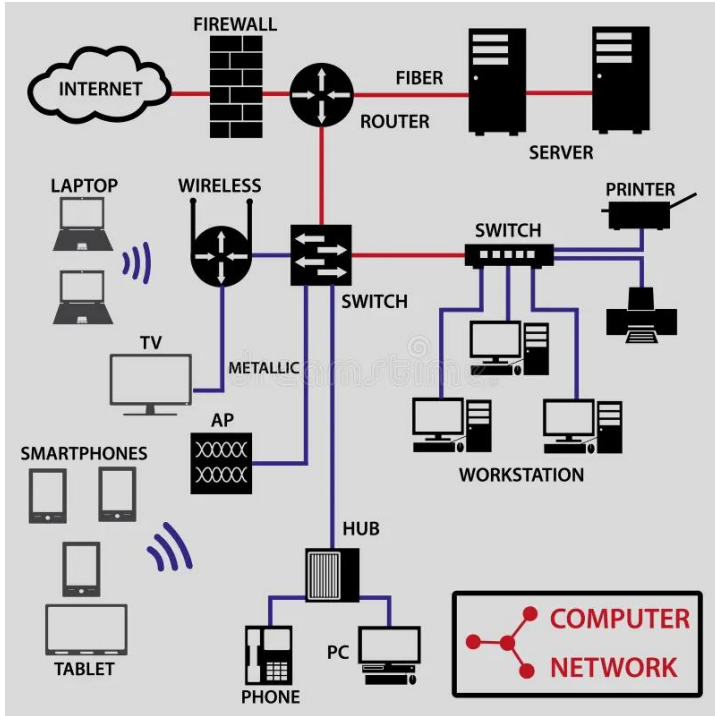
University of Virginia

wenxiw@virginia.edu

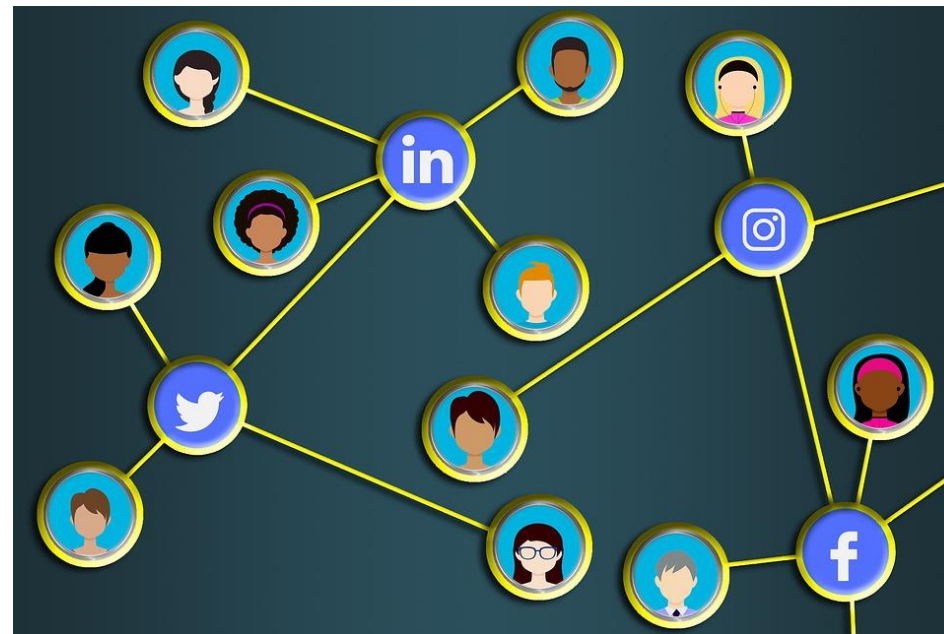


Why Graphs?

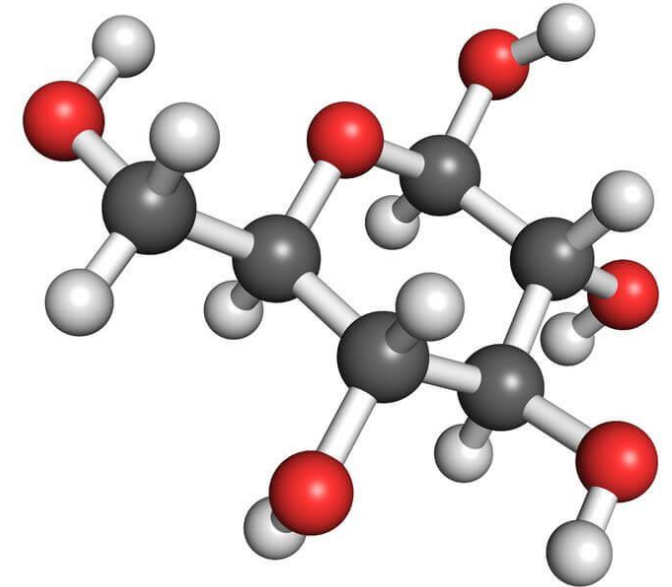
Graph is a general representation for specifying any entities and their relations/interactions.



Computer Network



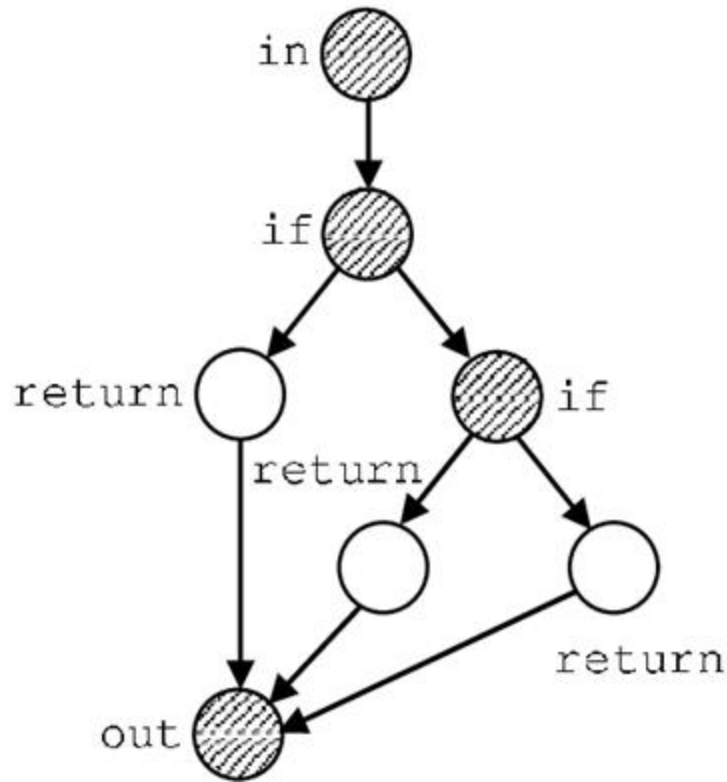
Social Network



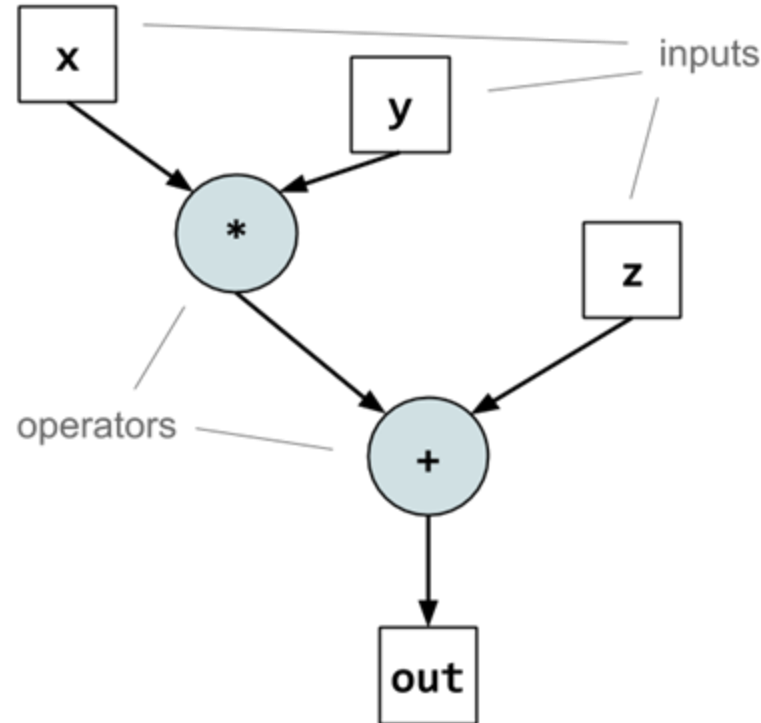
Molecules

Graphs for Software Engineering

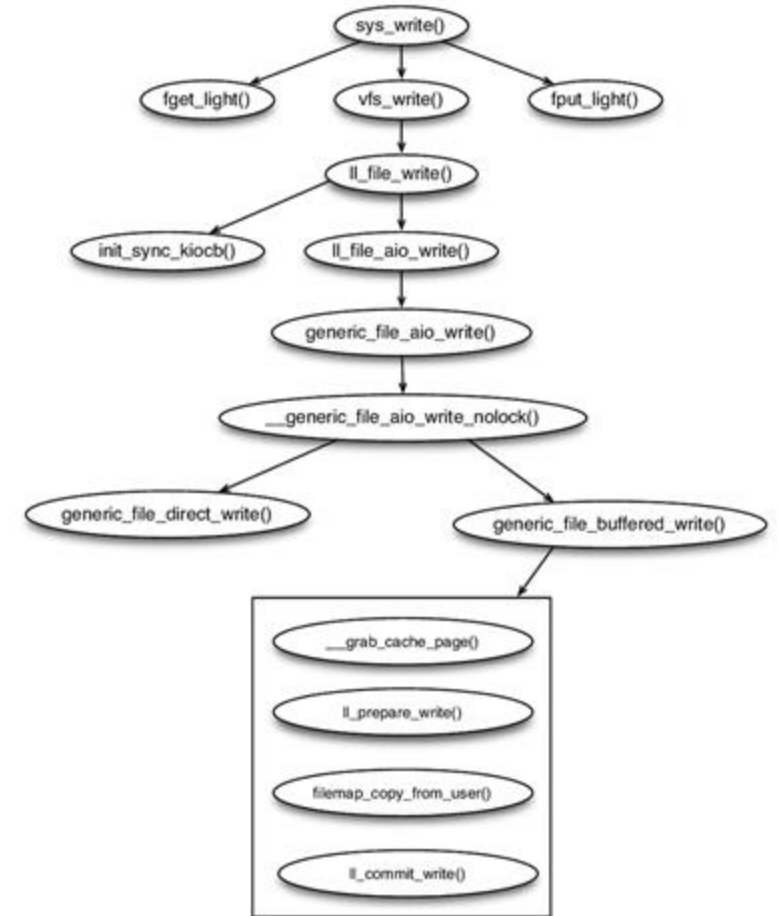
Software can be represented as graphs



Control Flow Graph



Data Flow Graph

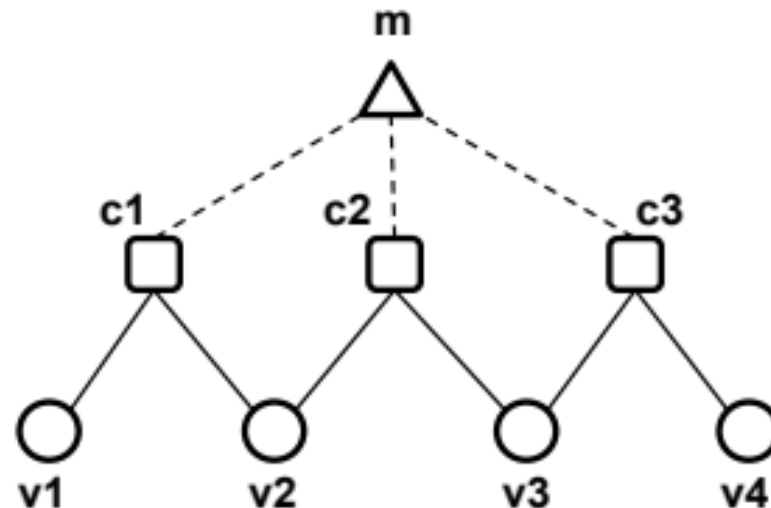


Call Graph

Graphs for Automated Reasoning

Logical formulas can be represented as graphs

Boolean formula: $(v1 \vee v2) \wedge (v2 \vee v3) \wedge (v3 \vee v4)$



Tasks on Graphs

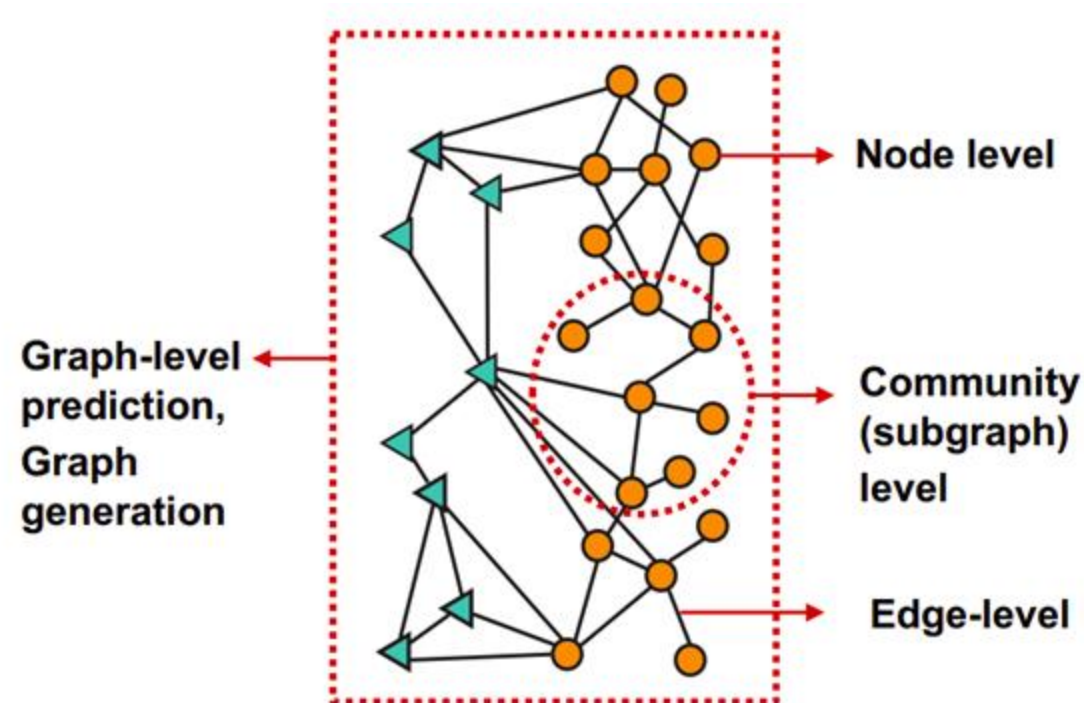
Typical Prediction Tasks (e.g., classification/regression) on Graphs

Node Level Prediction

Edge Prediction

Subgraph Level Prediction

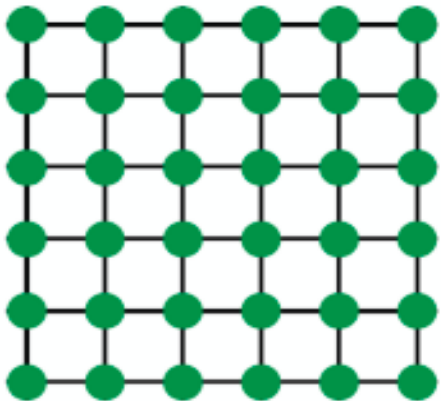
Graph Level Prediction



Motivations

Can classic deep learning techniques (e.g., CNN, RNN, LSTM) accomplish those tasks?

Mostly designed for handling the two kinds of graphs:
grids and sequences



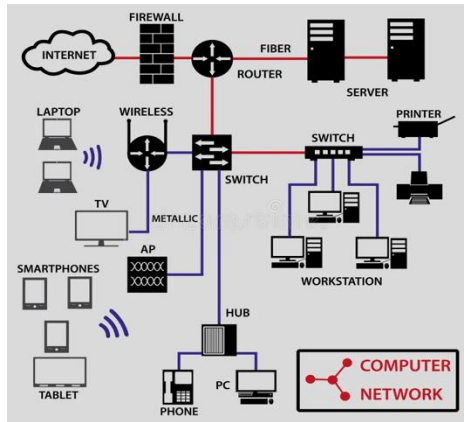
Graph representation for an images



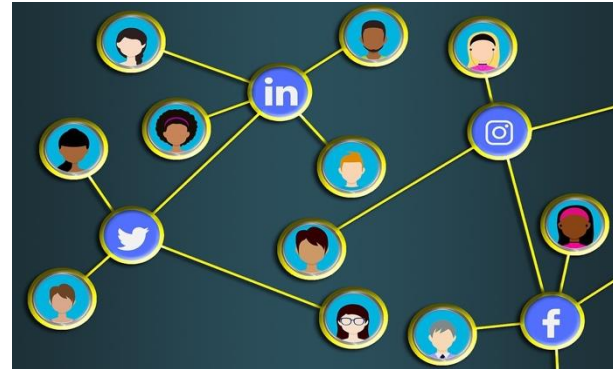
Graph representation for text

Motivations

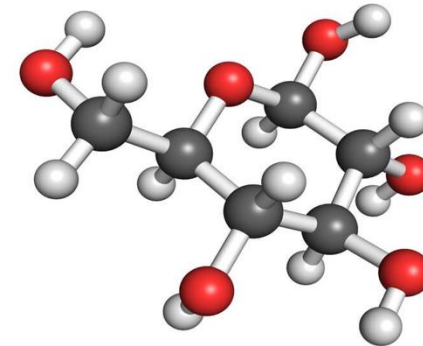
How to handle other types of graphs?



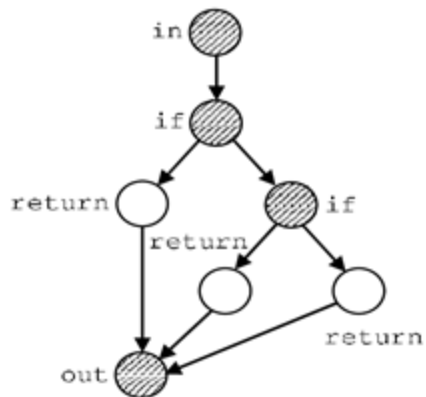
Computer Network



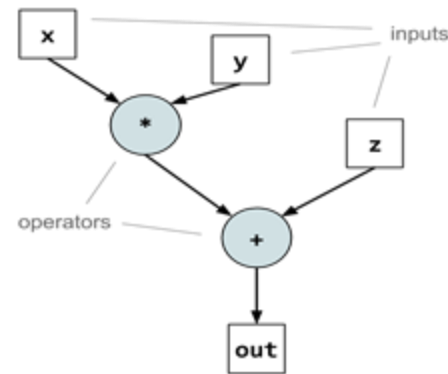
Social Network



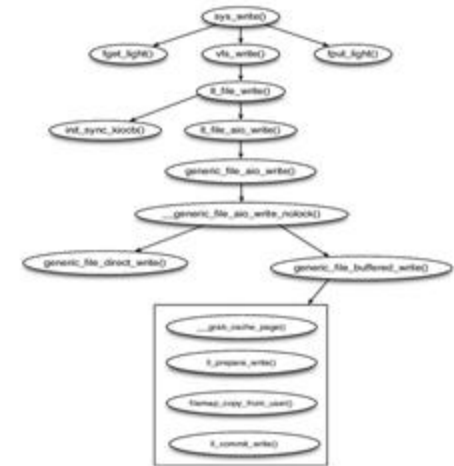
Molecules



Control Flow Graph



Data Flow Graph



Call Graph

Solution: Graph Neural Networks

Proposed in 2005^[1], became popular in 2017^[2]

Because of a powerful mechanism

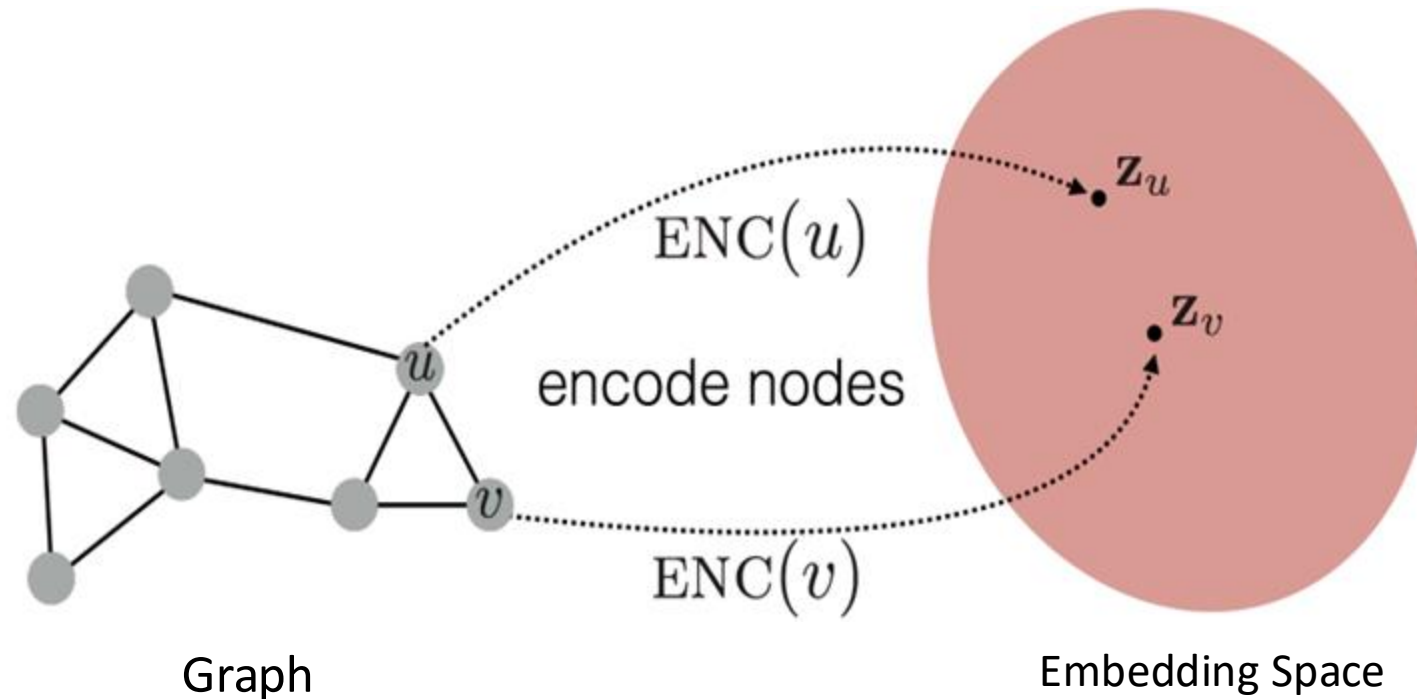
[1] Gori et al. *A new model for learning in graph domains*, 2005

[2] Gilmer et al. *Neural message passing for quantum chemistry*, ICML, 2017

GNN

Key idea: Node Embedding

Encode nodes into embeddings such that similar nodes in the graph are embedded close together.



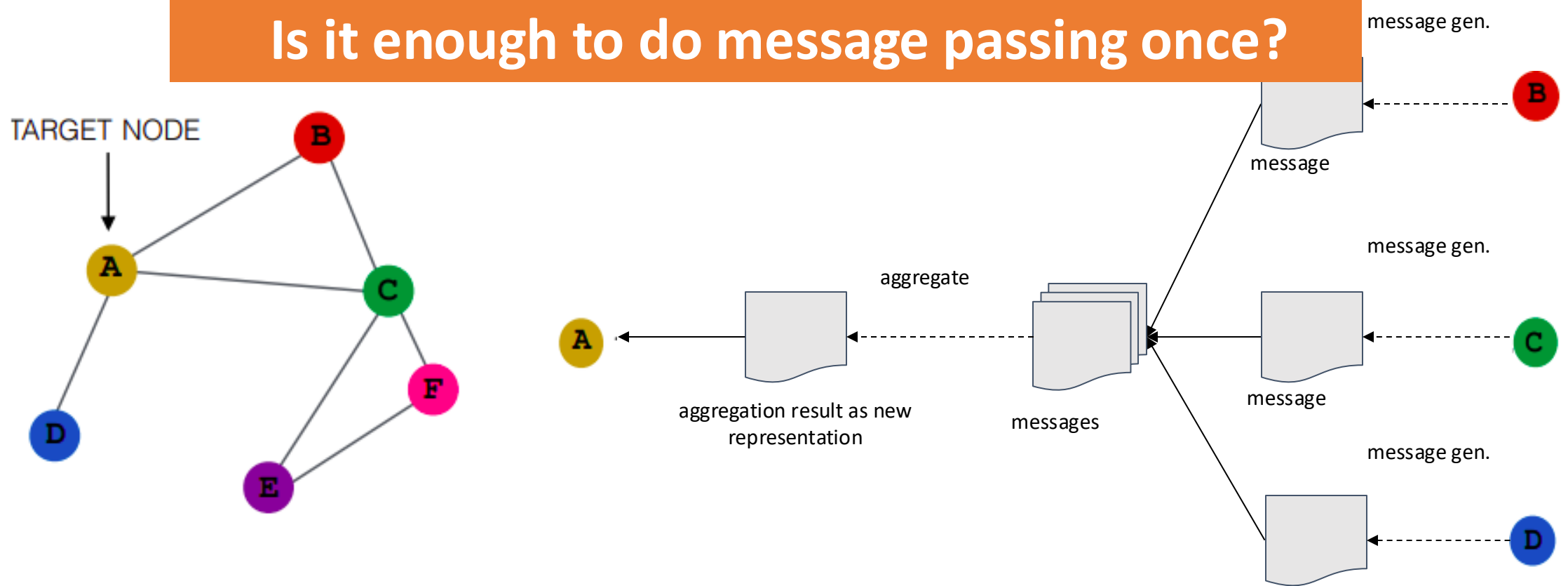
How to realize the encoder ENC?

GNN

Powerful mechanism: Message Passing

For each node, update its embedding based on its neighbor's embeddings

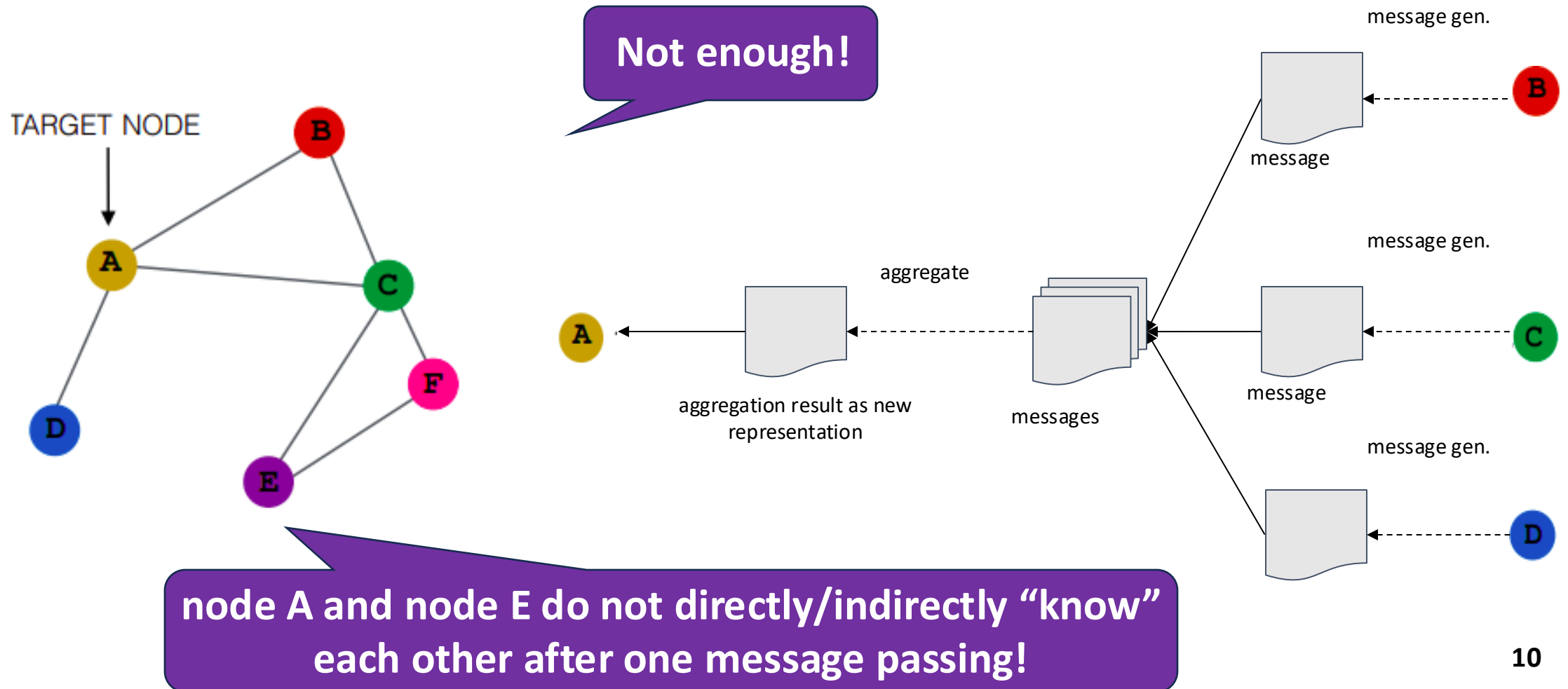
Is it enough to do message passing once?



GNN

Key idea: Message Passing

For each node, update its embedding based on its neighbor's embeddings

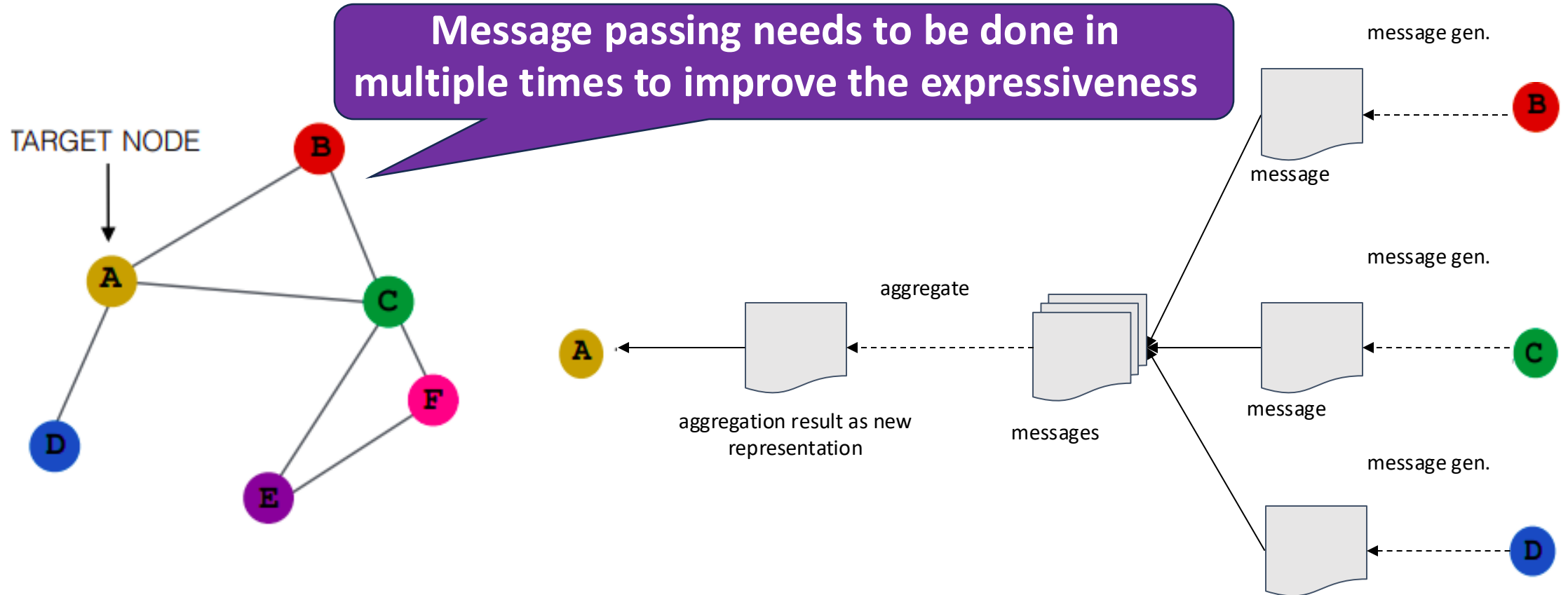


GNN

Key idea: Message Passing

For each node, update its embedding based on its neighbor's embeddings

Message passing needs to be done in multiple times to improve the expressiveness



GNN

So far, in summary

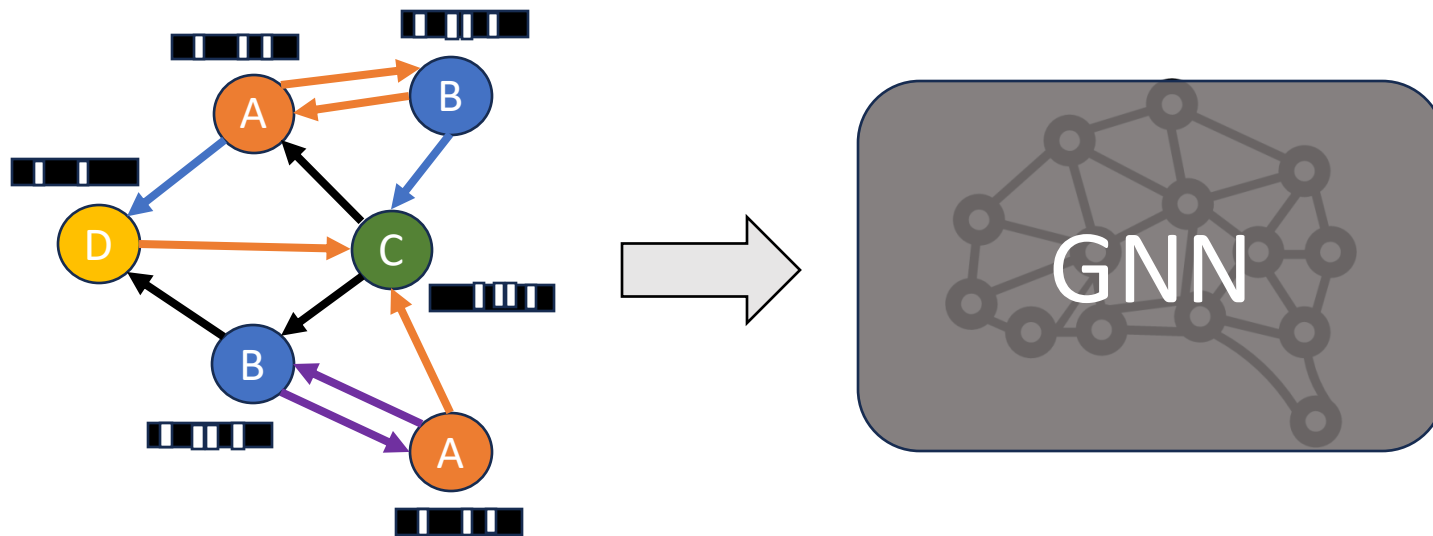
GNN

A type of neural networks



GNN

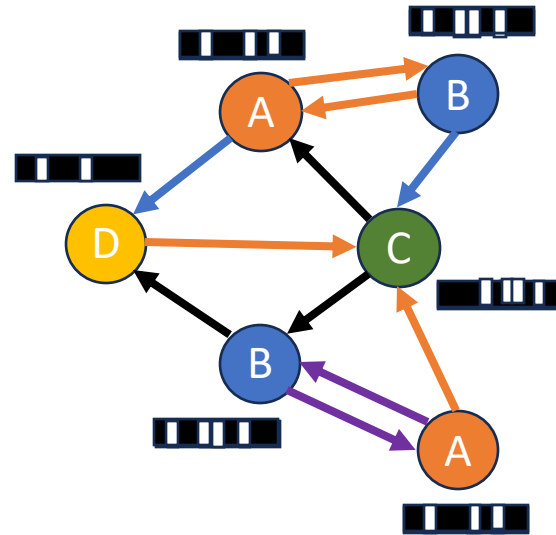
Operates on graph structured data



Initial node feature vectors

GNN

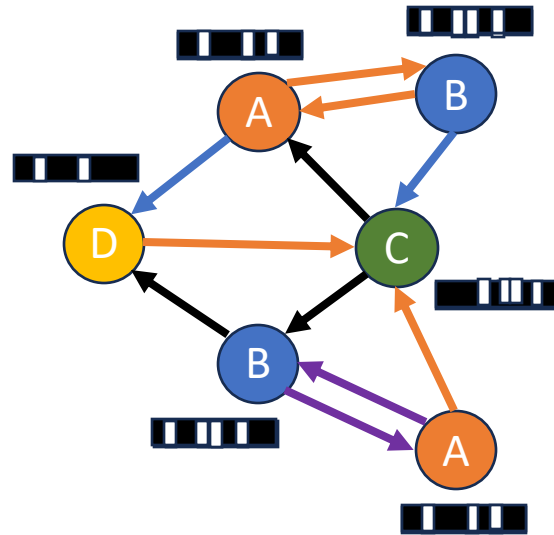
Message passing



GNN

Message passing

- aggregating and transforming node and edge information

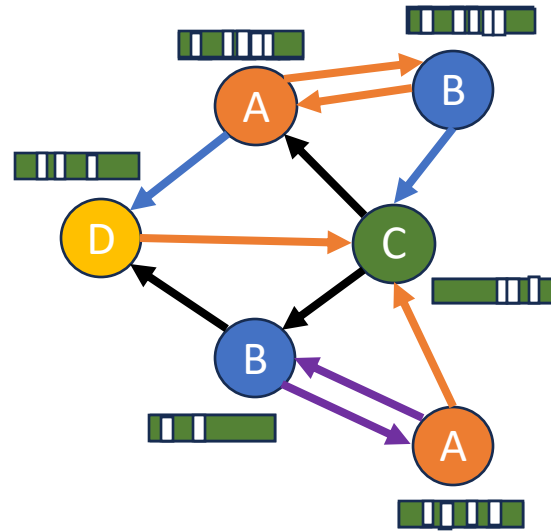


Round 1

GNN

Message passing

- aggregating and transforming node and edge information

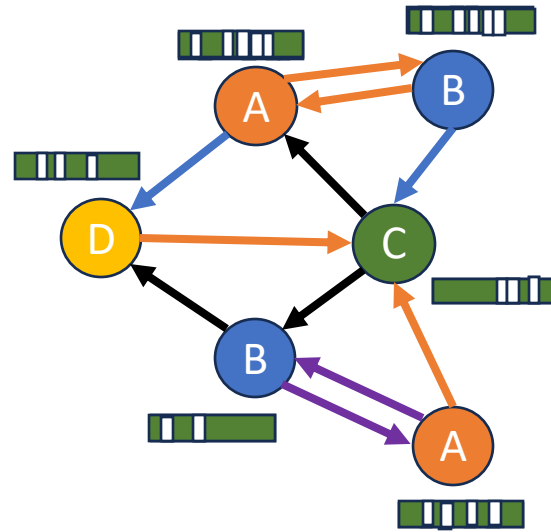


Round 1

GNN

Message passing

- aggregating and transforming node and edge information

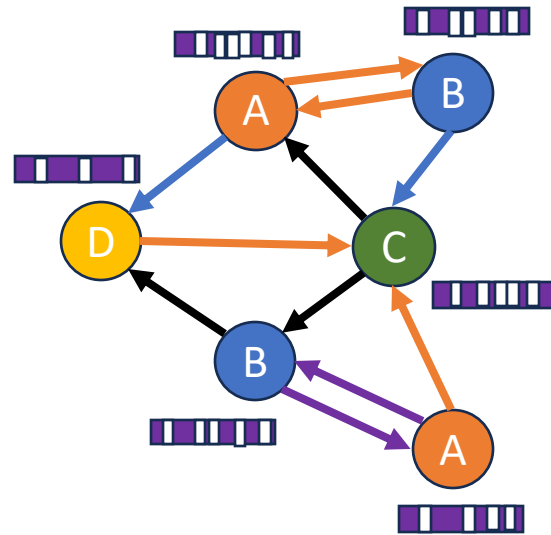


Round 2

GNN

Message passing

- aggregating and transforming node and edge information

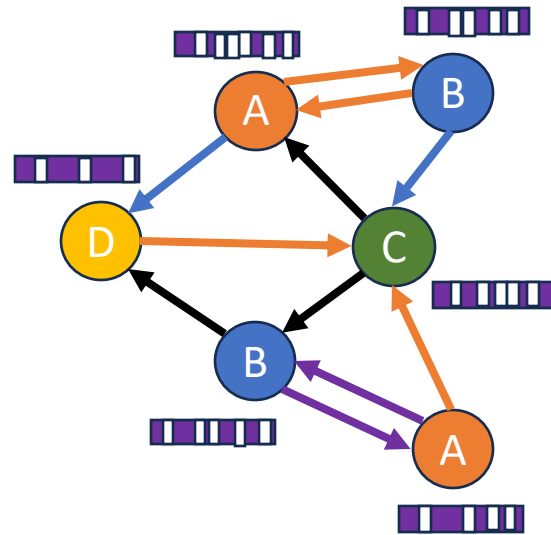


Round 2

GNN

Message passing

- aggregating and transforming node and edge information

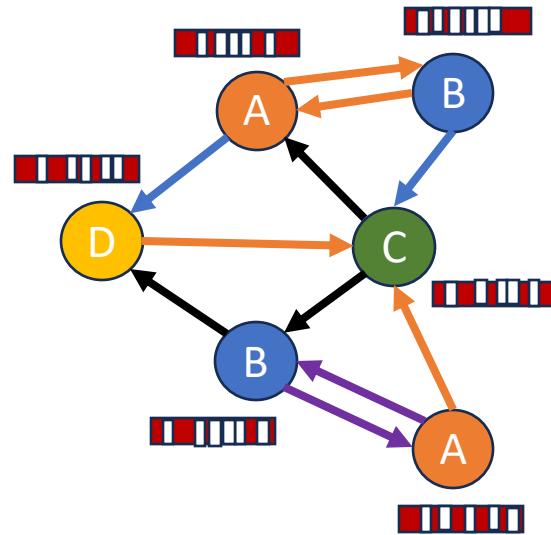


Round 3, 4, 5, ...

GNN

Message passing

- aggregating and transforming node and edge information

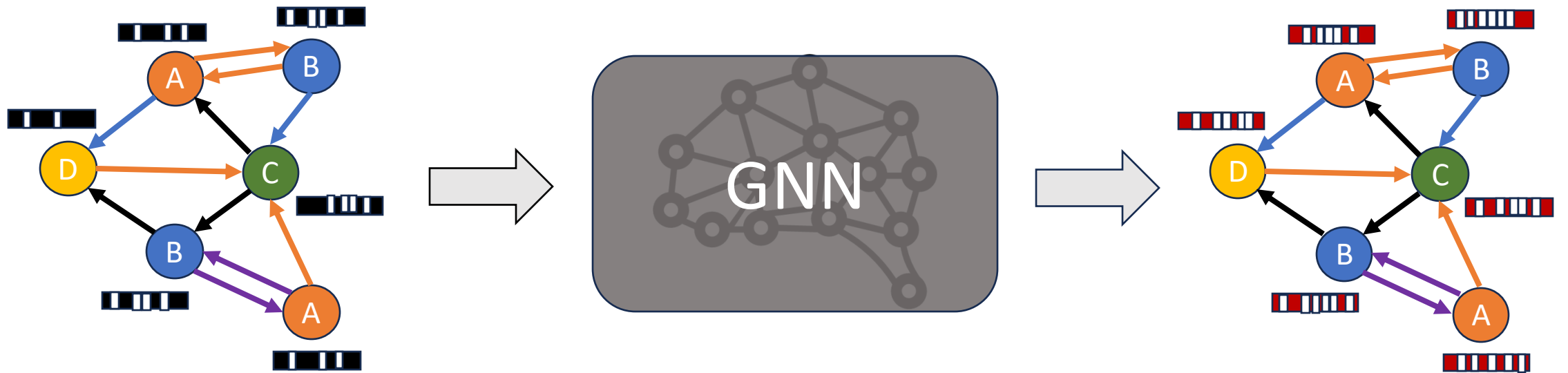


Round n

GNN

Capture graph structures

- reason about complex relationships/dependencies



Initial node feature vectors

Updated node embeddings

GNN

More message passing always result in better expressiveness?



Too many message passing may cause **over-smoothing** issue^[1]!

make embeddings indistinguishable

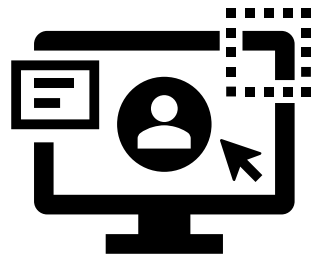
GNN

Ways to mitigate over-smoothing issue

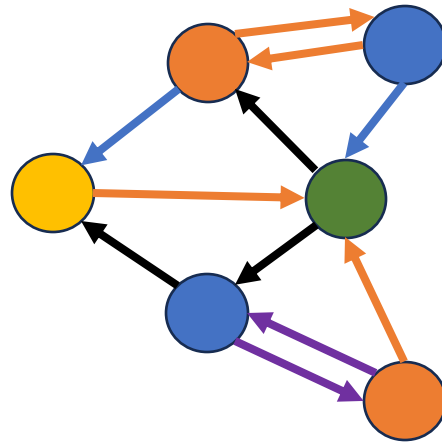
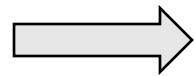
- **Restrict the number of message passing operations/layers (e.g., to the diameter of the graph).**
- **Normalization and regularization**
- **Residual/skip connections**

GNN for SE

Many SE problems can be naturally converted into graphs
without information loss



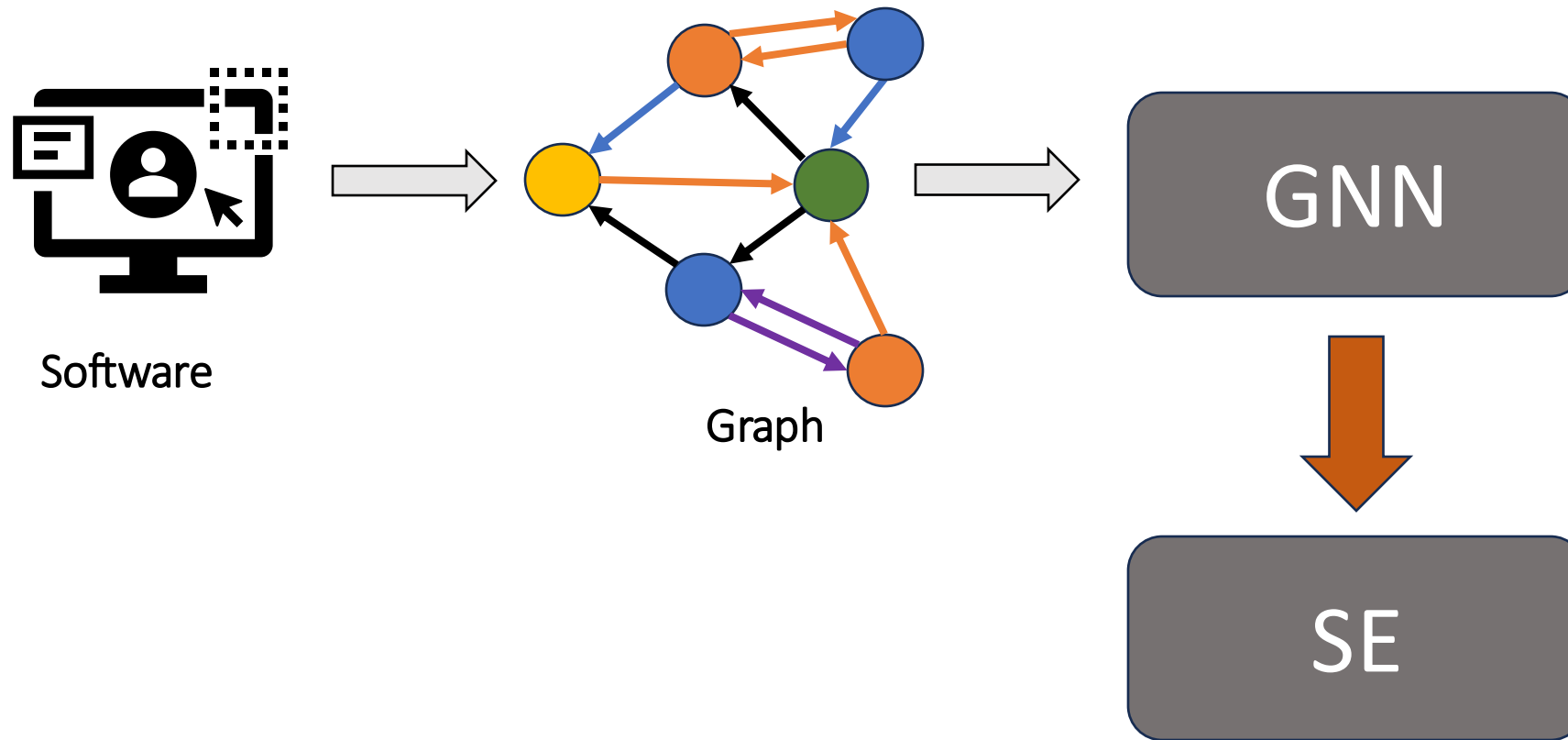
Software



No information loss!

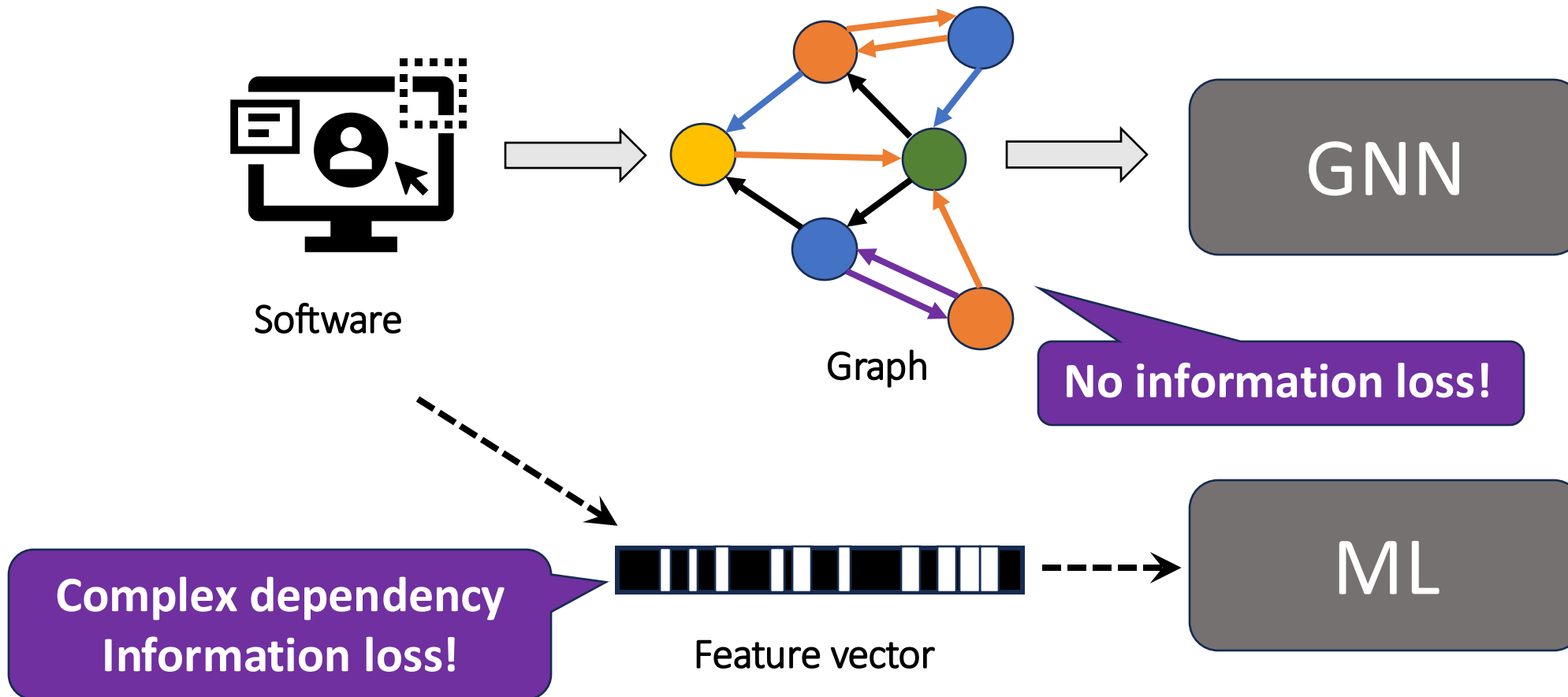
GNN for SE

GNN captures complex dependency information of SE problems



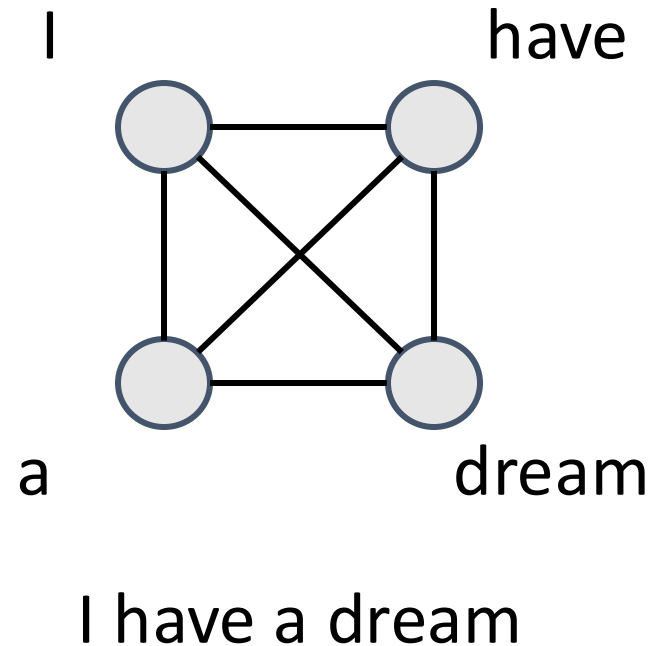
GNN for SE

GNN captures complex dependency information of SE problems



Broader View

Transformer can be regarded as a kind of GNN
on a **fully connected word graph***



* please refer to Page 71, <https://web.stanford.edu/class/cs224w/slides/03-GNN1.pdf>

Reinforcement Learning

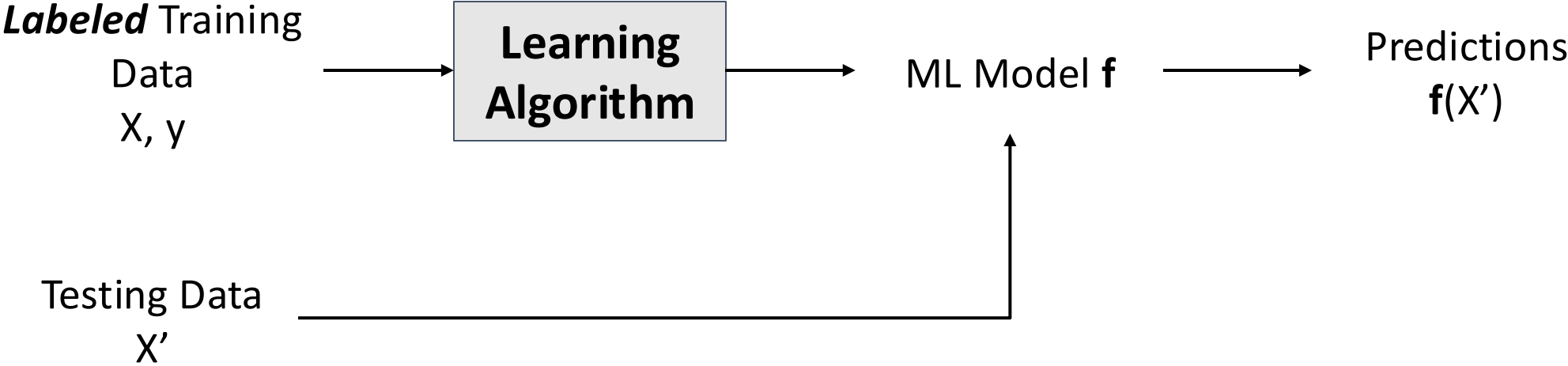
Wenxi Wang

University of Virginia

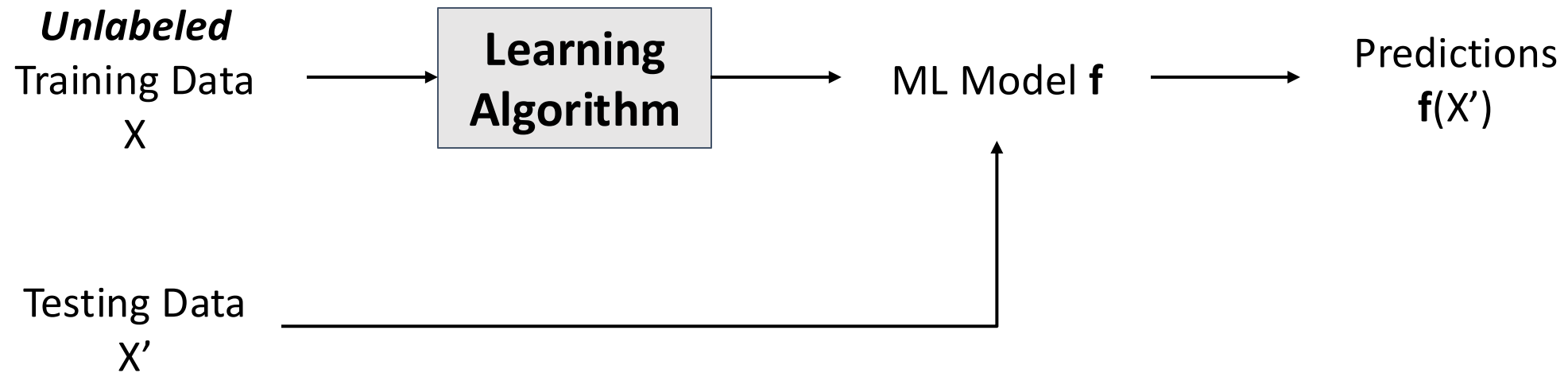
wenxiw@virginia.edu



Machine Learning Basics - Supervised Learning



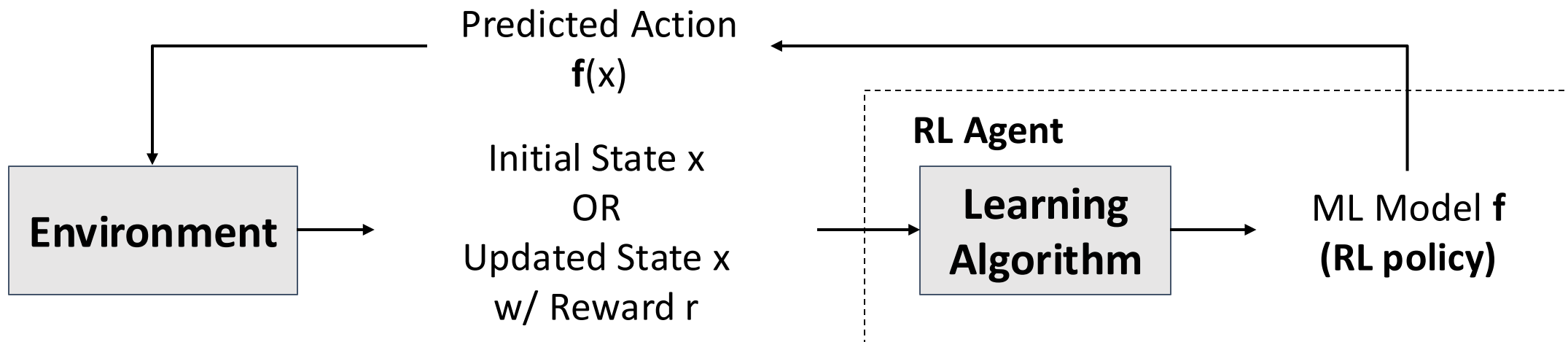
Machine Learning Basics - Unsupervised Learning



Machine Learning Basics - Reinforcement Learning

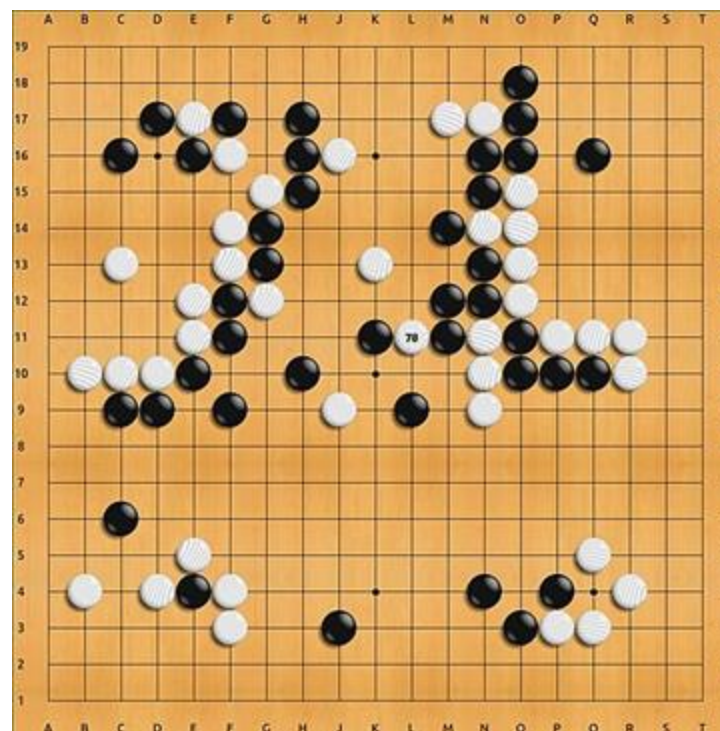
Reinforcement Learning differs from supervised/unsupervised learning in the following perspectives:

- The ML model (i.e, RL policy) is used for **decision making** (i.e., select an action based on the current state)
- The current state is not collected from training or testing datasets, but from the **environment**.
- There is no ground truth action (i.e., label) for each input state, but a **reward** which quantitatively assesses the decisions made by the RL policy is applied.
- The goal of the learning algorithm is to **maximize cumulative rewards**.



Why Reinforcement Learning?

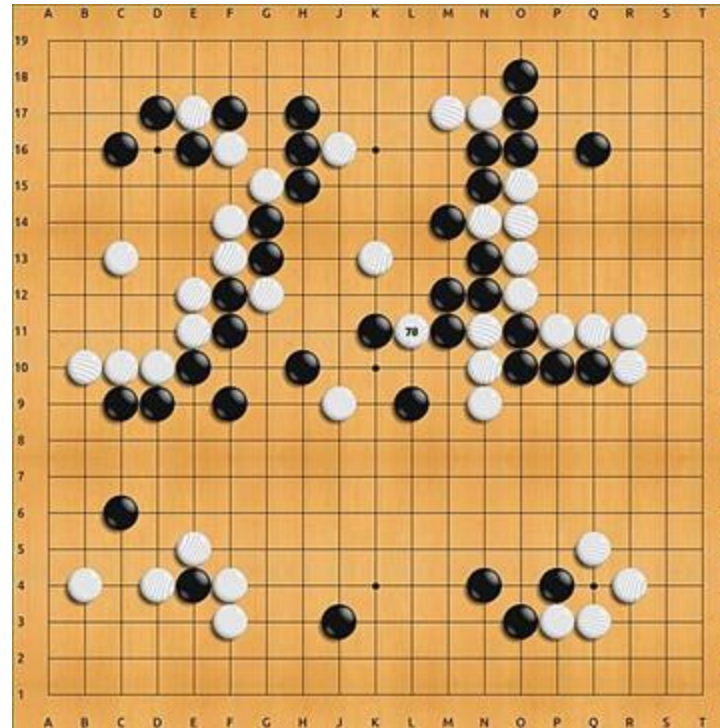
Supervised/unsupervised learning may not be suitable for **sequential decision making** problems.



The decision making problem in the Go game: given the current state of the board, decide where to put the next stone (e.g., the 78th stone)

Why Reinforcement Learning?

Supervised/unsupervised learning may not be suitable for sequential decision making problems.

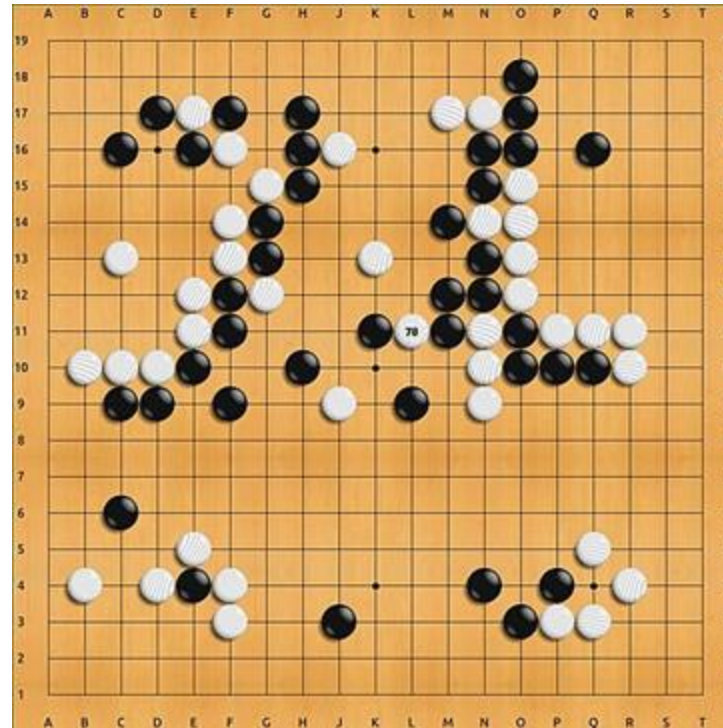


Unsupervised learning? It is not a typical unsupervised learning problem like clustering.

The decision making problem in the Go game: given the current state of the board, decide where to put the next stone (e.g., the 78th stone)

Why Reinforcement Learning?

Supervised/unsupervised learning may not be suitable for sequential decision making problems.



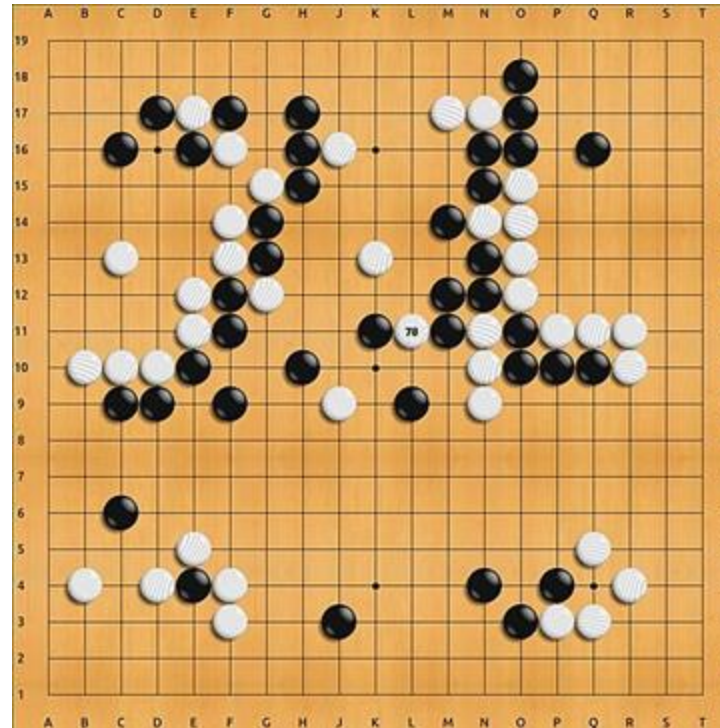
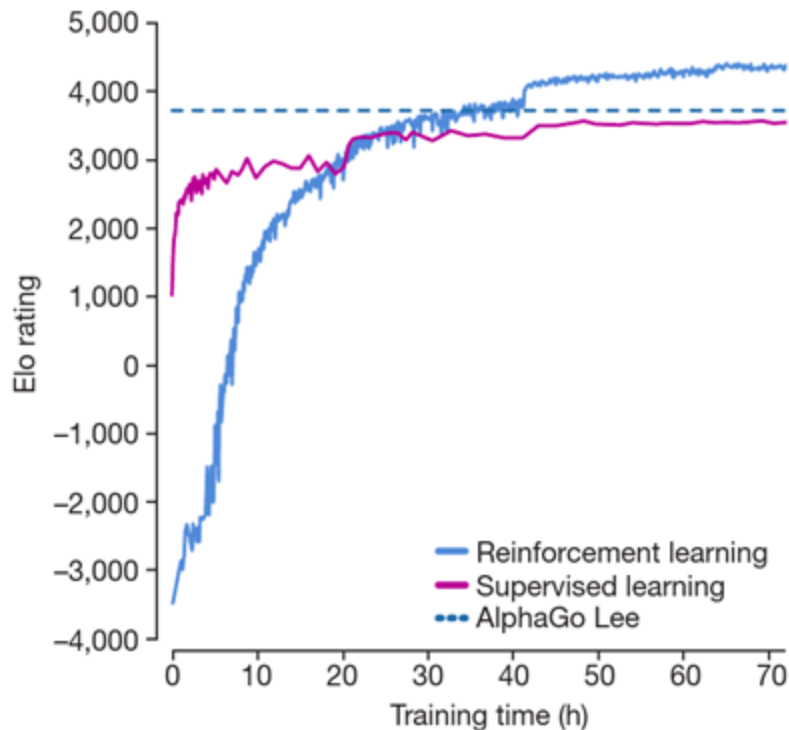
Unsupervised learning? It is not a typical unsupervised learning problem like clustering.

Supervised learning? How to obtain the ground truth? Existing human Go player records may not always be optimal!

The decision making problem in the Go game: given the current state of the board, decide where to put the next stone (e.g., the 78th stone)

Why Reinforcement Learning?

Supervised/unsupervised learning may not be suitable for sequential decision making problems.



Reinforcement Learning

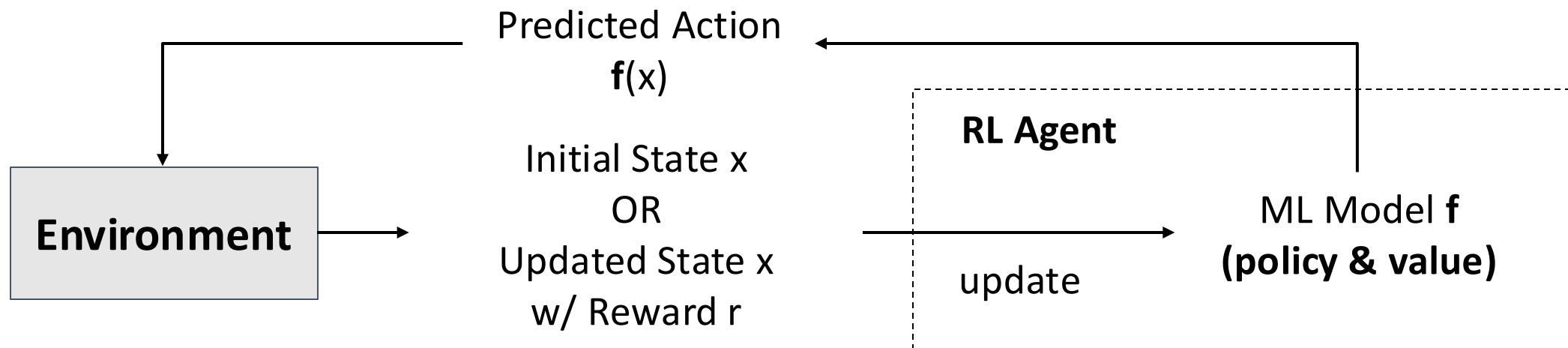
- **State:** all previous stone moves on the board
- **Action:** a stone move
- **Reward**
 - +1 for winning the game
 - -1 for losing the game
 - 0 for not winning or losing

The decision making problem in the Go game: given the current state of the board, decide where to put the next stone (e.g., the 78th stone)

Deep Reinforcement Learning Technique in AlphaGo Zero

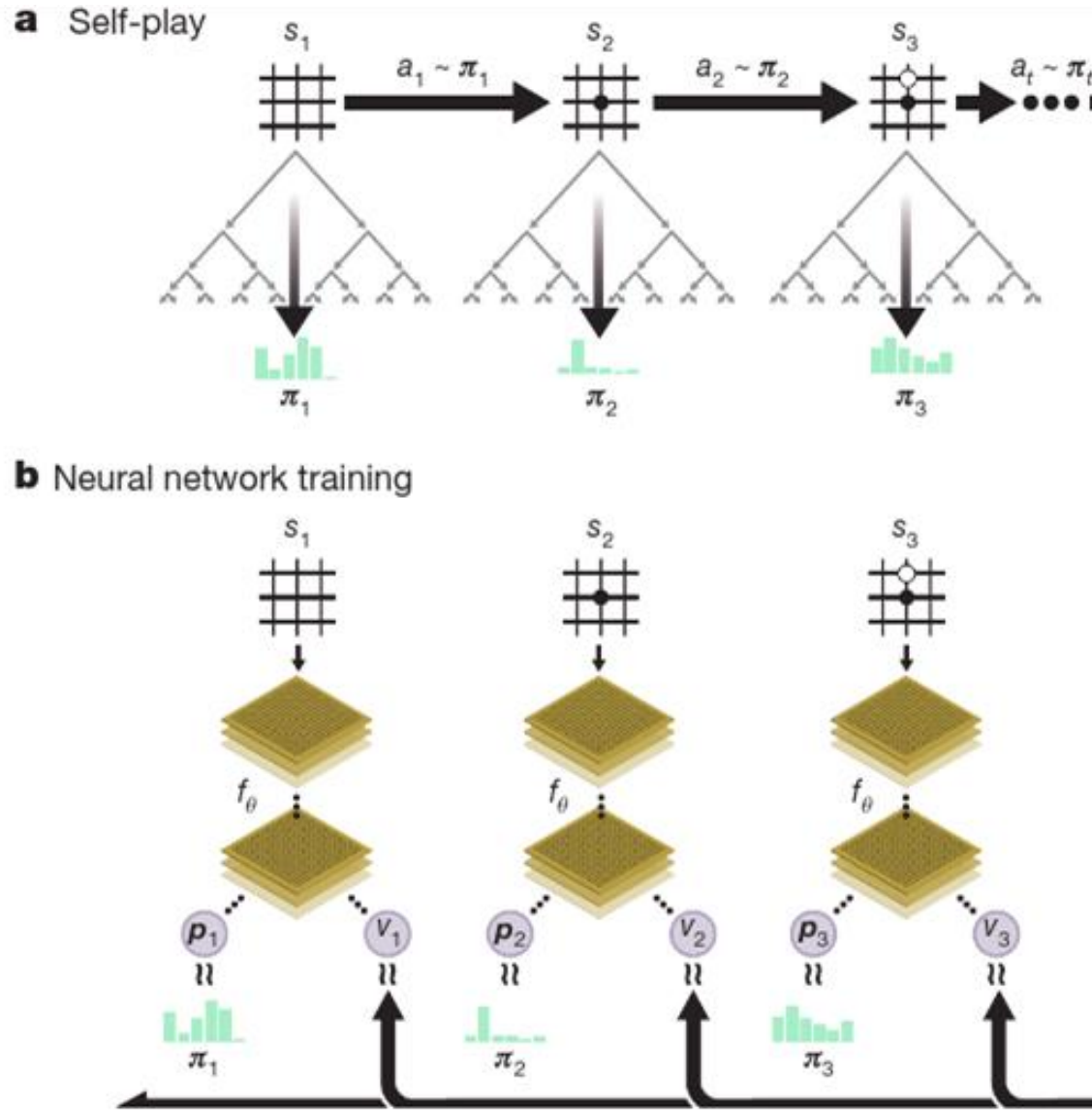
Core Ideas

- **Value Network:** a neural network for predicting a score of winning the game based on the current state. It is optimized using the reward.
- **Policy Network:** a neural network for selecting one stone move based on the current state. It is optimized towards a vector of search probabilities estimated via **Monte Carlo Search Tree**.
- **Policy network and value network can be combined into one network.**



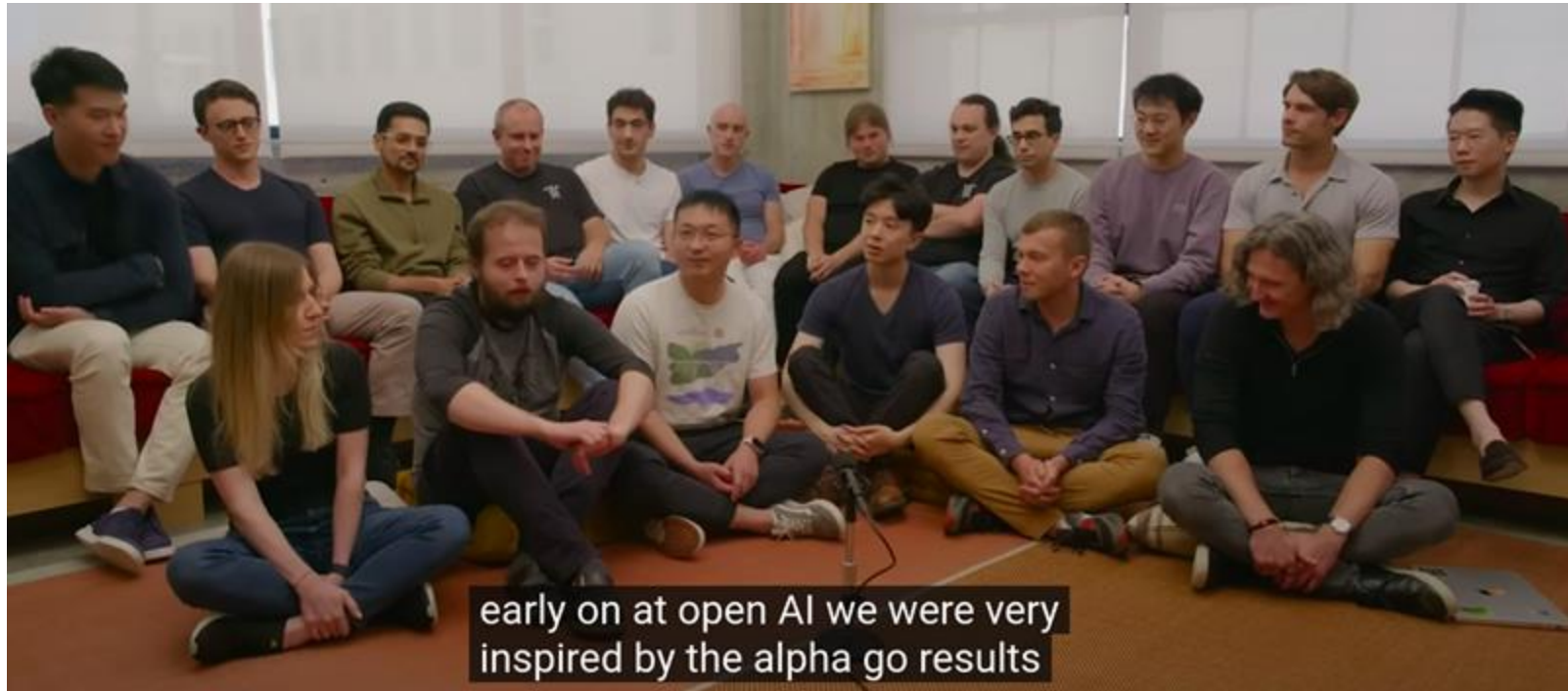
Deep Reinforcement Learning Technique in AlphaGo Zero

Core Ideas



Fun Fact

OpenAI o1 applies RL and is claimed to be inspired by AlphaGo *



More Learning Materials for Reinforcement Learning

- [AlphaGo Zero Paper](#)
- [Reinforcement Learning Book by Richard Sutton](#)