



# Graph Neural Networks

## And a Myriad of Things that are Relevant

Author: Viet Nhat Nguyen

University of Information Technology  
[21520378@gm.uit.edu.vn](mailto:21520378@gm.uit.edu.vn)





# Table of Contents

## 1 Fantastic Graphs and Where to Find Them

What is a Graph?

Where to Find them?

What Tasks to Perform on Graphs?

The challenges of using graphs in ML

## 2 Graph Neural Networks

The simplest GNN

GNN Predictions by Pooling Information

Passing messages between parts of the graph

## 3 Into the Weeds

Other types of graphs





What is a Graph?

# What is a Graph?



$$G = \{V, E\}$$

- A graph represents the relation (*edges*) between a collection of entities (*nodes*).
- $V$  Vertices or nodes
- $E$  Edges or links
- $G$  Graph

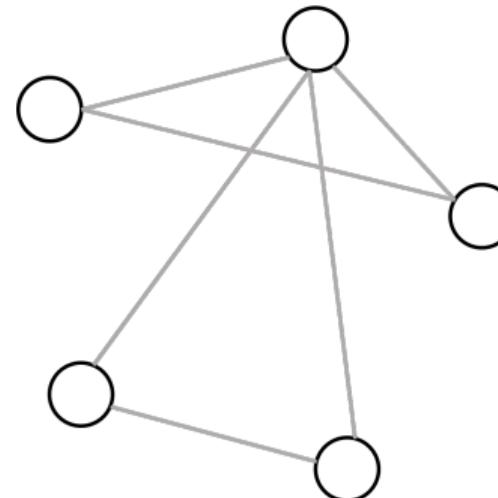




What is a Graph?



# What is a Graph?

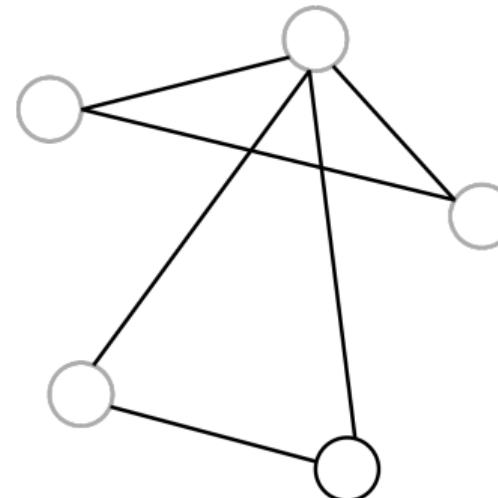




What is a Graph?



# What is a Graph?

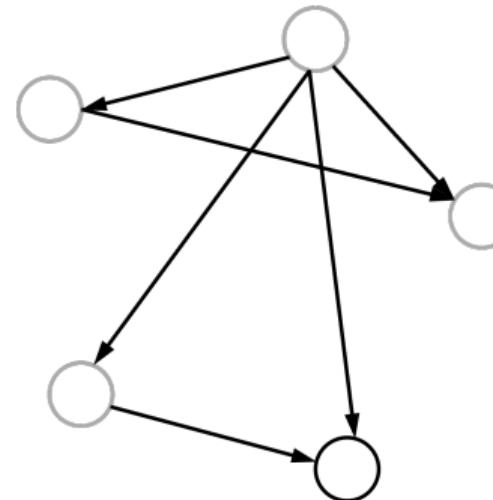




What is a Graph?



# What is a Graph?

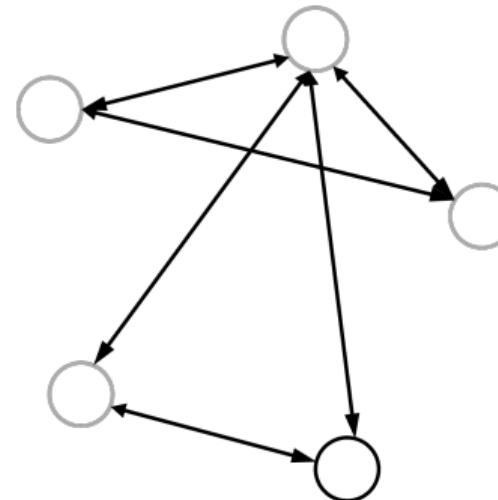




What is a Graph?



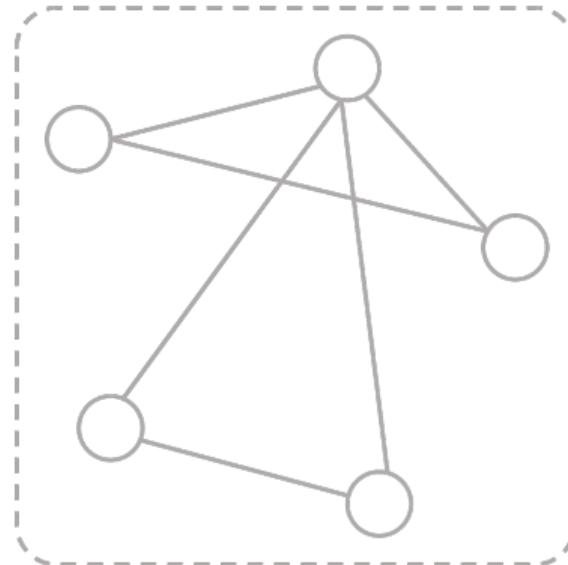
# What is a Graph?





What is a Graph?

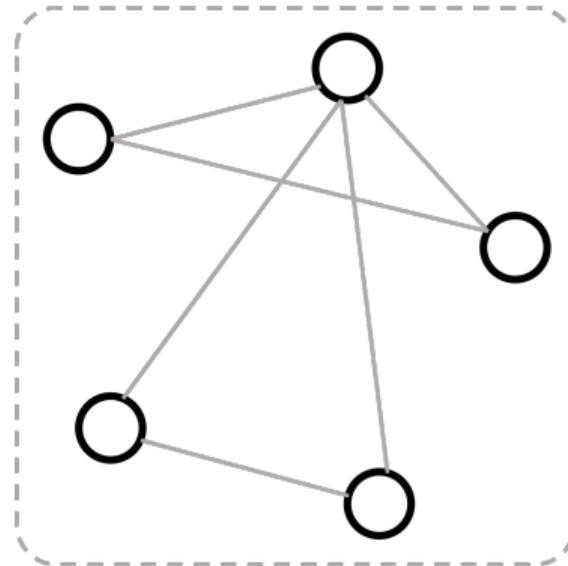
# Attributes of Graph





What is a Graph?

# Attributes of Graph



## V Vertex (or node) attributes

e.g., node identity, number of neighbors

## E Edge (or link) attributes and directions

e.g., edge identity, edge weight

## U Global (or master node) attributes

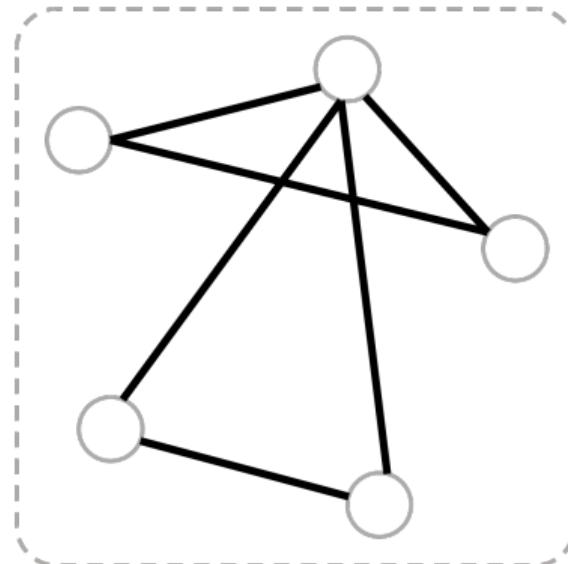
e.g., number of nodes, longest path





What is a Graph?

# Attributes of Graph



- V Vertex (or node) attributes**  
e.g., node identity, number of neighbors
- E Edge (or link) attributes and directions**  
e.g., edge identity, edge weight

- U Global (or master node) attributes**  
e.g., number of nodes, longest path

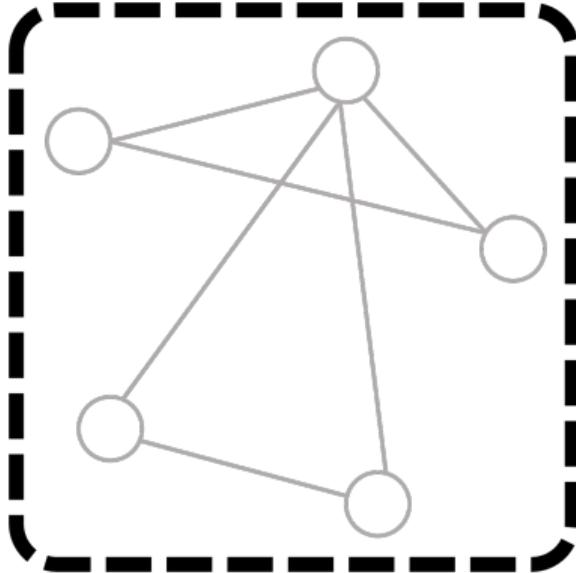




What is a Graph?



# Attributes of Graph



## V Vertex (or node) attributes

e.g., node identity, number of neighbors

## E Edge (or link) attributes and directions

e.g., edge identity, edge weight

## U Global (or master node) attributes

e.g., number of nodes, longest path

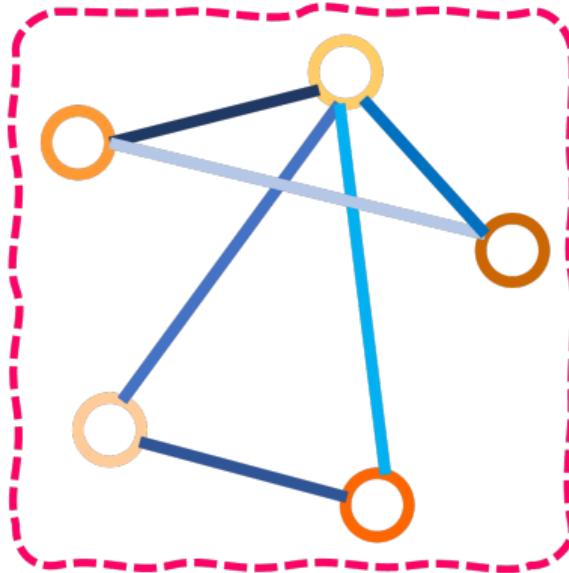




What is a Graph?



# Attributes of Graph



**V** Vertex (or node) embedding

e.g., node identity, number of neighbors

**E** Edge (or link) embedding

e.g., edge identity, edge weight

**U** Global (or master node) embedding

e.g., number of nodes, longest path





Where to Find them?



**Graphs are all around us;** real world objects are often defined in terms of their connections to other things.





## Where to Find them?

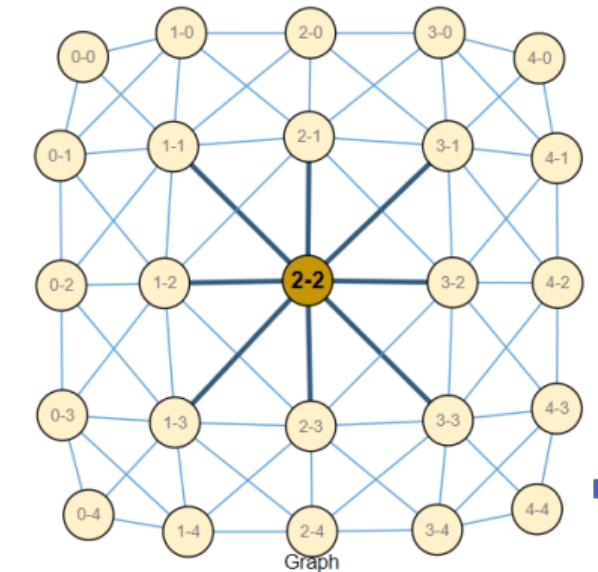
# Images as Graphs

|     |     |            |     |     |
|-----|-----|------------|-----|-----|
| 0-0 | 1-0 | 2-0        | 3-0 | 4-0 |
| 0-1 | 1-1 | 2-1        | 3-1 | 4-1 |
| 0-2 | 1-2 | <b>2-2</b> | 3-2 | 4-2 |
| 0-3 | 1-3 | 2-3        | 3-3 | 4-3 |
| 0-4 | 1-4 | 2-4        | 3-4 | 4-4 |

## Image Pixels



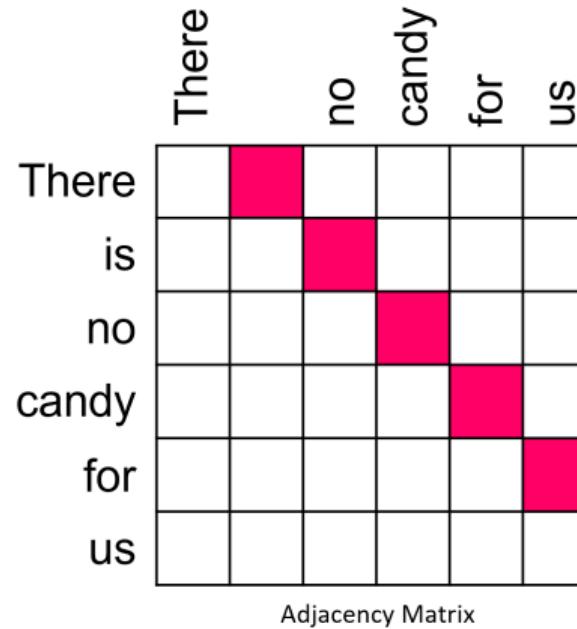
## Adjacency Matrix



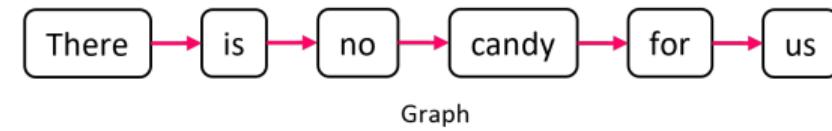


Where to Find them?

# Text as Graphs



**Text:** There is no candy for us



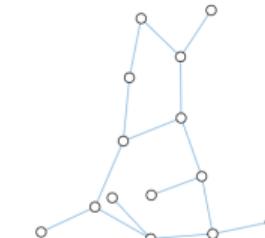
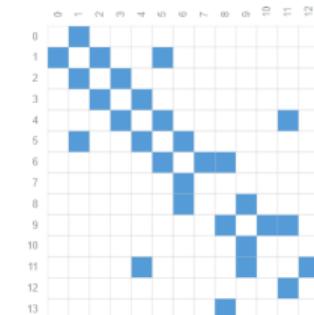
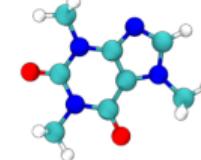


Where to Find them?



# Graph-valued Data in the Wild

- Molecules



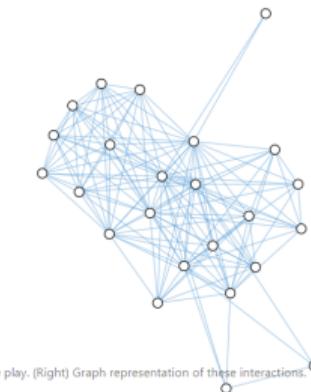


Where to Find them?



# Graph-valued Data in the Wild

- Molecules
- Social networks



(Left) Image of a scene from the play "Othello". (Center) Adjacency matrix of the interaction between characters in the play. (Right) Graph representation of these interactions.

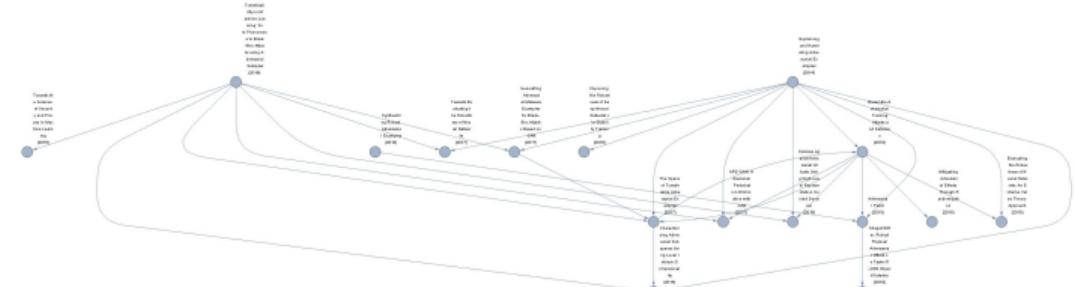




## Where to Find them?

## Graph-valued Data in the Wild

- Molecules
  - Social networks
  - Citation networks





Where to Find them?



# Graph-valued Data in the Wild

- Molecules
- Social networks
- Citation networks
- etc

| Dataset                  | Domain           | Edges per node (degree) |        |        |     |       |     |
|--------------------------|------------------|-------------------------|--------|--------|-----|-------|-----|
|                          |                  | graphs                  | nodes  | edges  | min | mean  | max |
| karate club              | Social network   | 1                       | 34     | 78     |     | 4.5   | 17  |
| qm9                      | Small molecules  | 134k                    | ≤ 9    | ≤ 26   | 1   | 2     | 5   |
| Cora                     | Citation network | 1                       | 23,166 | 91,500 | 1   | 7.8   | 379 |
| Wikipedia links, English | Knowledge graph  | 1                       | 12M    | 378M   |     | 62.24 | 1M  |

Summary statistics on graphs found in the real world. Numbers are dependent on featurization decisions. More useful statistics and graphs can be found in KONECT[14].





What Tasks to Perform on Graphs?



# What types of problems have graph structured data?





What Tasks to Perform on Graphs?

# Graph-level Task

- Predicting the property of an entire graph





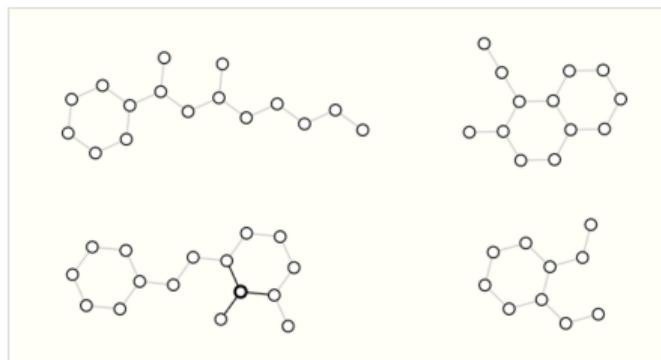
What Tasks to Perform on Graphs?



# Graph-level Task

- Predicting the property of an entire graph

- Input:** graphs

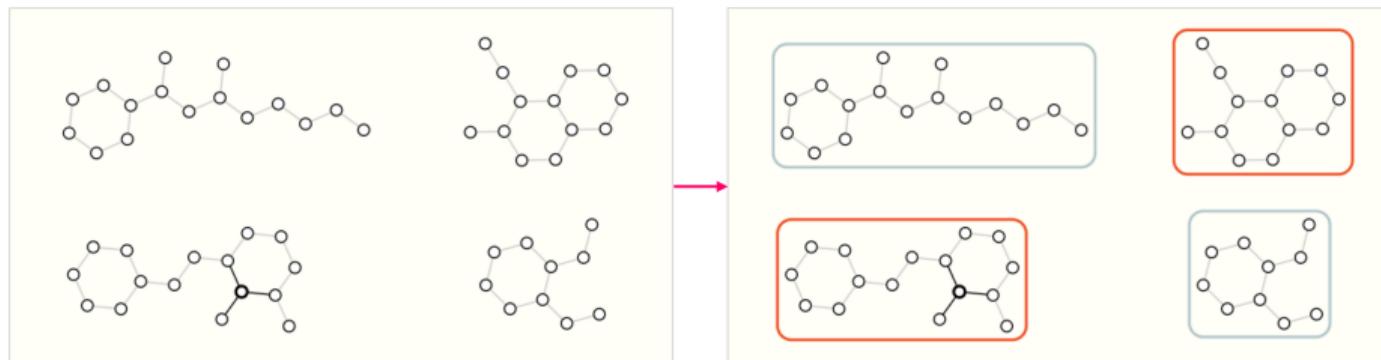




What Tasks to Perform on Graphs?

# Graph-level Task

- Predicting the property of an entire graph





What Tasks to Perform on Graphs?

# Graph-level Task

- Predicting the property of an entire graph
  - the smell of the molecular
  - image classification
  - sentiment analysis of text





What Tasks to Perform on Graphs?

## Node-level Task

- Predicting the identity or role of each node





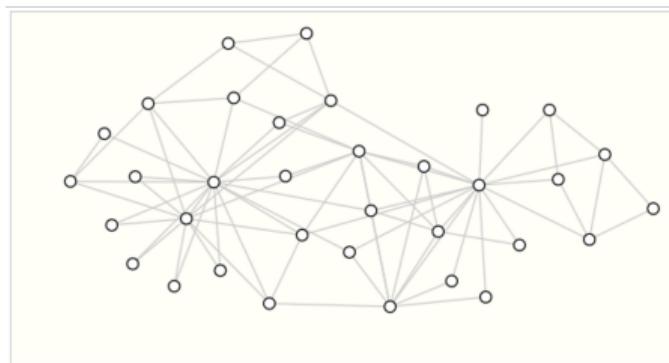
What Tasks to Perform on Graphs?



# Node-level Task

- Predicting the identity or role of each node

- **Input:** graphs (unlabeled nodes)

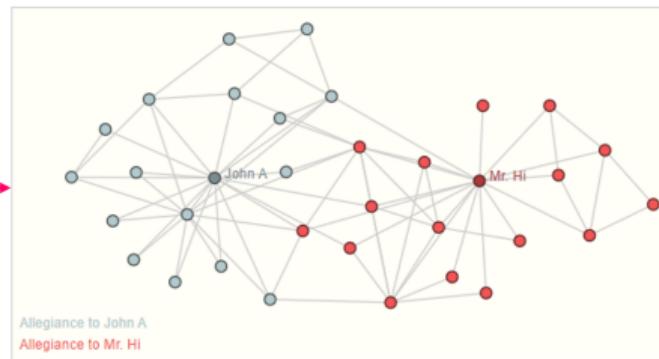
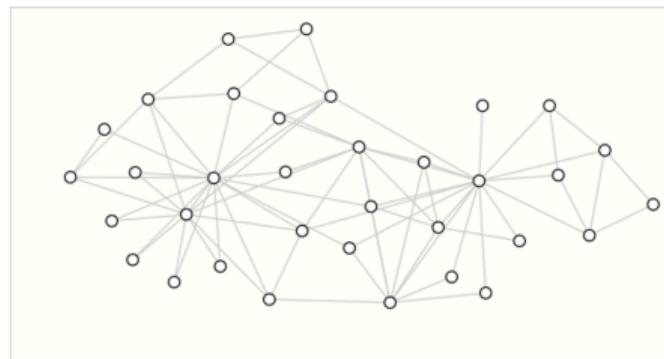




What Tasks to Perform on Graphs?

# Node-level Task

- Predicting the identity or role of each node





What Tasks to Perform on Graphs?

# Node-level Task

- Predicting the identity or role of each node
  - image segmentation
  - PoS of each word



- **Input:** graphs (unlabeled nodes)
- **Output:** graph node labels





What Tasks to Perform on Graphs?

## Edge-level Task

- Predicting the identity or role of each edge

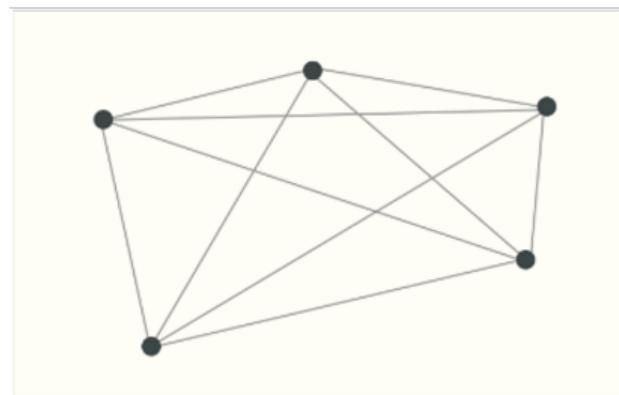




What Tasks to Perform on Graphs?

# Edge-level Task

- Predicting the identity or role of each edge

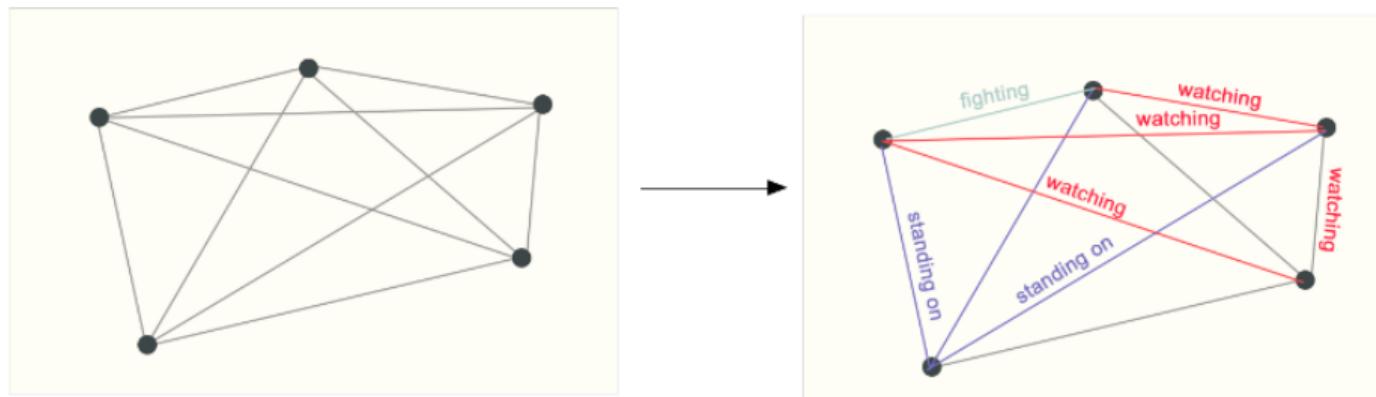




What Tasks to Perform on Graphs?

# Edge-level Task

- Predicting the identity or role of each edge





What Tasks to Perform on Graphs?

## Edge-level Task

- Predicting the identity or role of each edge
  - science understanding
- **Input:** graphs (unlabeled edges)
- **Output:** labels for edges





What Tasks to Perform on Graphs?

## Edge-level Task



nVn

GNNs



Summer 2023

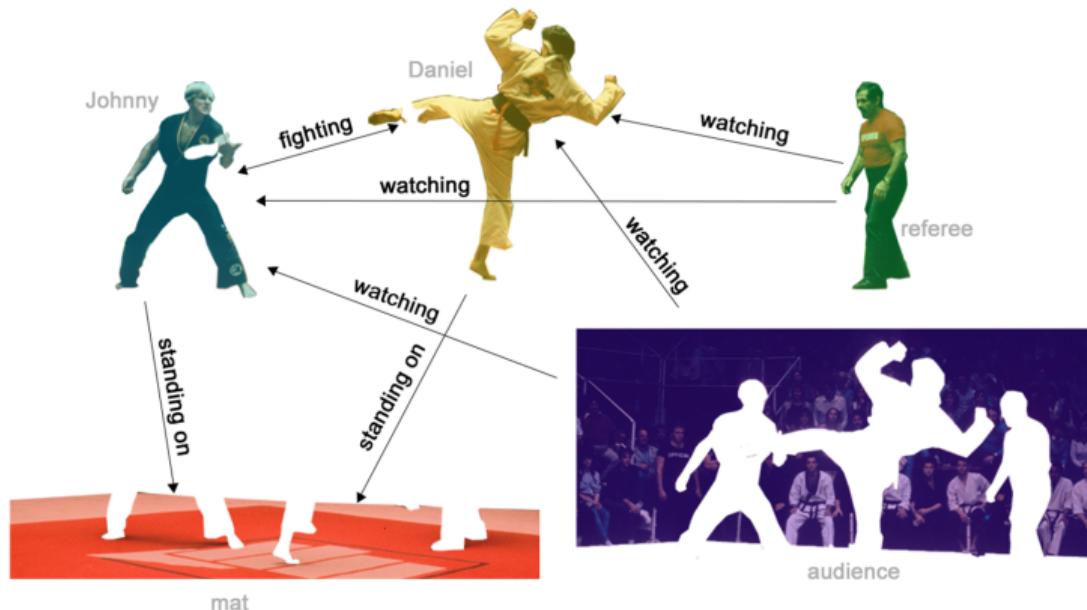




What Tasks to Perform on Graphs?



# Edge-level Task





The challenges of using graphs in ML



How do we go about solving these different graph tasks with neural networks?





The challenges of using graphs in ML



# How to Represent Graphs

Graphs have 4 types of information:

- nodes,
- edges,
- global-context,
- and connectivity!





The challenges of using graphs in ML



# How to Represent Graphs

Graphs have 4 types of information:

- nodes,
- edges,
- global-context,
- and **connectivity!**

$$N = \{node_i\} \quad [n_{nodes}, node_{dim}]$$

$$E = \{edge_i\} \quad [n_{edges}, edge_{dim}]$$

$$U = \text{master node's embedding} [, global_{dim}]$$





The challenges of using graphs in ML



# Representing Graph's Connectivity

- One elegant way to represent graphs is as **Adjacency lists**: These describe the connectivity of edge  $e_k$  between nodes  $n_i$  and  $n_j$  as a tuple  $(i, j)$  in the  $k$ -th entry of an adjacency list.
  - memory-efficient (even with sparse matrices),
  - permutation invariant.





The simplest GNN



- **GNNs = Graph Neural Networks**
- **A GNN is an optimizable transformation on all attributes of the graph** (i.e., nodes, edges, global-context) **that preserves graph symmetries** (connectivity or permutation invariances)
- **"graph-in, graph-out"** architecture

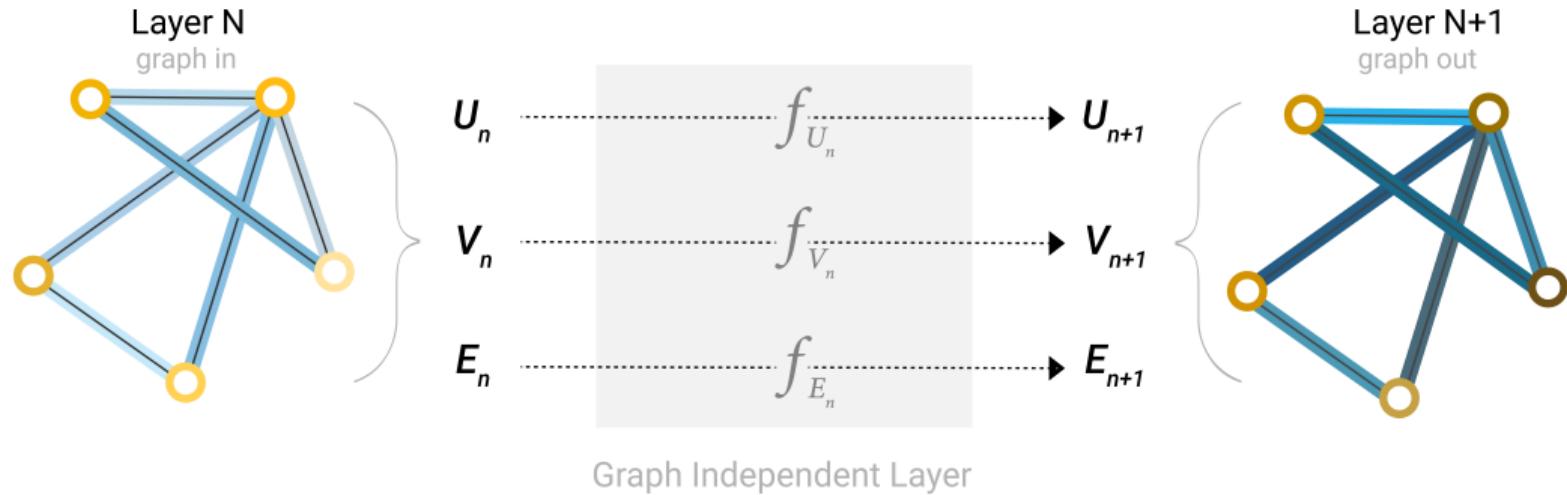




The simplest GNN



# The Simplest GNN



update function  $f = \text{[neural network diagram]}, \dots$

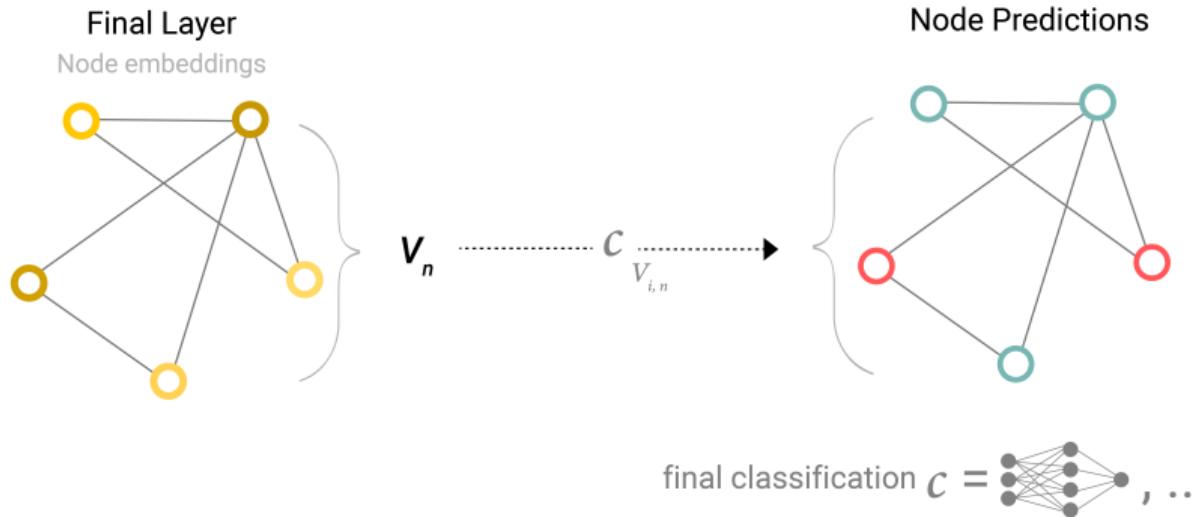




GNN Predictions by Pooling Information



# GNN Predictions





# Pooling Information

- **Pooling:** Collecting information from 1 component and giving them to another.





# Pooling Information

- **Pooling:** Collecting information from 1 component and giving them to another.
  - ➊ For each item to be pooled, *gather* each of their embeddings and concatenate them into a matrix.





# Pooling Information

- **Pooling:** Collecting information from 1 component and giving them to another.

- ① For each item to be pooled, *gather* each of their embeddings and concatenate them into a matrix.
- ② The gathered embeddings are then *aggregated*, using via a operation (e.g., sum, mean).





# Pooling Information

- **Pooling:** Collecting information from 1 component and giving them to another.
  - 1 For each item to be pooled, *gather* each of their embeddings and concatenate them into a matrix.
  - 2 The gathered embeddings are then *aggregated*, using via a operation (e.g., sum, mean).
- Abbreviation:  $\rho$  e.g., gathering information from edges to nodes  $\rho_{E_n \rightarrow V_n}$ .

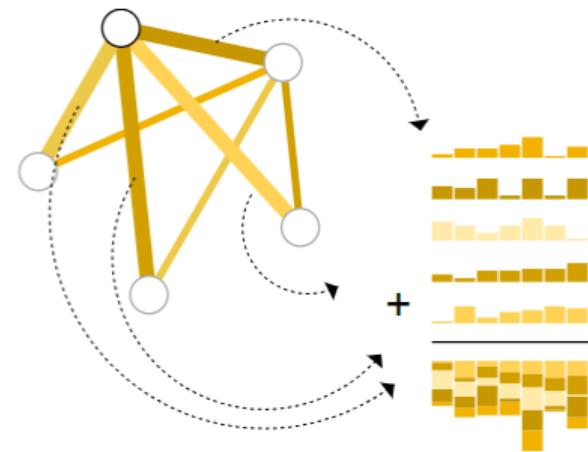




GNN Predictions by Pooling Information



# Pooling Information



Aggregate information  
from adjacent edges

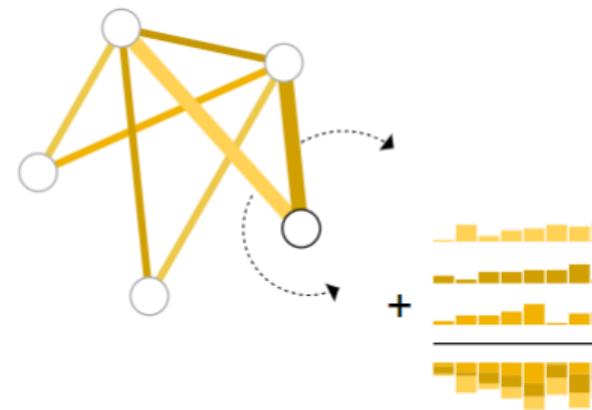




GNN Predictions by Pooling Information



# Pooling Information



Aggregate information  
from adjacent edges

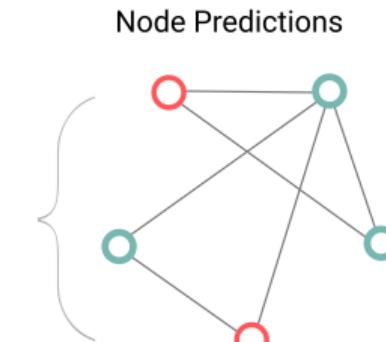
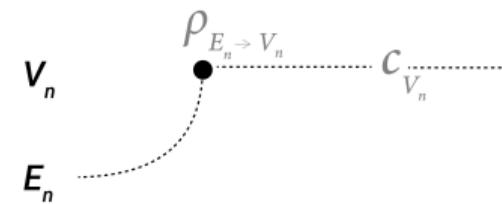
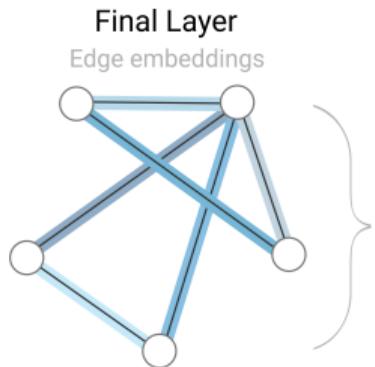




GNN Predictions by Pooling Information



# GNN Predictions by Pooling Information



pooling function  $\rho$   
final classification  $c = \dots$

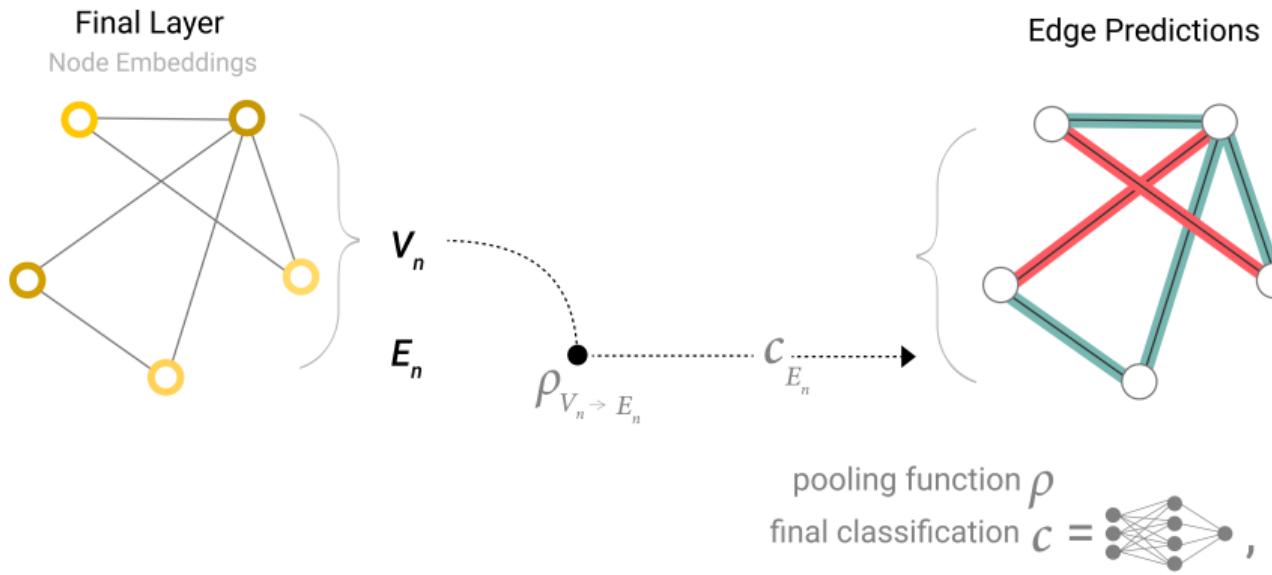




GNN Predictions by Pooling Information



# GNN Predictions by Pooling Information

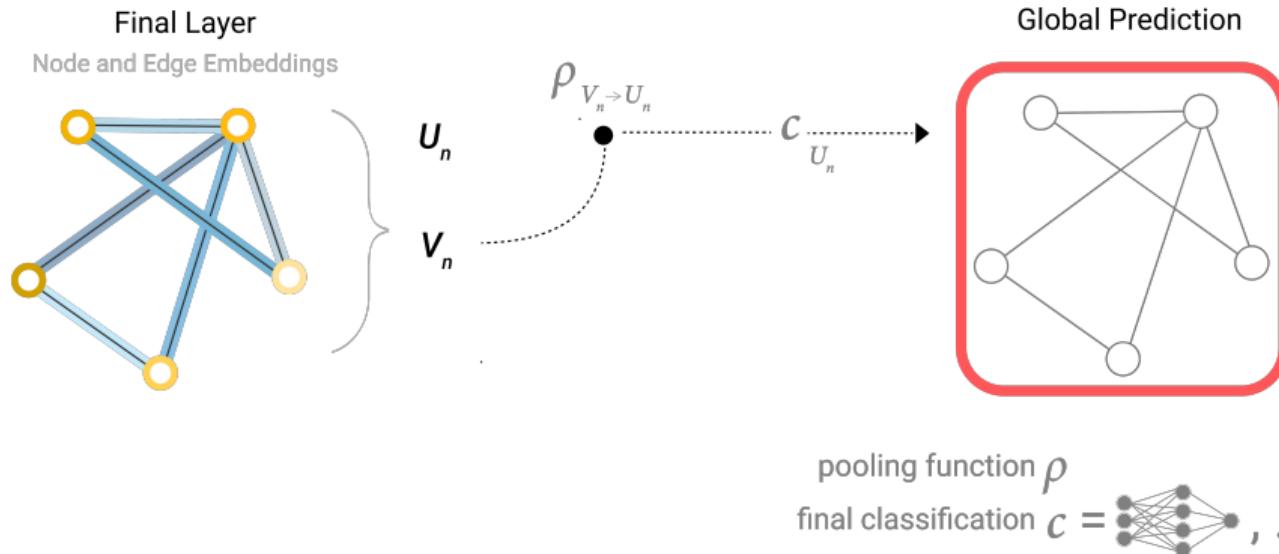




GNN Predictions by Pooling Information



# GNN Predictions by Pooling Information

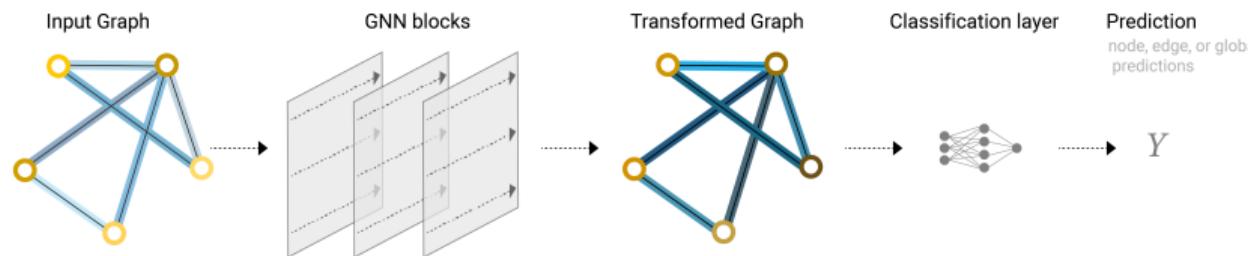




GNN Predictions by Pooling Information



# End2End Prediction Task with a GNN Model





GNN Predictions by Pooling Information



# End2End Prediction Task with a GNN Model

- We still don't use the connectivity of the graph
  - Each component is processed independently
  - ONLY use connectivity when prediction





Passing messages between parts of the graph



...What if we try using pooling inside GNN Layers?





Passing messages between parts of the graph



...What if we try using pooling inside GNN Layers?

Or to make our learned embeddings aware of graph **connectivity**?





Passing messages between parts of the graph



# Message Passing

- **Message Passing:**

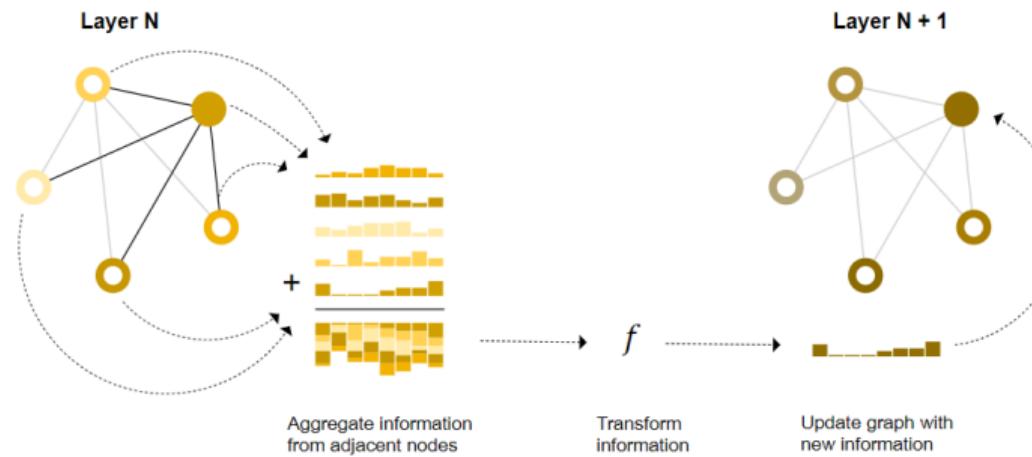
- ① For each node in the graph, *gather* all the neighboring node embedding (or messages), which is the  $g$  function described above.
  - ② *Aggregate* all messages via an aggregate function (e.g, sum).
  - ③ All pooled messages are passed through an *update function*, usually a learned neural network.
- Just as pooling can be applied to either nodes or edges, message passing can occur between either nodes or edges.





Passing messages between parts of the graph

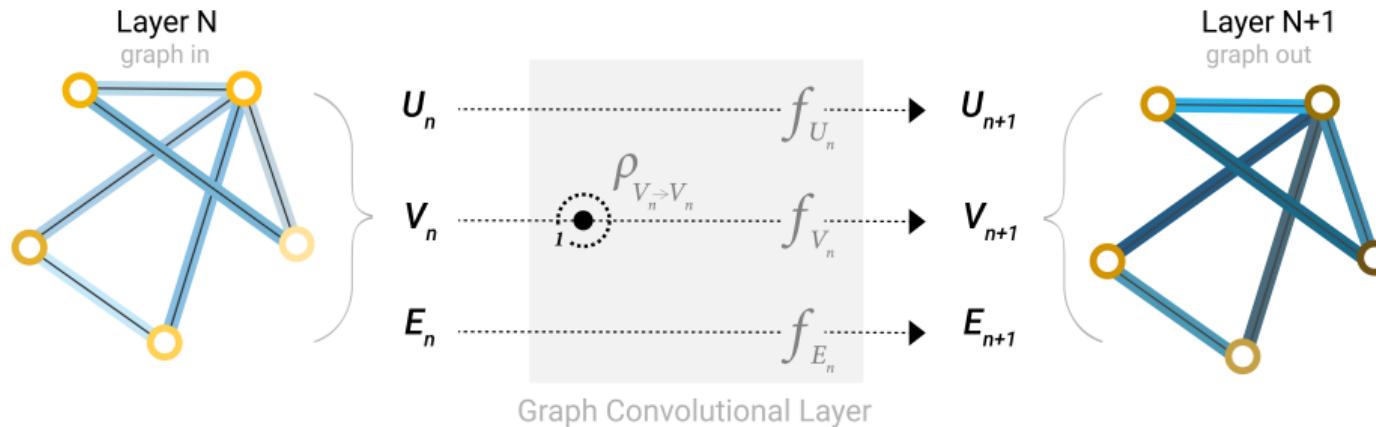
# Message Passing





Passing messages between parts of the graph

# GCN architecture



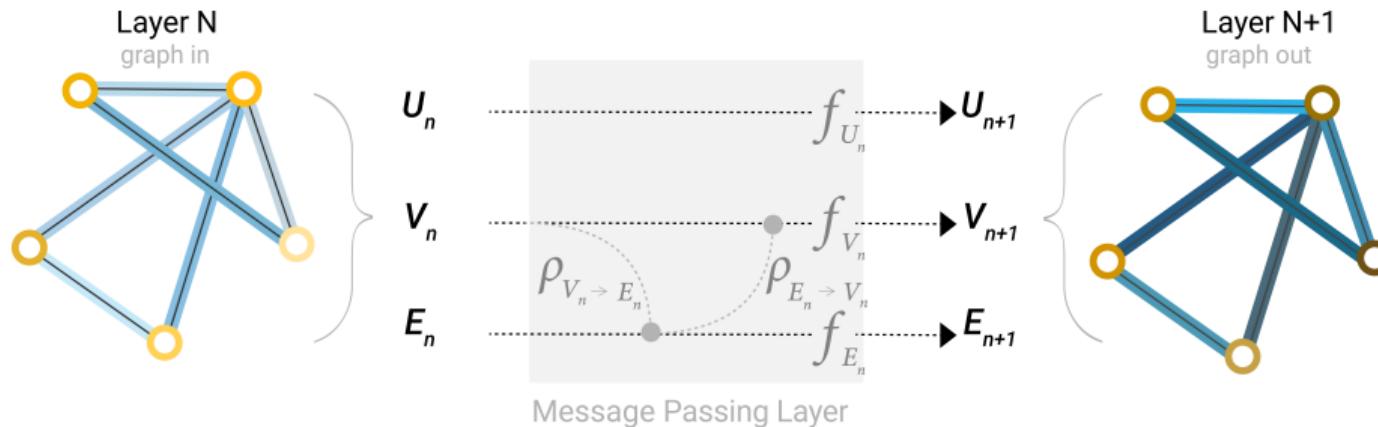
update function  $f = \dots$ , ...  
 pooling function  $\rho$





Passing messages between parts of the graph

# Learning Edge Representations



update function  $f = \text{[neural network]}, \dots$   
 pooling function  $\rho$





Passing messages between parts of the graph



# Adding Global Representations

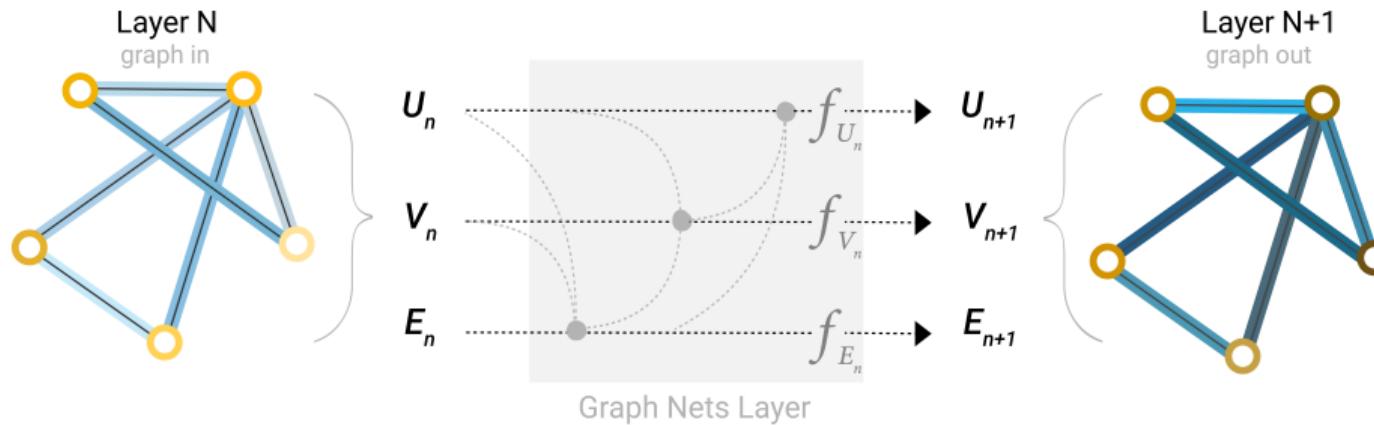
- So far, there is still one flaw:
  - Nodes that far away from each other in the graph may never be able to efficiently transfer information to one another, even which several layers.
  - e.g., if we have  $k$ -layers, information will propagate at most  $k$ -steps away.
- One solution is by using the global representations or **master node**.





## Passing messages between parts of the graph

# Adding Global Representations



update function  $f =$   , ...  
 pooling function  $p$

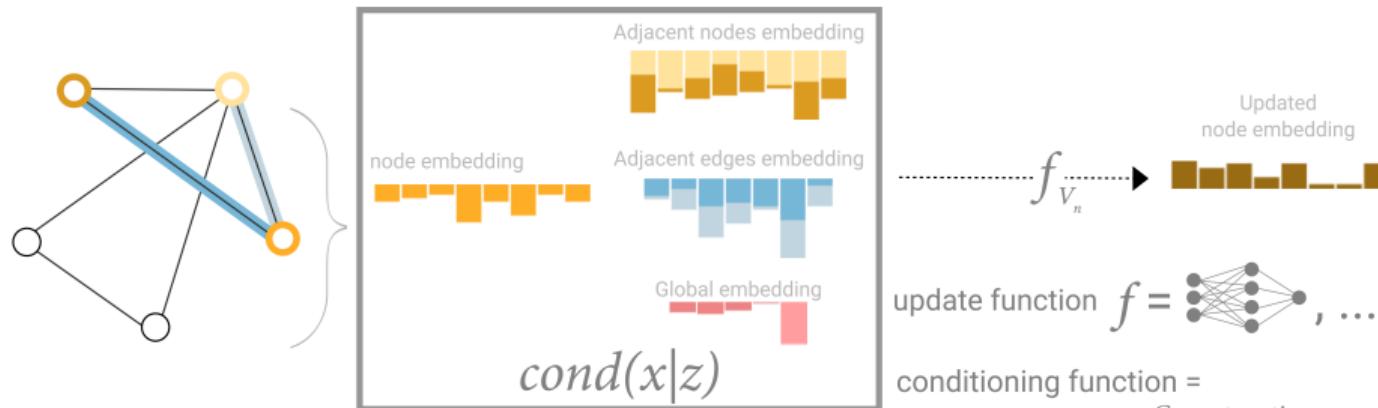




Passing messages between parts of the graph



# Adding Global Representations





Other types of graphs



# References I

