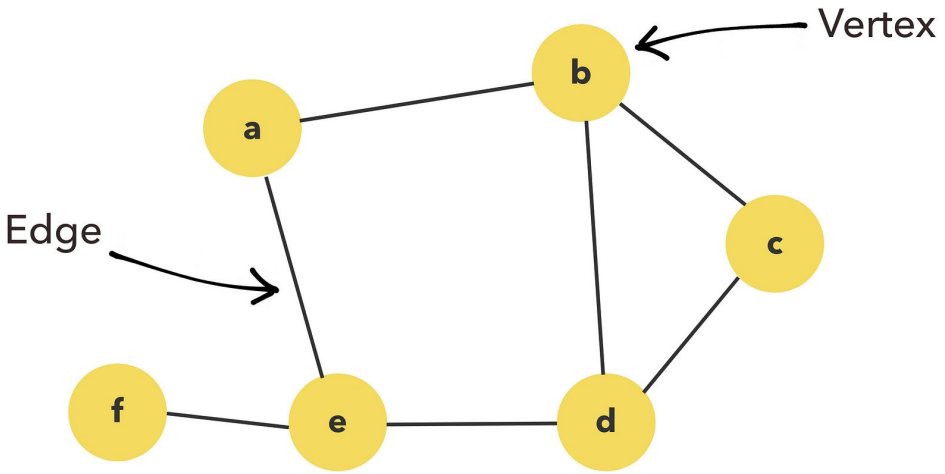
TEXAS
The University of Texas at Austin

Graph Neural Networks

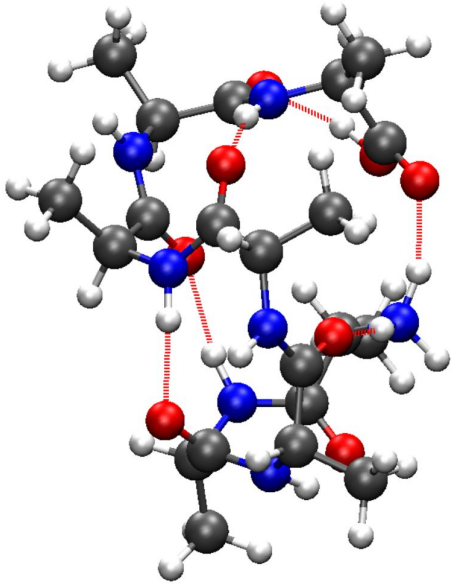
Sicong Che
Kate Zhang
11/10/2024

What is a Graph?

- G = (V, E)
- Global attributes
- Very flexible
- Good for modeling complex relationships

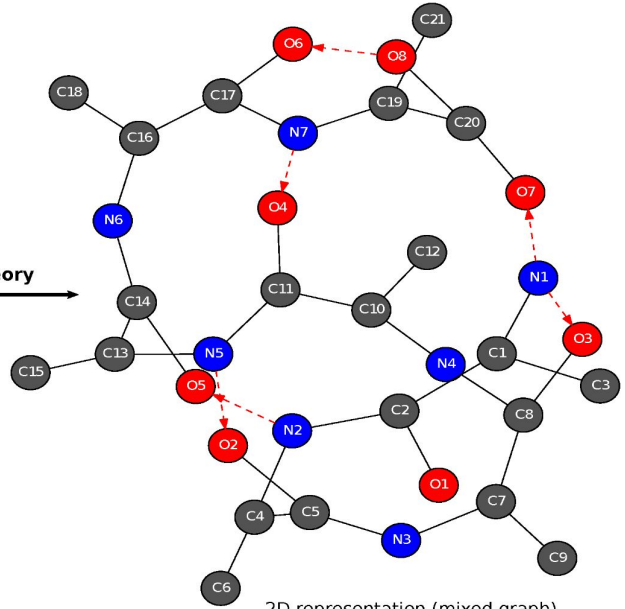


Graphs in the Real World

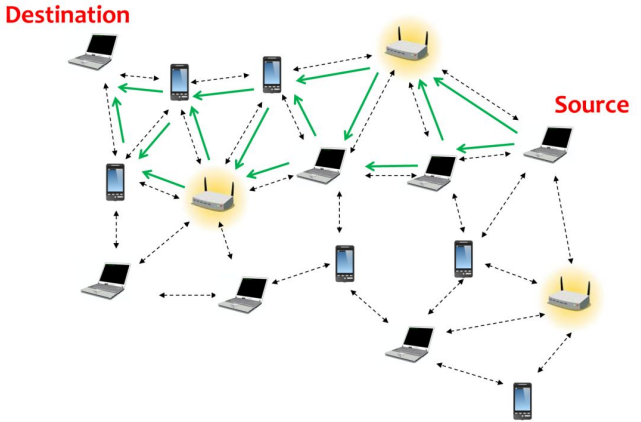


3D representation (position)

Graph theory

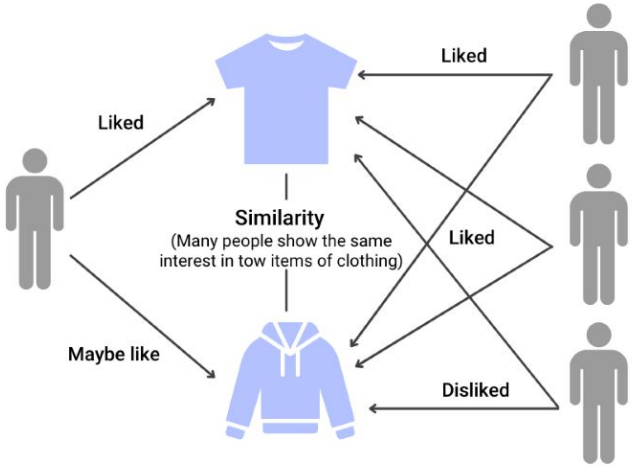


2D representation (mixed graph)



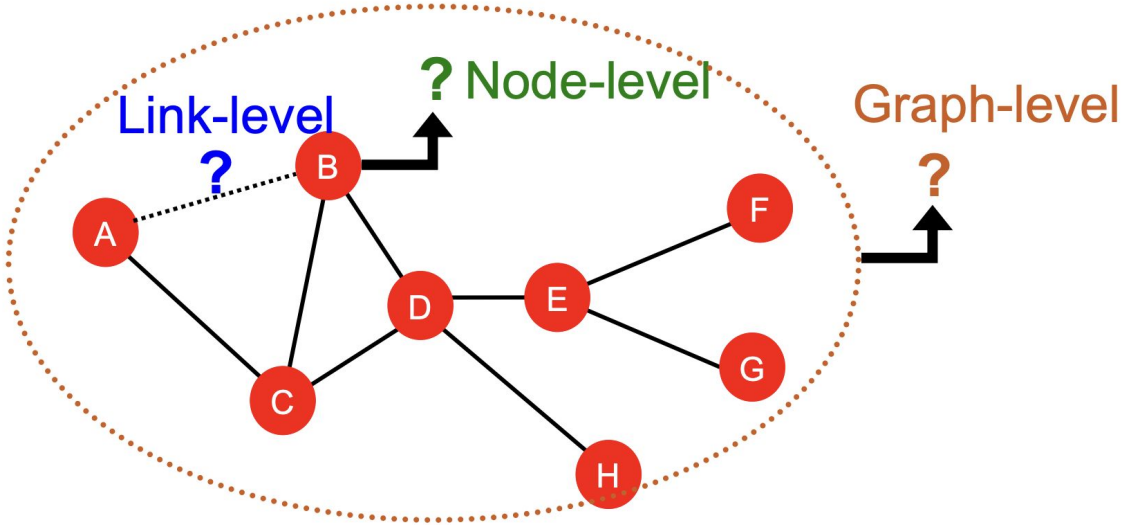
What is a GNN? Why do we need GNNs?

- Neural networks that operate on graph data
- Can explore huge graphs
- Traditional neural nets struggle with unstructured data
- Applications:
  - Drug discovery
  - Fraud detection
  - Recommender systems



Graph Prediction Problems

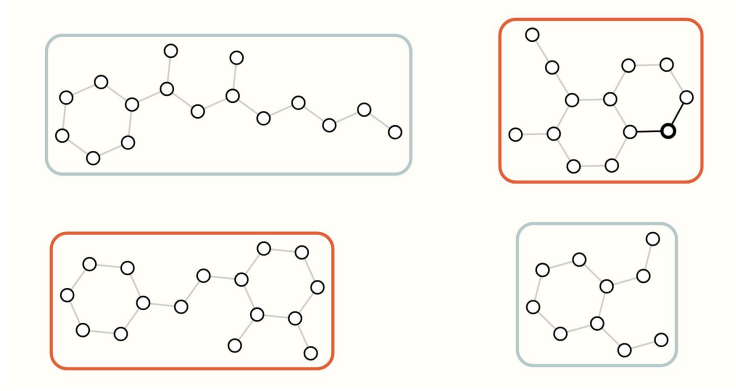
- Node-level prediction
- Link-level prediction
- Graph-level prediction



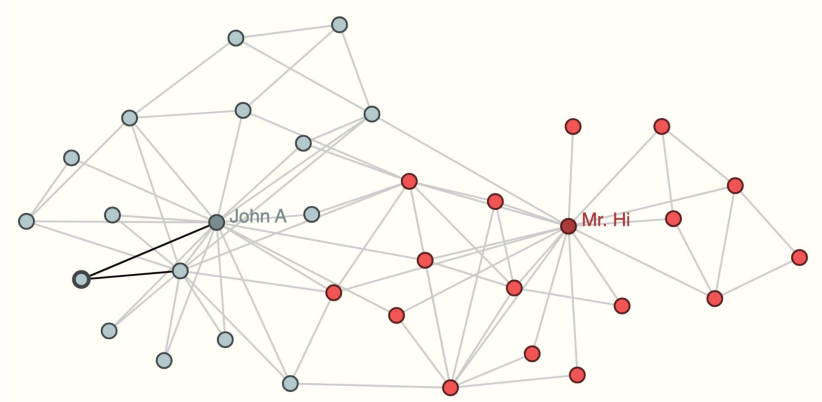
source: https://web.stanford.edu/class/cs224w/slides/01-intro.pdf

Prediction Examples

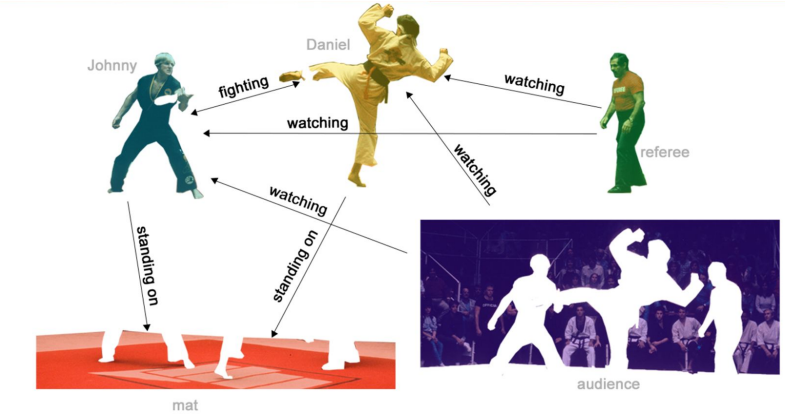
Graph-Level



Node-Level

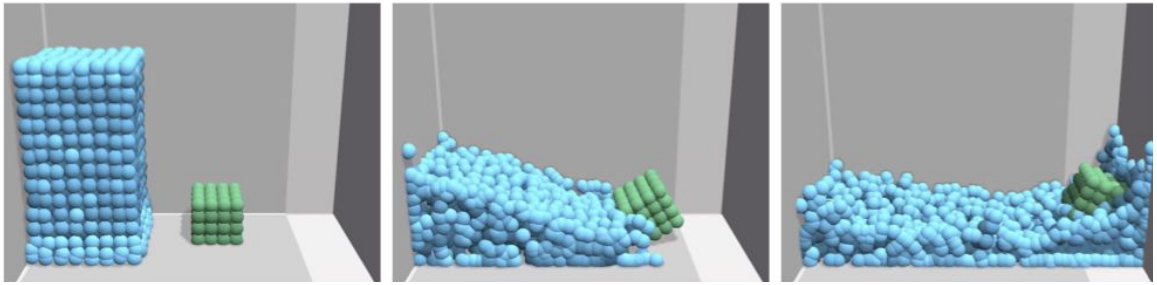


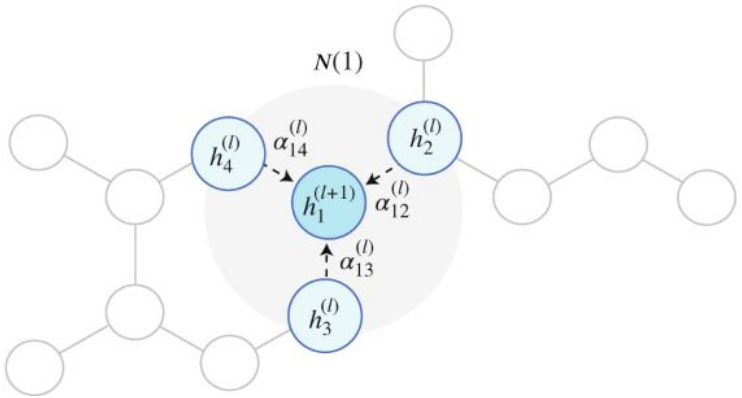
Link-Level



Other Common Tasks

- Clustering
- Graph generation
- Graph evolution





General Architecture of GNN

Goal: graph structure + node features -> representation vector of a node/graph

Main Parts

1. Aggregate the representations of neighbors
2. Readout/Global Pooling

General Architecture of GNN - Part 1

1.Aggregate the representations of neighbors

embedding of node v

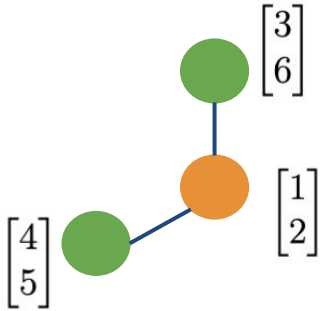
$$a_v = \text{AGGREGATE}\left(\{h_u : u \in \mathcal{N}(v)\}\right), \quad h_v = \text{COMBINE}^{(k)}\left(h_v, a_v\right),$$

embedding of node u, node u are the neighbors of node v

For example:

AGGREGATE: $a_v = \max\left(\{\text{ReLU}\left(W \cdot h_u\right) \mid \forall u \in \mathcal{N}(v)\}\right),$

COMBINE: $W \cdot \begin{bmatrix} h_v & a_v \end{bmatrix}$



Assume that W assigns a weight of 1 to all values for simplicity.
AGGREGATE: ReLU([3,6]) = [3, 6]; ReLU([4,5]) = [4, 5]; so a_v = [4,6]
COMBINE: Linear mapping

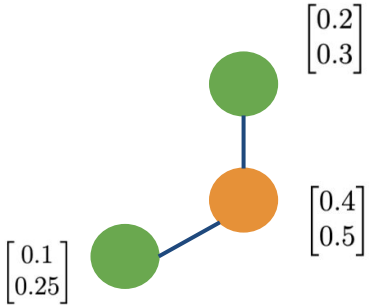
General Architecture of GNN - Part 2

2.Readout/Global Pooling

$$h_G = \mathrm{READOUT}\left(\{h_v \mid v \in G\}\right).$$

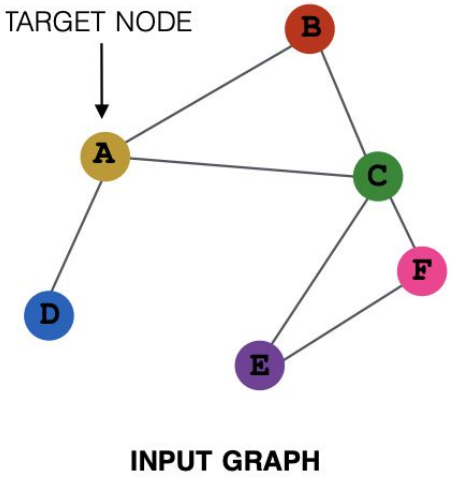
For example: sum global pooling

hG = sum_pooling([0.2, 0.3], [0.1, 0.25], [0.4, 0.5]) = [0.7, 1.05]

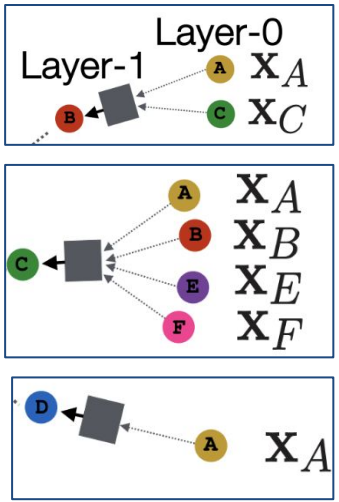


Layers of GNN

Typically, Neural Networks have many layers. What are the layers in a GNN like?



INPUT GRAPH



update the embedding of node B

update the embedding of node C

update the embedding of node C

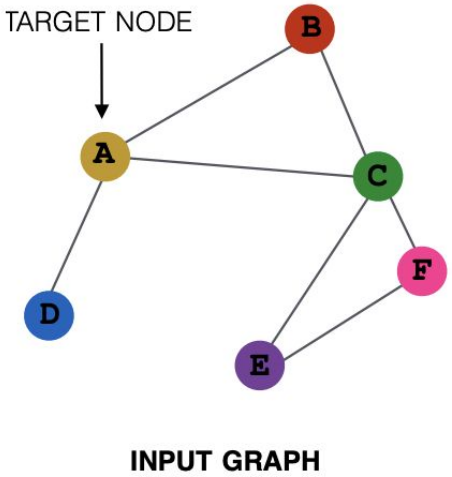
... ...

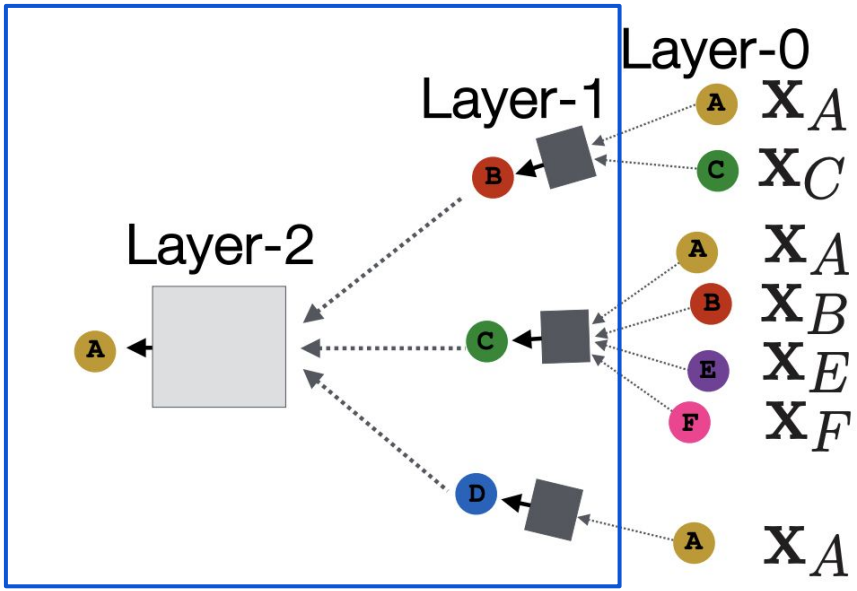
After the first update, we can obtain the embeddings for all nodes at layer 1.

source:https://web.stanford.edu/class/cs224w/slides/03-GNN1.pdf

Layers of GNN

Typically, Neural Networks have many layers. What are the layers in a GNN like?





source:https://web.stanford.edu/class/cs224w/slides/03-GNN1.pdf

Intuition from CNNs

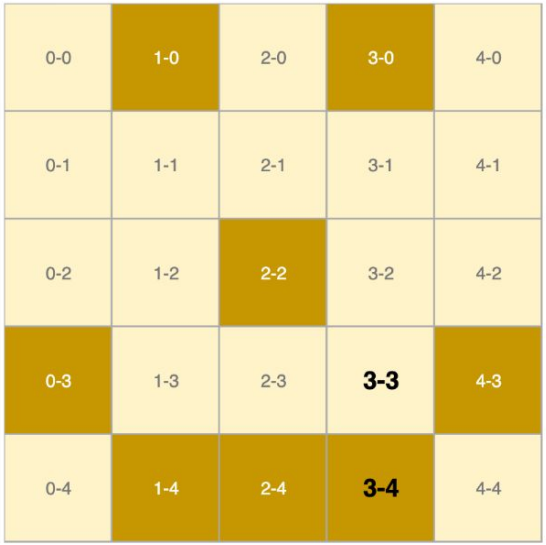
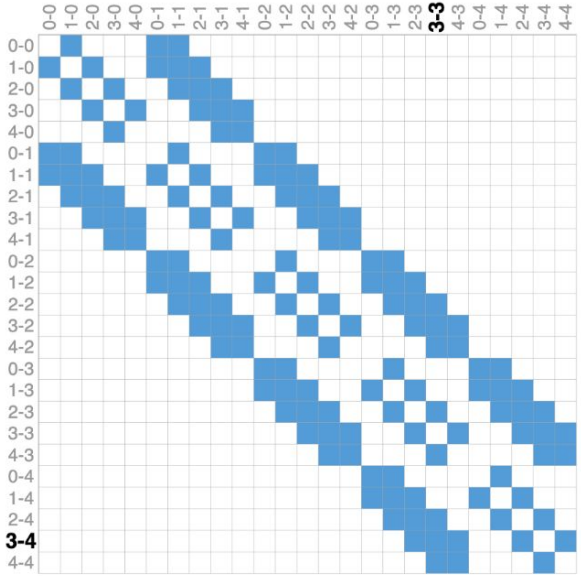
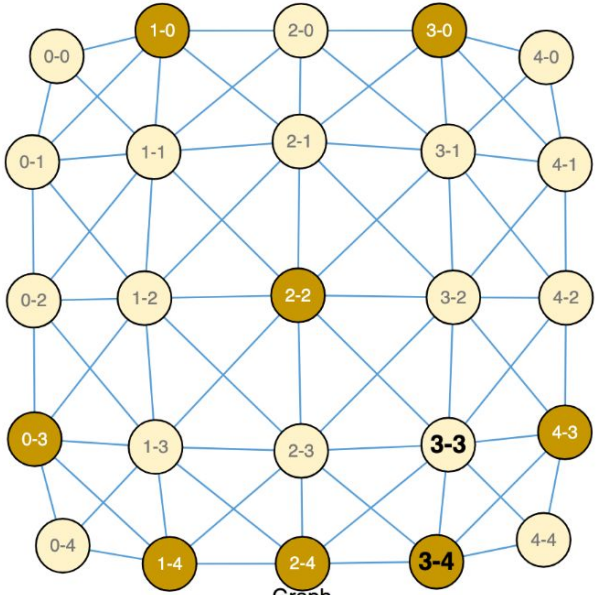


Image Pixels



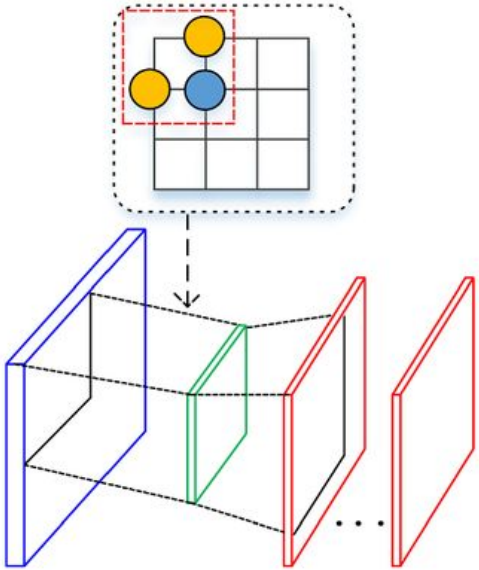
Adjacency Matrix



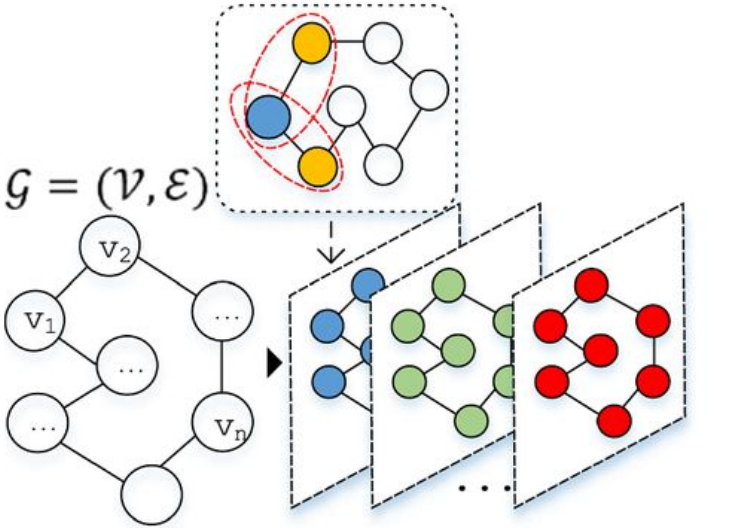
Graph

CNNs vs. GNNs

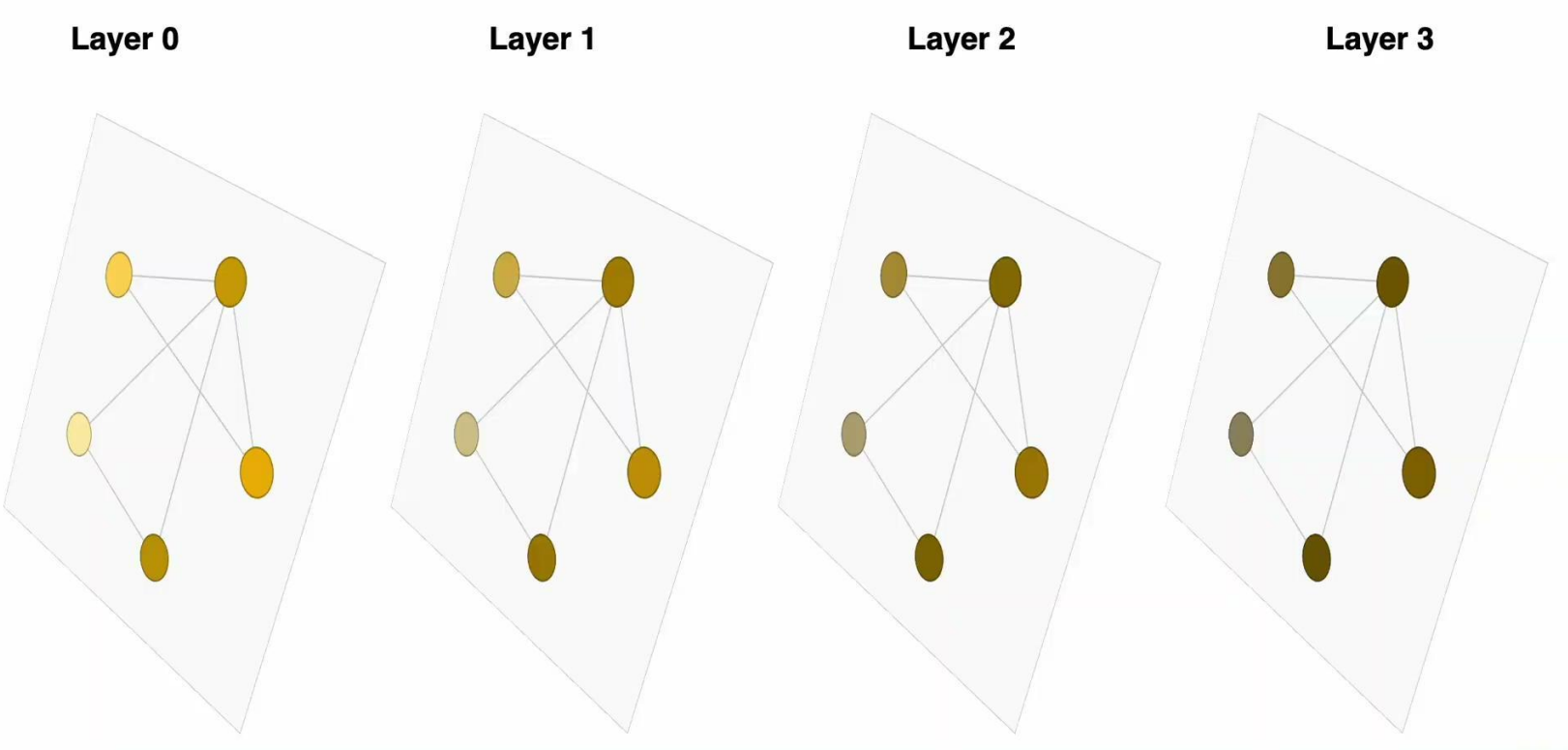
Convolution operation over grid-based structure



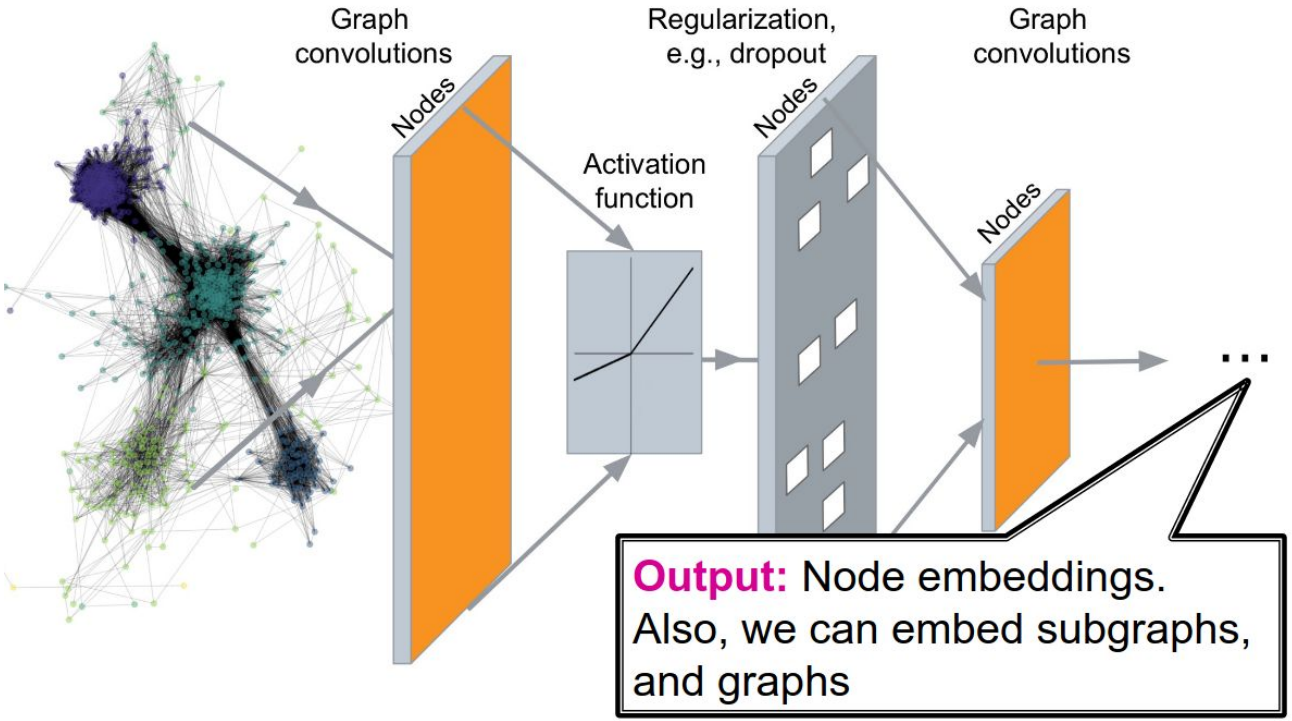
Convolution operation over graph-based structure



Message Passing



Layers of GNN



Classic GNN Model - GCN(Graph Convolutional Network)

Basic approach: **Average** neighbor messages and apply a nonlinear function

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

The 0-layer node embeddings are the same as the node features.

$$\mathbf{h}_v^{(k+1)} = \sigma\left(W_k \sum_{u \in \mathcal{N}(v)} \frac{\mathbf{h}_u^{(k)}}{|\mathcal{N}(v)|} + B_k \mathbf{h}_v^{(k)}\right), \quad \forall k \in \{0, \ldots, K-1\}$$

Classic GNN Model - GCN

Basic approach: **Average** neighbor messages and apply a nonlinear function

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

$$\mathbf{h}_v^{(k+1)} = \sigma\left(W_k \sum_{u \in \mathcal{N}(v)} \frac{\mathbf{h}_u^{(k)}}{|\mathcal{N}(v)|} + B_k \mathbf{h}_v^{(k)}\right), \quad \forall k \in \{0, \ldots, K-1\}$$

N(v) represents the neighbors of node v.
Average neighbor's previous layer embeddings.

Classic GNN Model - GCN

Basic approach: **Average** neighbor messages and apply a nonlinear function

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

$$\mathbf{h}_v^{(k+1)} = \sigma\left(W_k \sum_{u \in \mathcal{N}(v)} \frac{\mathbf{h}_u^{(k)}}{|\mathcal{N}(v)|} + B_k \mathbf{h}_v^{(k)}\right), \quad \forall k \in \{0, \ldots, K-1\}$$

embedding of v at layer k

Classic GNN Model - GCN

Basic approach: **Average** neighbor messages and apply a nonlinear function

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

$$\mathbf{h}_v^{(k+1)} = \sigma\left(W_k \sum_{u \in \mathcal{N}(v)} \frac{\mathbf{h}_u^{(k)}}{|\mathcal{N}(v)|} + B_k \mathbf{h}_v^{(k)} \right), \quad \forall k \in \{0, \ldots, K-1\}$$

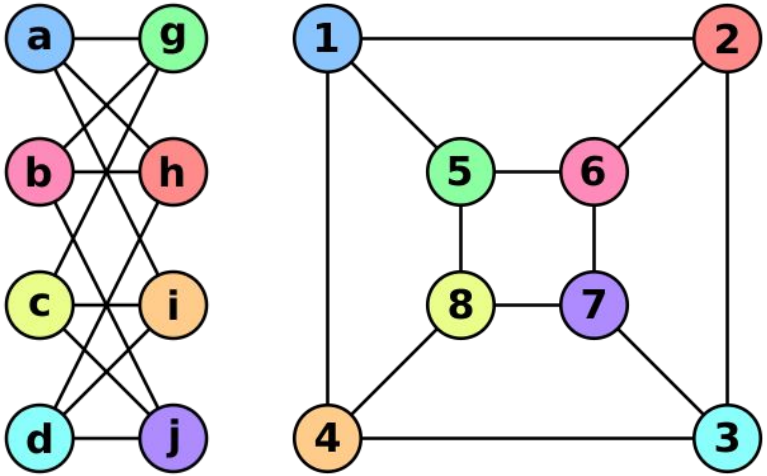
non-linear function, such as ReLU

Although GCN is a good model for graph datasets, its ability to distinguish between different graph structures is limited.

GIN(Graph Isomorphism Network) improves upon GCN, achieving better performance.

Before discussing GIN, let's first talk about what **expressive power** is and the **limitations** of GCN.

Isomorphic Graphs



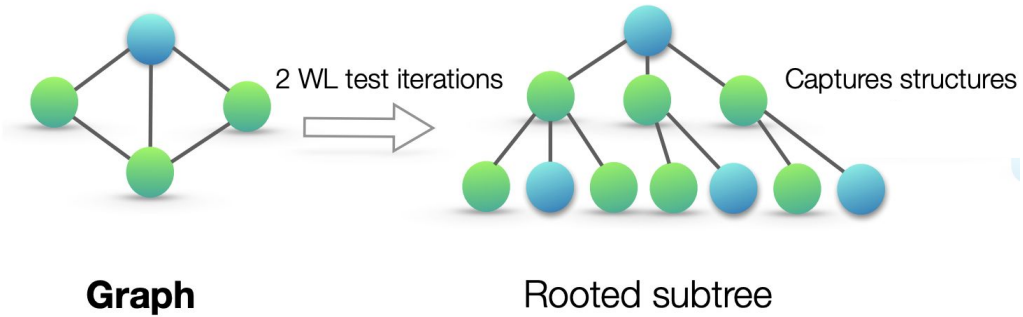
Graphs G and H are isomorphic if there is a bijection between their vertex sets:

$f: V(G) \rightarrow V(H)$

such that vertices u and v of G are adjacent **if and only if** f(u) and f(v) are adjacent in H

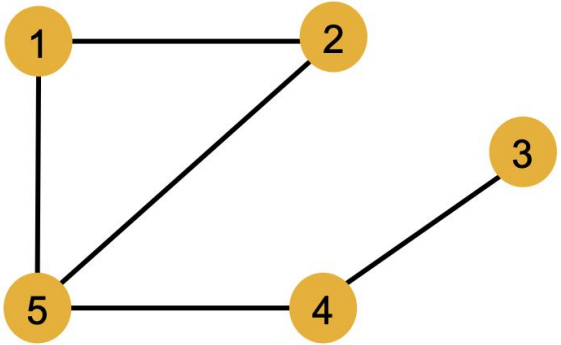
Weisfeiler-Leman Test

- Heuristic to distinguish between non-isomorphic graphs
- Generalization of the color refinement algorithm
- Limitations



Expressive Power of GNN

One aspect of expressive power is distinguishing **different graph structures**.
Specifically, here we consider **local neighborhood structures** around each node in a graph.



Assume all the nodes share the same feature

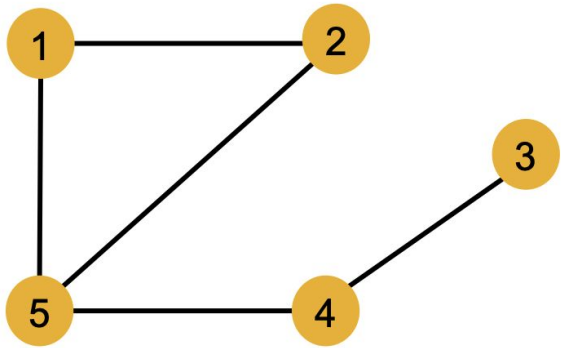
- Node 1 and Node 5 have **different** local neighborhood structure
- Node 1 and Node 2 have the **same** local neighborhood structure.(isomorphism)

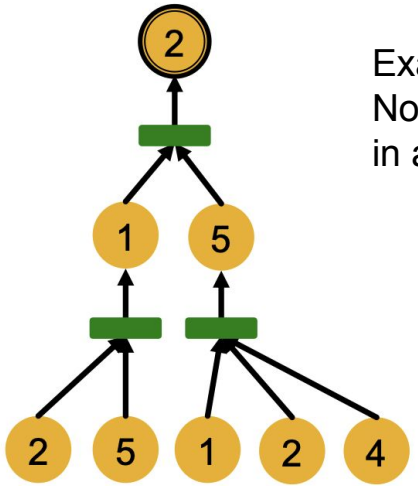
How does a GNN capture local neighborhood structures?

Computational Graph

A GNN generates node embeddings through a computational graph defined by the neighborhood.

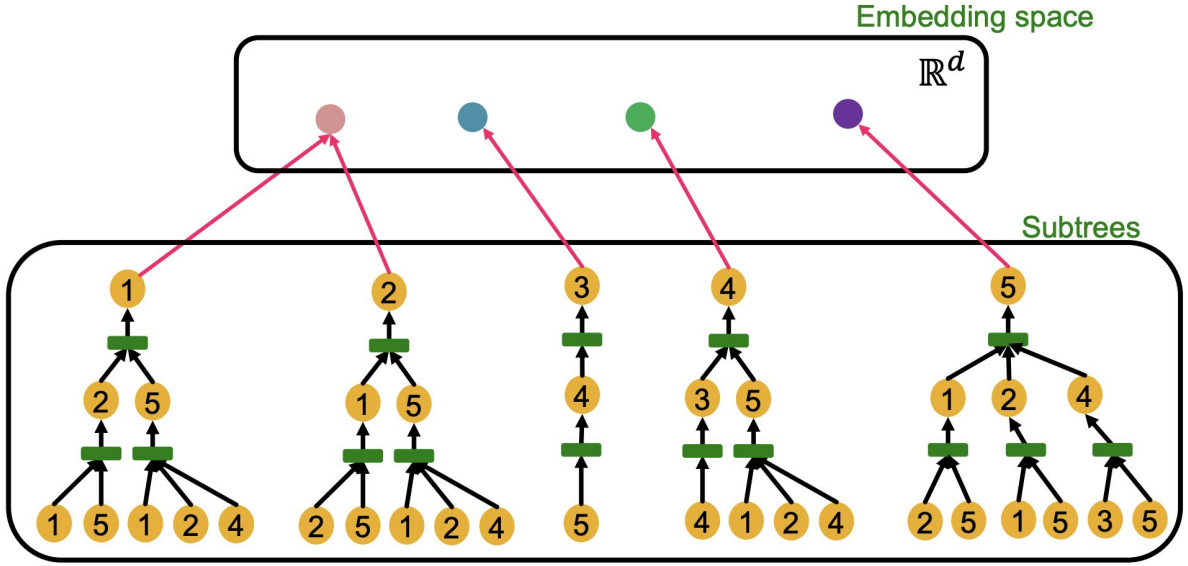
In each layer, a GNN aggregates neighboring node embeddings.





Example:
Node 2's computational graph
in a 2-layer GNN.

Expressive GNN

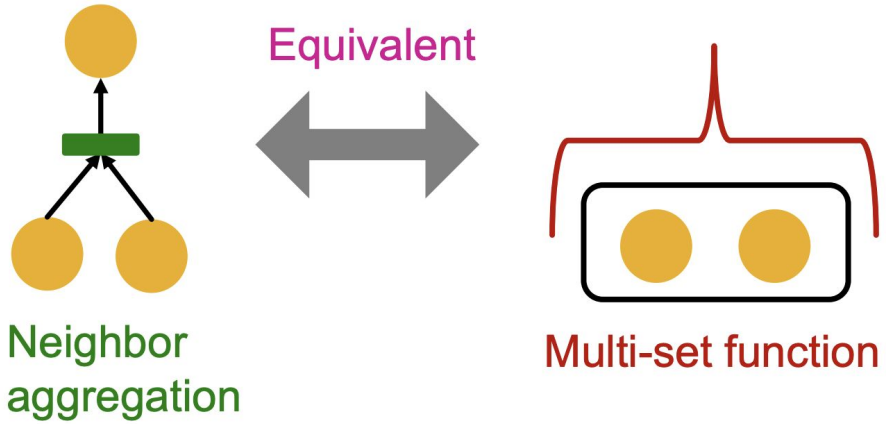


Expressive GNN maps different rooted subtrees(computational graph) into different node embeddings. [injective]

GNN can fully distinguish different subtree structures if every step of its **neighbor aggregation** is injective.

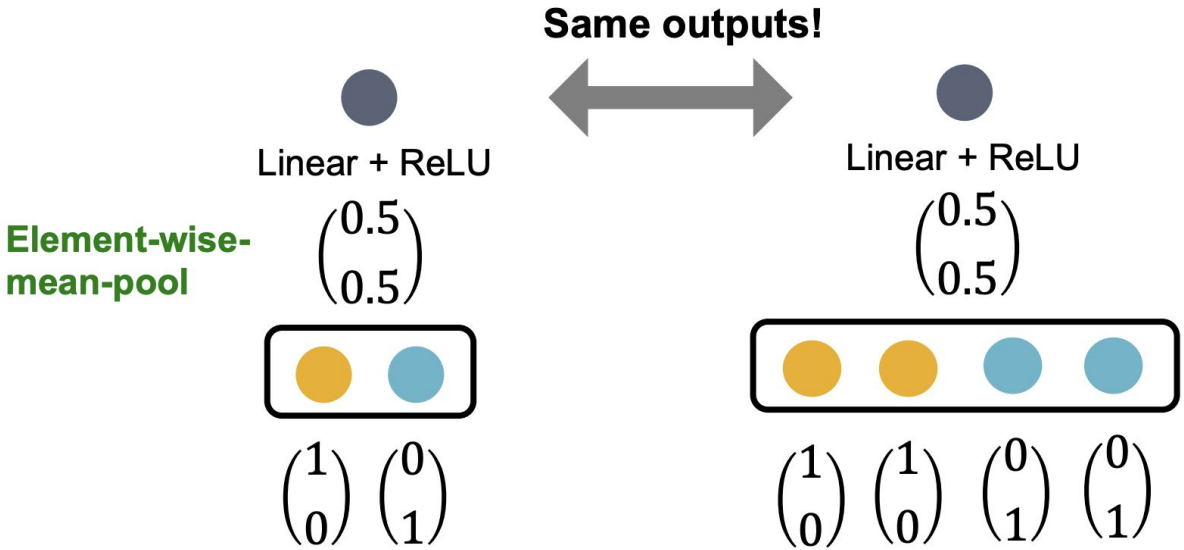
Multi-set function

Neighbor aggregation can be abstracted as a function over a multi-set (a set with repeating elements)



Limitations of GCN

GCN's aggregation function **cannot** distinguish between different multi-sets with the **same color proportion** since it takes **element-wise means**.



Improving GNNs - GIN (Graph Isomorphism Network)

Insight: designing **injective** **neighbor aggregation function** over multisets
How?

Any injective multi-set function can be expressed as:

$$g(X) = \phi\left(\sum_{x \in X} f(x)\right).$$

How to model ϕ and f?

Xu, Keyulu, et al. "How powerful are graph neural networks?." *arXiv preprint arXiv:1810.00826* (2018).

Improving GNNs - GIN (Graph Isomorphism Network)

Insight: designing **injective neighbor aggregation function** over multisets

Universal Approximation Theorem

1-hidden-layer **MLP** with sufficiently-large hidden dimensionality and appropriate non-linearity can approximate any continuous function to an arbitrary accuracy.

So GIN uses MLP to model any injective multiset function.

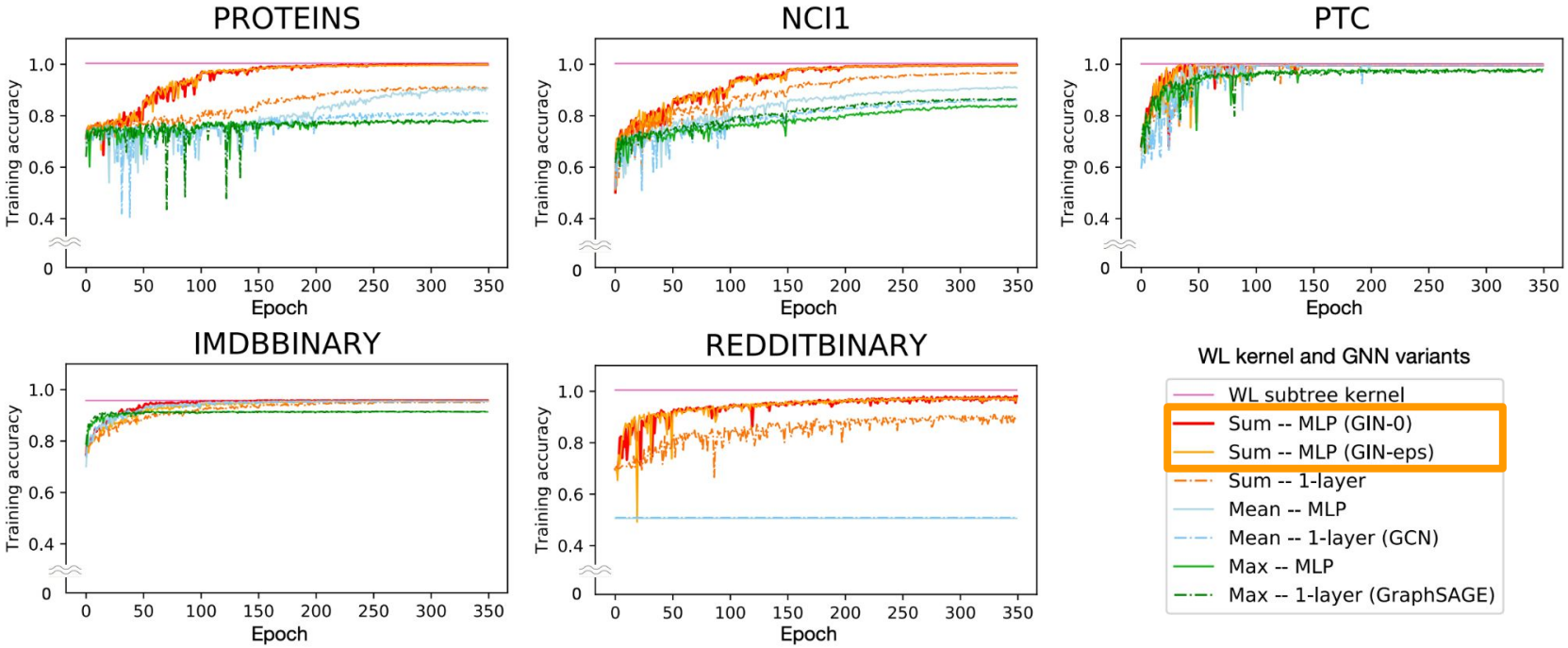
Improving GNNs - GIN (Graph Isomorphism Network)

Insight: designing injective neighbor aggregation function over multisets

Apply MLP, element-wise sum, followed by another MLP.

$$\mathrm{MLP}_{\Phi}\left(\sum_{x \in S} \mathrm{MLP}_{f}(x)\right)$$

GIN Training Performance



ICLR 2019. “How Powerful are Graph Neural Networks?”

GIN Test Set Accuracies

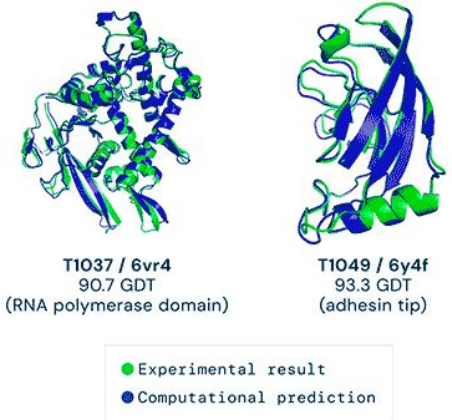
	Datasets	IMDB-B	IMDB-M	RDT-B	RDT-M5K	COLLAB	MUTAG	PROTEINS	PTC	NCI1
Datasets	# graphs	1000	1500	2000	5000	5000	188	1113	344	4110
	# classes	2	3	2	5	3	2	2	2	2
	Avg # nodes	19.8	13.0	429.6	508.5	74.5	17.9	39.1	25.5	29.8
Baselines	WL subtree	73.8 ± 3.9	50.9 ± 3.8	81.0 ± 3.1	52.5 ± 2.1	78.9 ± 1.9	90.4 ± 5.7	75.0 ± 3.1	59.9 ± 4.3	**86.0 ± 1.8** *
	DCNN	49.1	33.5	–	–	52.1	67.0	61.3	56.6	62.6
	PATCHYSAN	71.0 ± 2.2	45.2 ± 2.8	86.3 ± 1.6	49.1 ± 0.7	72.6 ± 2.2	**92.6 ± 4.2** *	75.9 ± 2.8	60.0 ± 4.8	78.6 ± 1.9
	DGCNN	70.0	47.8	–	–	73.7	85.8	75.5	58.6	74.4
	AWL	74.5 ± 5.9	51.5 ± 3.6	87.9 ± 2.5	54.7 ± 2.9	73.9 ± 1.9	87.9 ± 9.8	–	–	–
GNN variants	SUM–MLP (**GIN-0**)	**75.1 ± 5.1**	**52.3 ± 2.8**	**92.4 ± 2.5**	**57.5 ± 1.5**	**80.2 ± 1.9**	**89.4 ± 5.6**	**76.2 ± 2.8**	**64.6 ± 7.0**	**82.7 ± 1.7**
	SUM–MLP (**GIN-ϵ**)	**74.3 ± 5.1**	**52.1 ± 3.6**	**92.2 ± 2.3**	**57.0 ± 1.7**	**80.1 ± 1.9**	**89.0 ± 6.0**	**75.9 ± 3.8**	63.7 ± 8.2	**82.7 ± 1.6**
	SUM–1-LAYER	74.1 ± 5.0	**52.2 ± 2.4**	90.0 ± 2.7	55.1 ± 1.6	**80.6 ± 1.9**	**90.0 ± 8.8**	**76.2 ± 2.6**	63.1 ± 5.7	82.0 ± 1.5
	MEAN–MLP	73.7 ± 3.7	**52.3 ± 3.1**	50.0 ± 0.0	20.0 ± 0.0	79.2 ± 2.3	83.5 ± 6.3	75.5 ± 3.4	**66.6 ± 6.9**	80.9 ± 1.8
	MEAN–1-LAYER (GCN)	74.0 ± 3.4	51.9 ± 3.8	50.0 ± 0.0	20.0 ± 0.0	79.0 ± 1.8	85.6 ± 5.8	76.0 ± 3.2	64.2 ± 4.3	80.2 ± 2.0
	MAX–MLP	73.2 ± 5.8	51.1 ± 3.6	–	–	–	84.0 ± 6.1	76.0 ± 3.2	64.6 ± 10.2	77.8 ± 1.3
	MAX–1-LAYER (GraphSAGE)	72.3 ± 5.3	50.9 ± 2.2	–	–	–	85.1 ± 7.6	75.9 ± 3.2	63.9 ± 7.7	77.7 ± 1.5

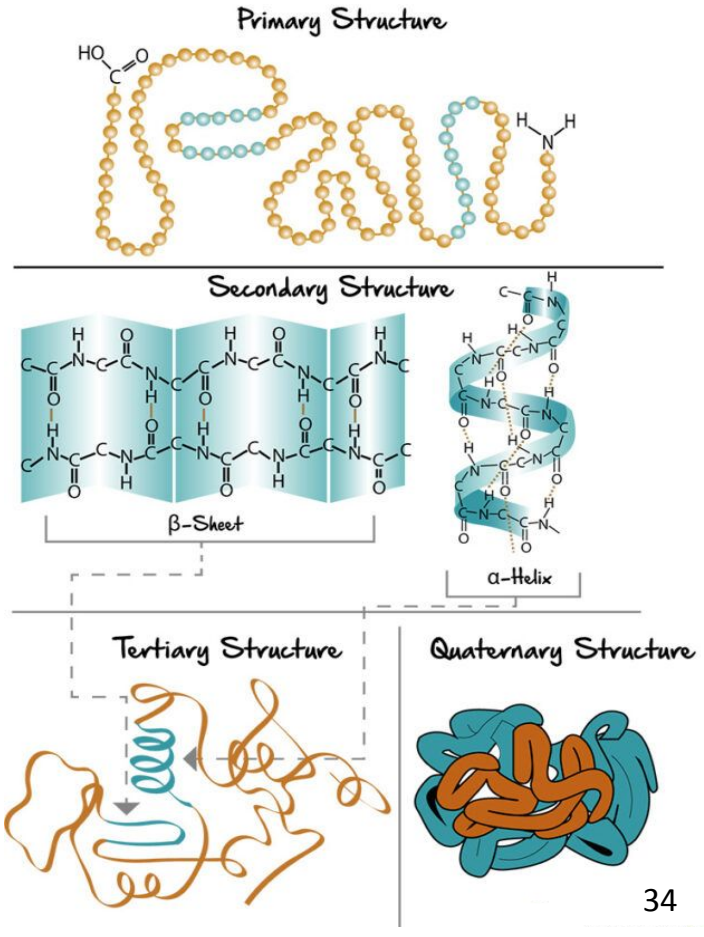
ICLR 2019. “How Powerful are Graph Neural Networks?”

Application: Protein Folding Problem

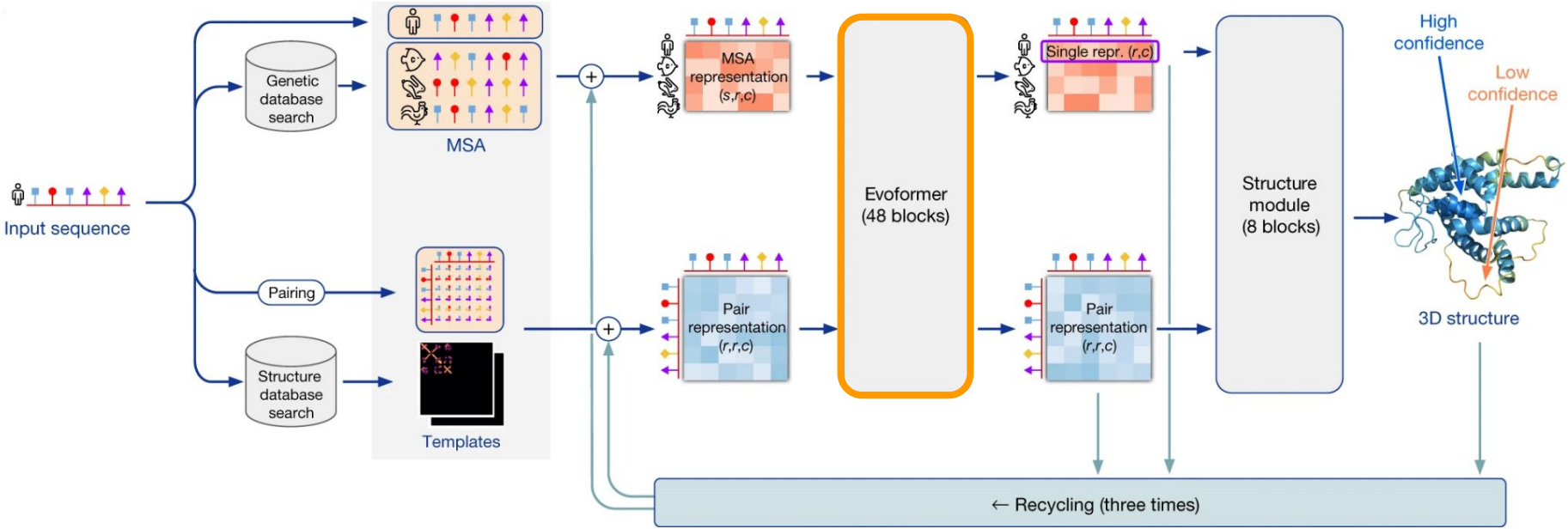
Given a protein's amino acid sequence, can we predict the 3D structure of the underlying protein?

- Solved in 2020 by AlphaFold model
- Key idea: protein as spatial graph

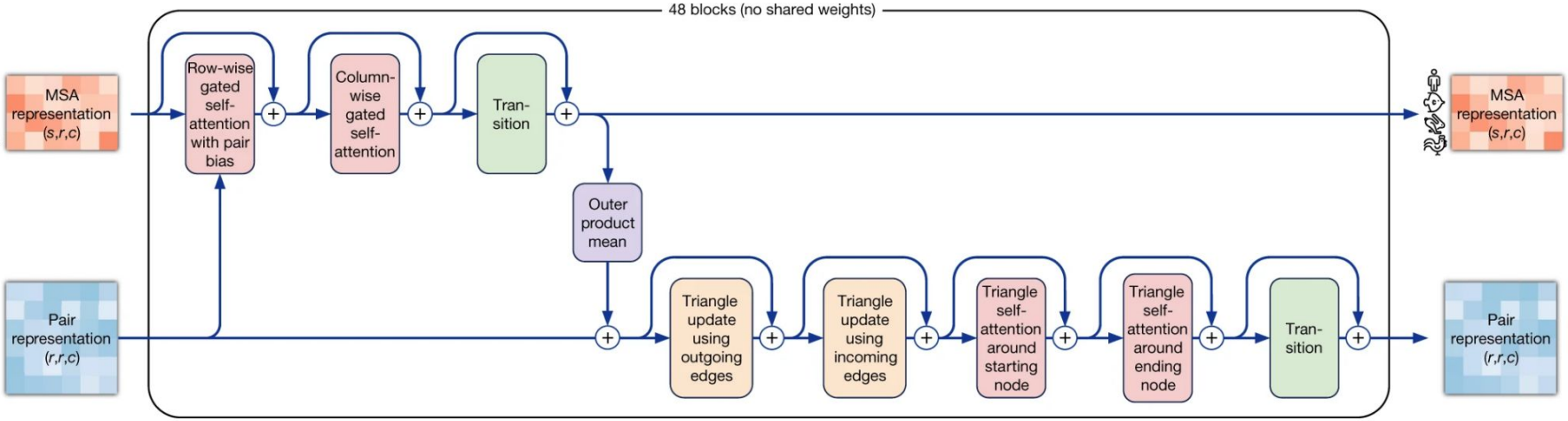


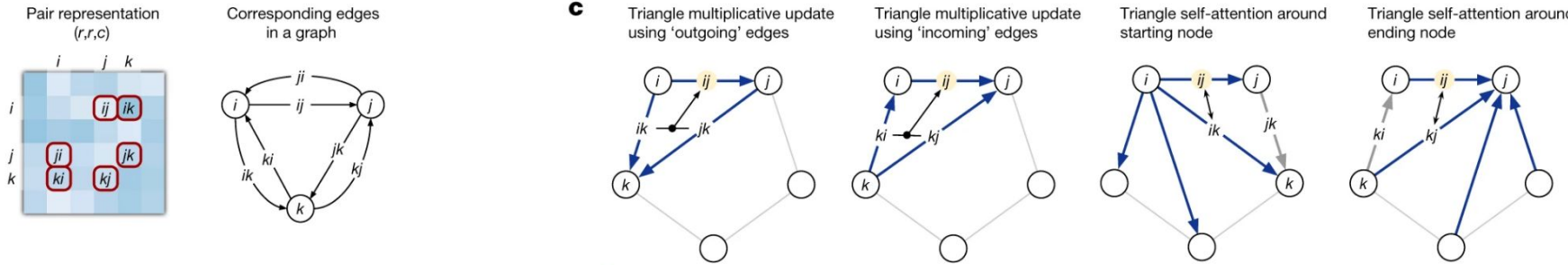


AlphaFold Architecture



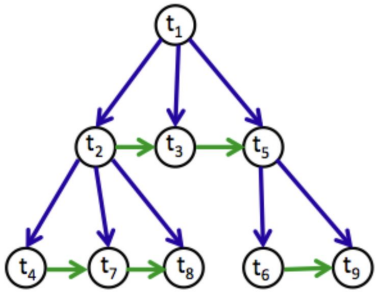
Evoformer Block



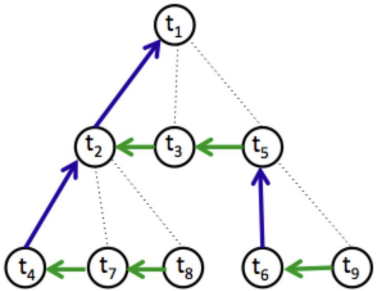


Application: Reddit Conversation Modeling

- Graph-structured bi-directional LSTM model
- Evaluated on predicting karma
- Textual context
- Hierarchical and temporal conversation structure



(a) Forward hierarchical and timing structure



(b) Backward hierarchical and timing structure

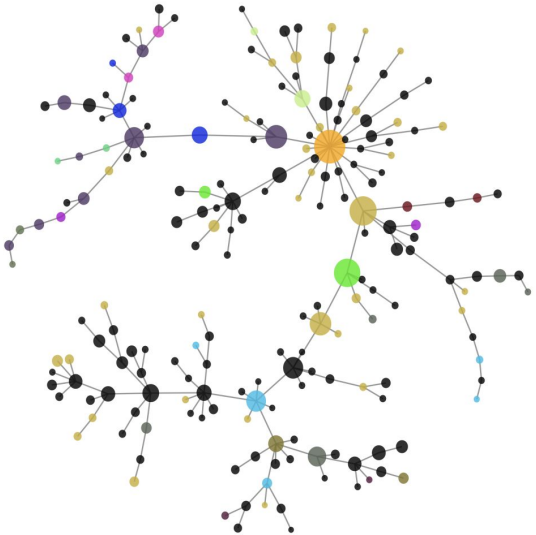
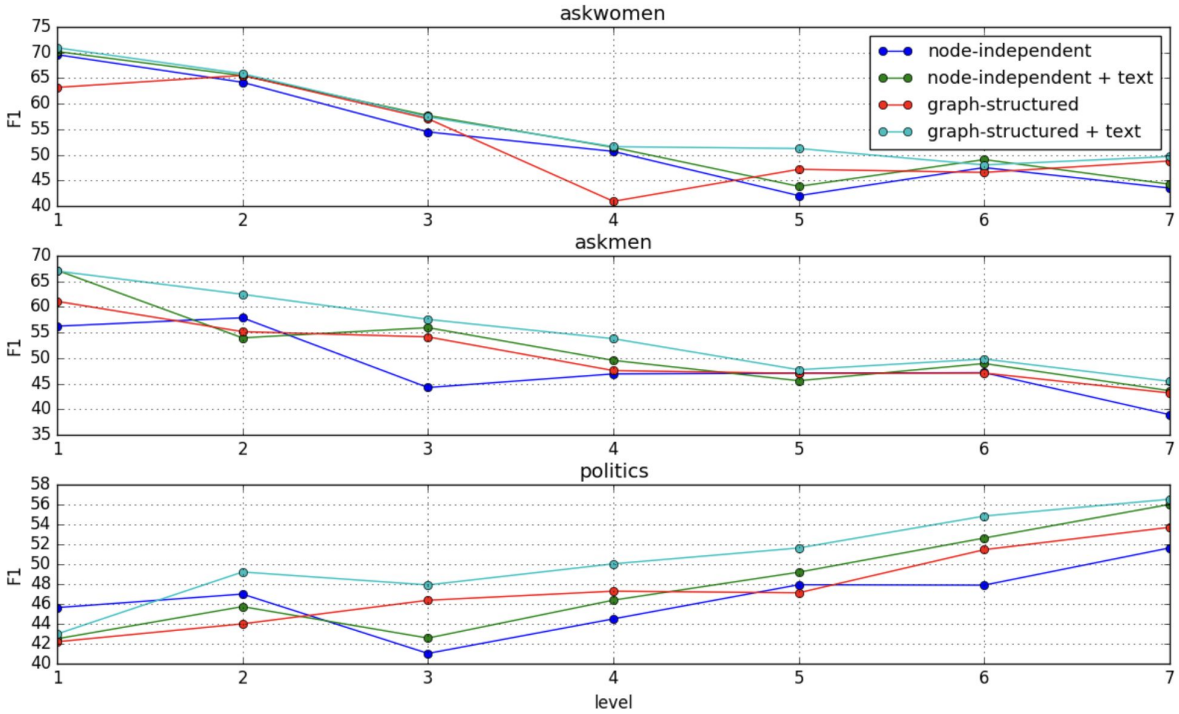
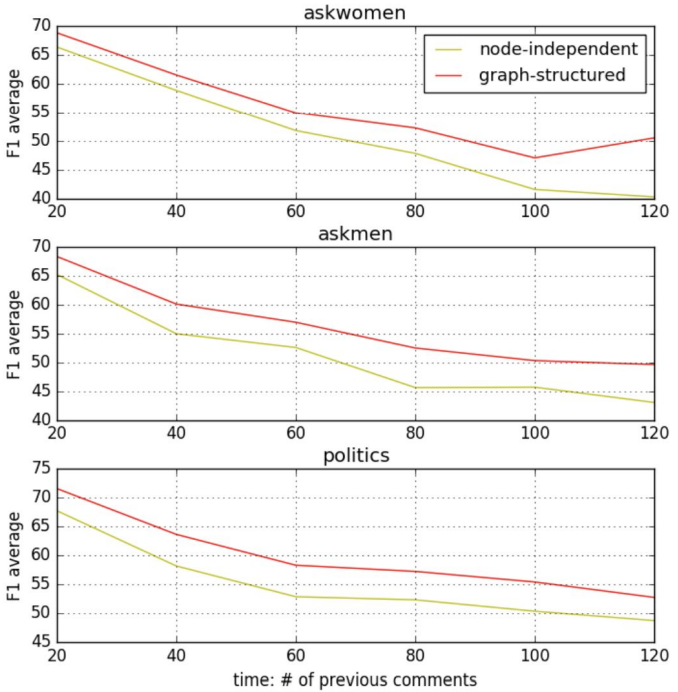


Figure 1: Visualization of a sample thread on Reddit.

2017. “Conversation Modeling on Reddit using a Graph-Structured LSTM”

Model Performance Across Forums





2017. “Conversation Modeling on Reddit using a Graph-Structured LSTM”

Conclusion

- GNNs are very powerful and can produce promising results, especially for data already structured as graphs
- Important applications across diverse fields
- Limitations in distinguishing between non-isomorphic graphs

TEXAS
The University of Texas at Austin

Thank You!