

How Powerful are Graph Neural Networks

Jure Leskovec



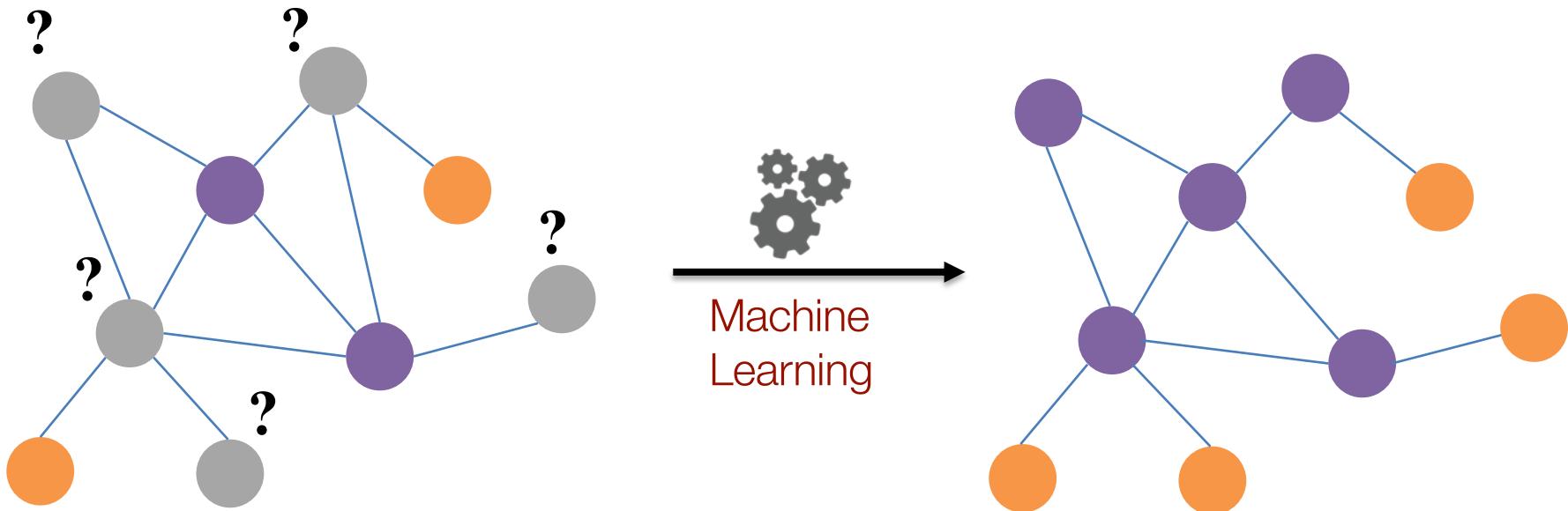
CHAN ZUCKERBERG
BIOHUB

Tasks on Networks

Classical ML tasks in networks:

- Node classification
 - Predict a type of a given node
- Link prediction
 - Predict whether two nodes are linked
- Community detection
 - Identify densely linked clusters of nodes
- Network similarity
 - How similar are two (sub)networks

ML on Graphs

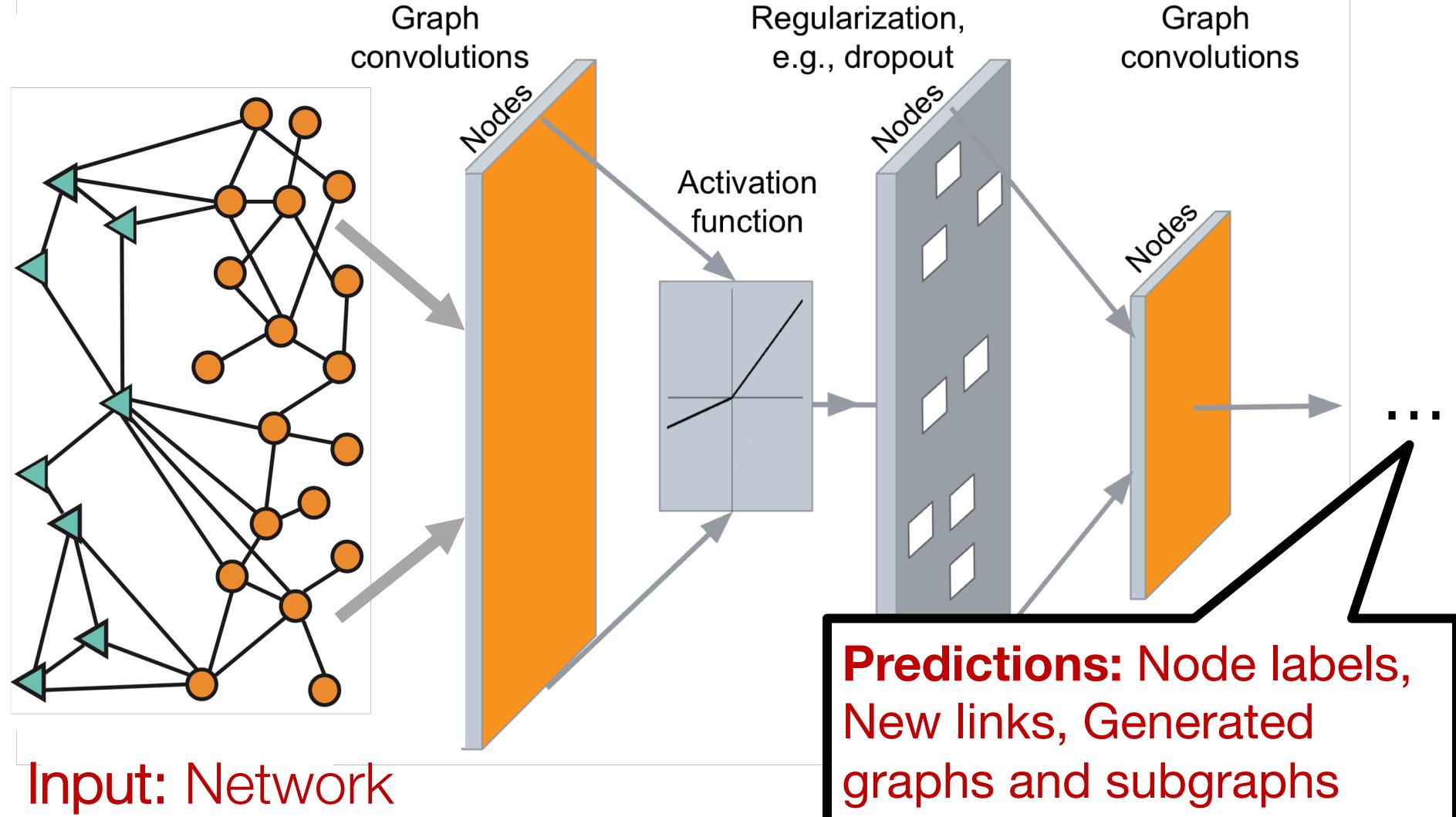


Example: Node Classification

Many possible ways to create node features:

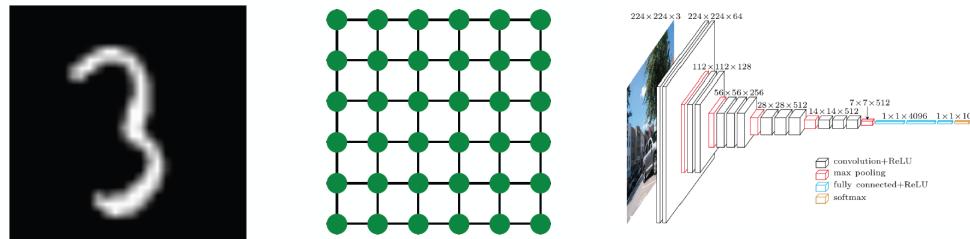
- Node degree, PageRank score, Motifs, Degree of neighbors, Clustering, ...

Deep Learning in Graphs

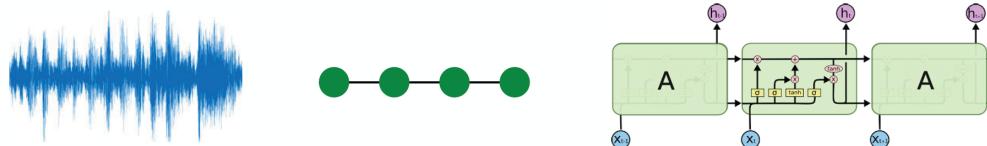


Why is it Hard?

- Modern deep learning toolbox is designed for simple sequences or grids
 - CNNs for fixed-size images/grids....



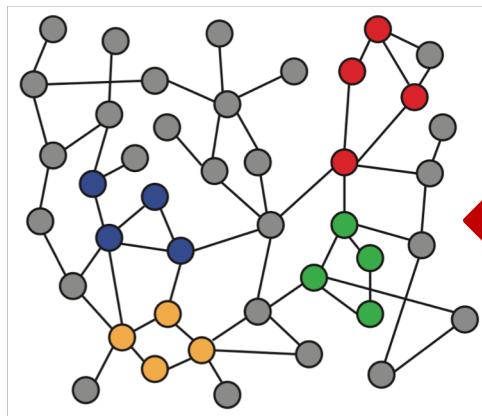
- RNNs or word2vec for text/sequences...



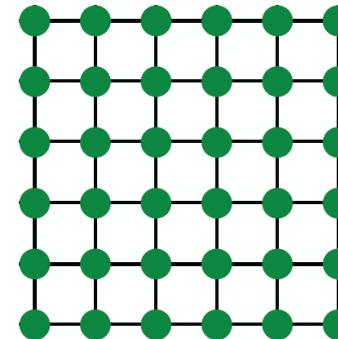
Why is it Hard?

But networks are far more complex!

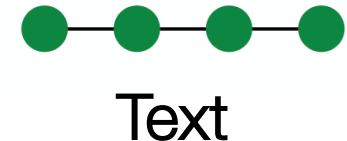
- Arbitrary size and complex topological structure (i.e., no spatial locality like grids)



Networks



Images



Text

- No fixed node ordering or reference point
- Often dynamic and have multimodal features

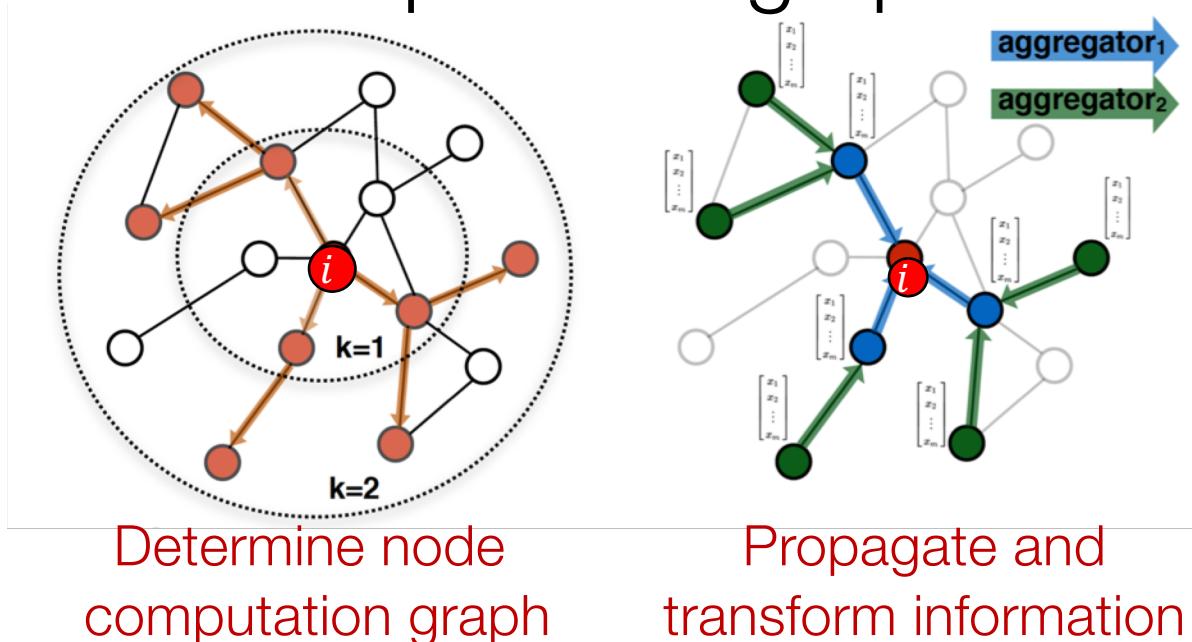
Setup

We have a graph G :

- V is the **vertex set**
- A is the (binary) **adjacency matrix**
- $X \in \mathbb{R}^{m \times |V|}$ is a matrix of **node features**
 - Meaningful node features:
 - Social networks: User profile
 - Biological networks: Gene expression profiles, gene functional information

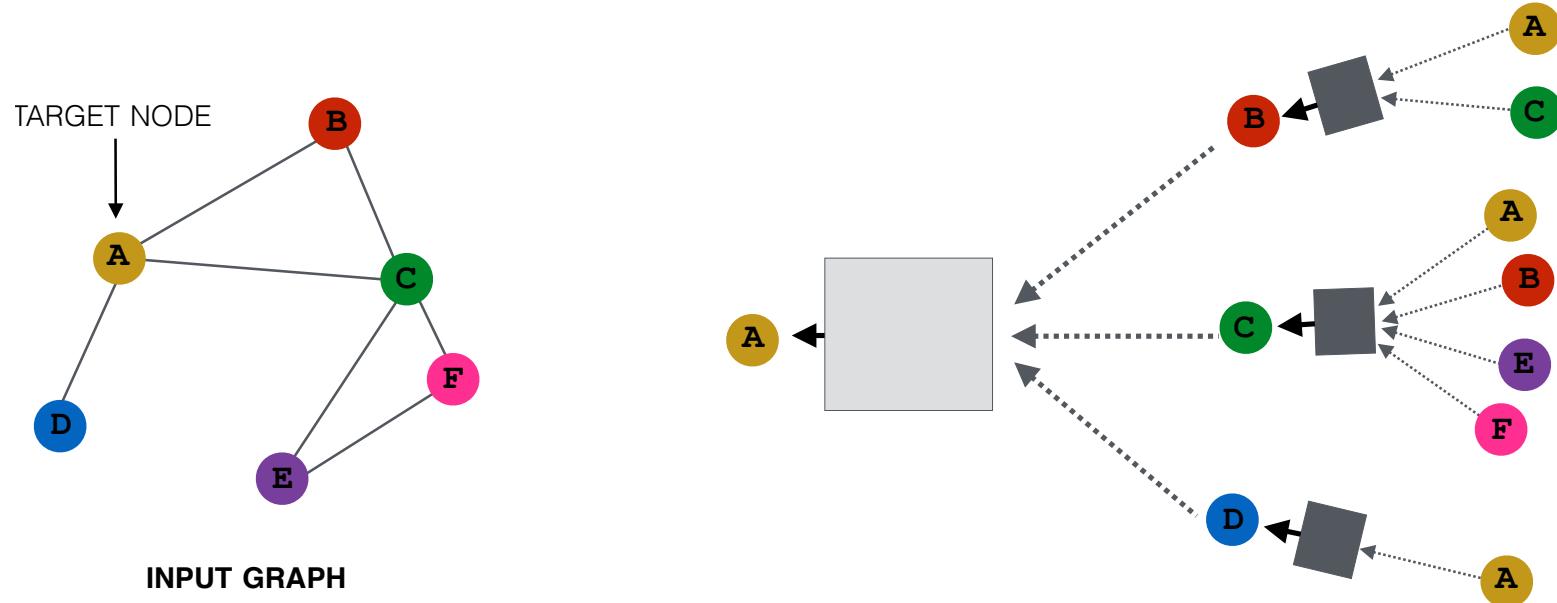
Graph Neural Networks

Idea: Node's neighborhood defines a computation graph



Learn how to propagate information across the graph to compute node features

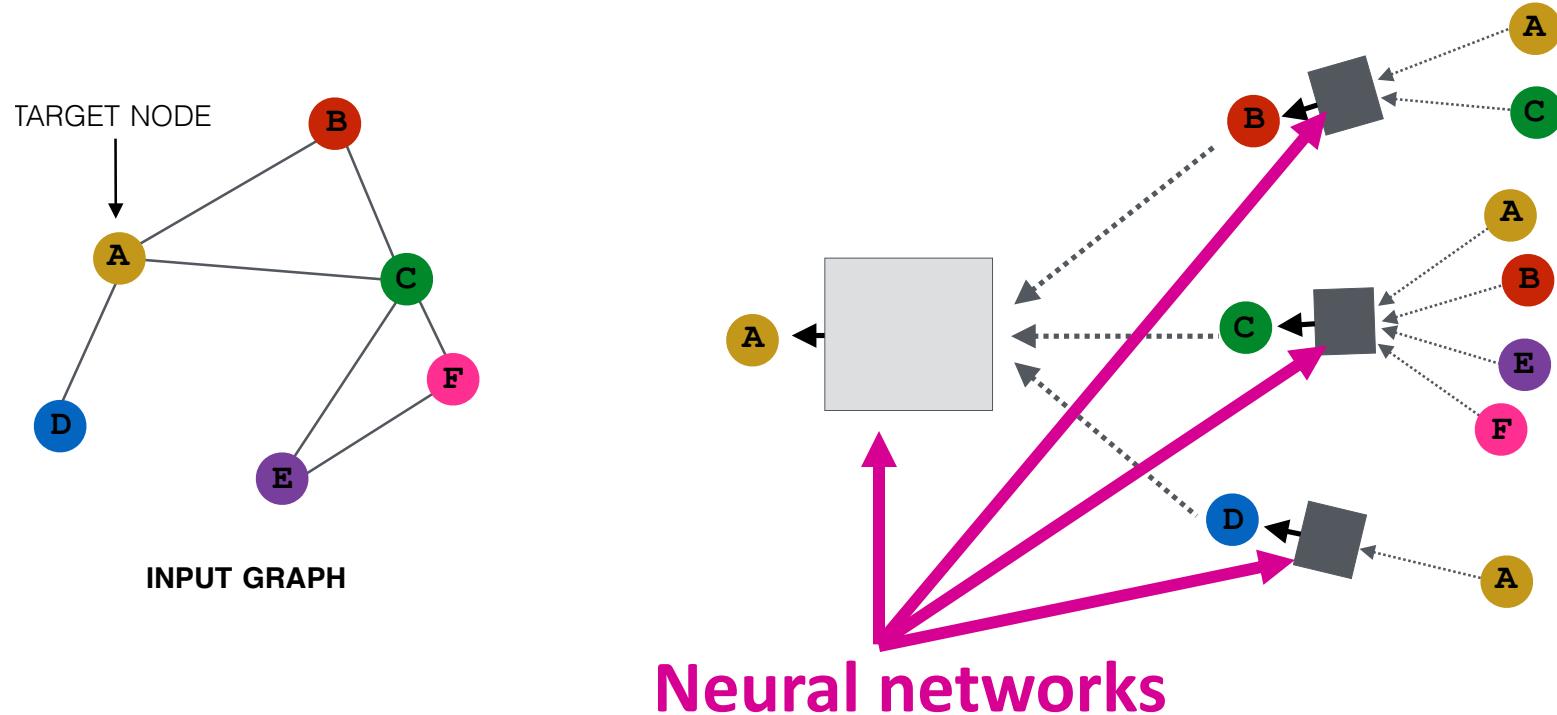
Graph Neural Networks



Each node defines a computation graph

- Each edge in this graph is a transformation/aggregation function

Graph Neural Networks

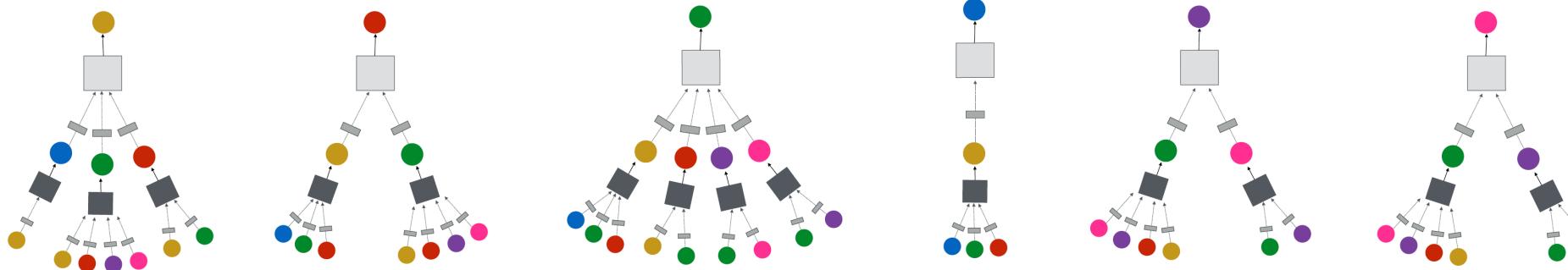
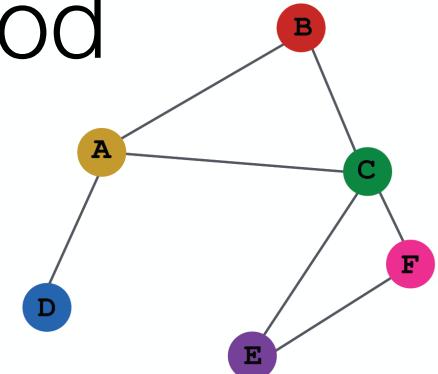


Intuition: Nodes aggregate information from their neighbors using neural networks

Idea: Aggregate Neighbors

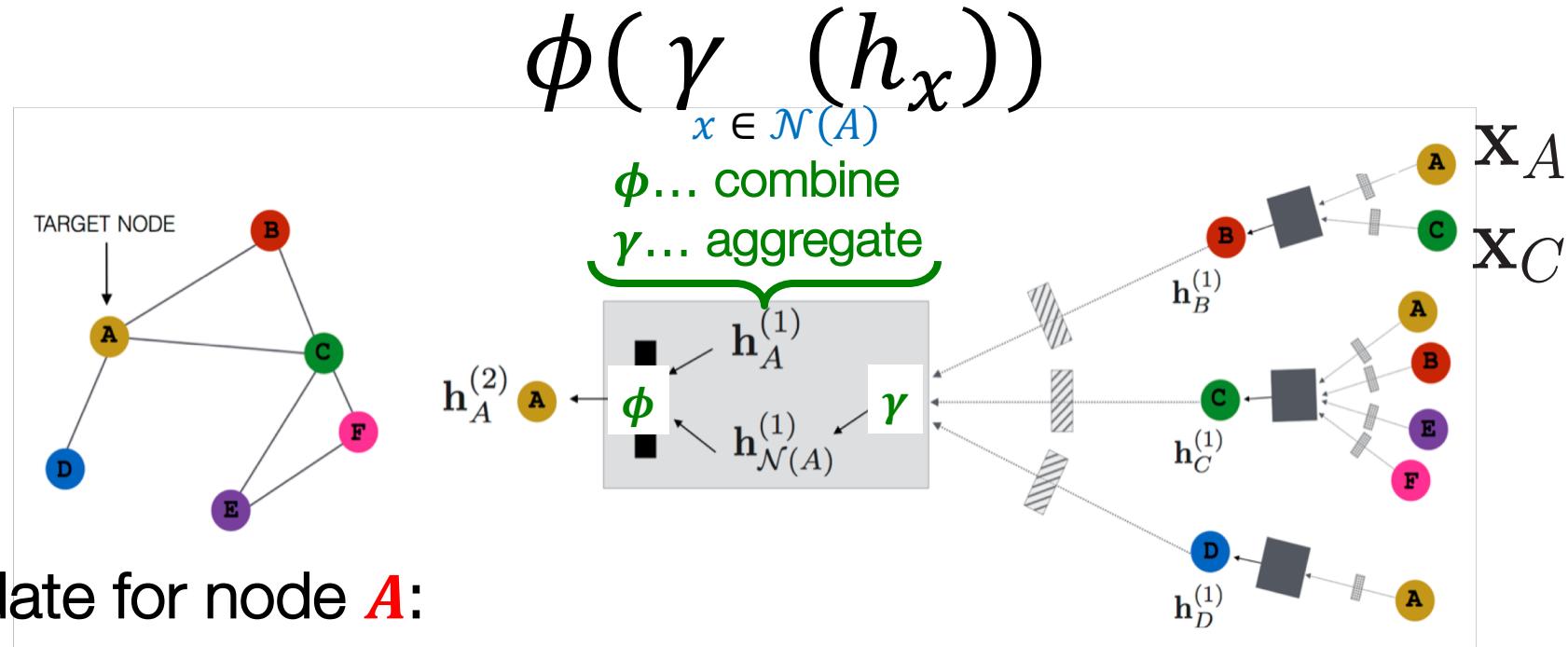
Intuition: Network neighborhood defines a computation graph

Every node defines a computation graph based on its neighborhood!



Can be viewed as learning a generic linear combination of graph low-pass and high-pass operators

Our Approach: GraphSAGE



Update for node A :

$$h_A^{(k+1)} = \sigma \left(W^{(k)} h_A^{(k)}, \gamma \left(\sigma(Q^{(k)} h_x^{(k)}) \right) \right)$$

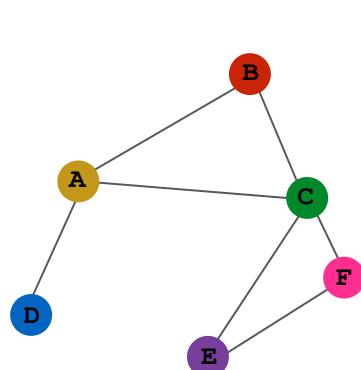
where

- $h_A^{(k+1)}$ is the $k + 1^{\text{st}}$ level embedding of node A
- $W^{(k)} h_A^{(k)}$ is the transformed embedding of node A from level k
- $\gamma \left(\sigma(Q^{(k)} h_x^{(k)}) \right)$ is the aggregated transformed embeddings of neighbors $x \in \mathcal{N}(A)$

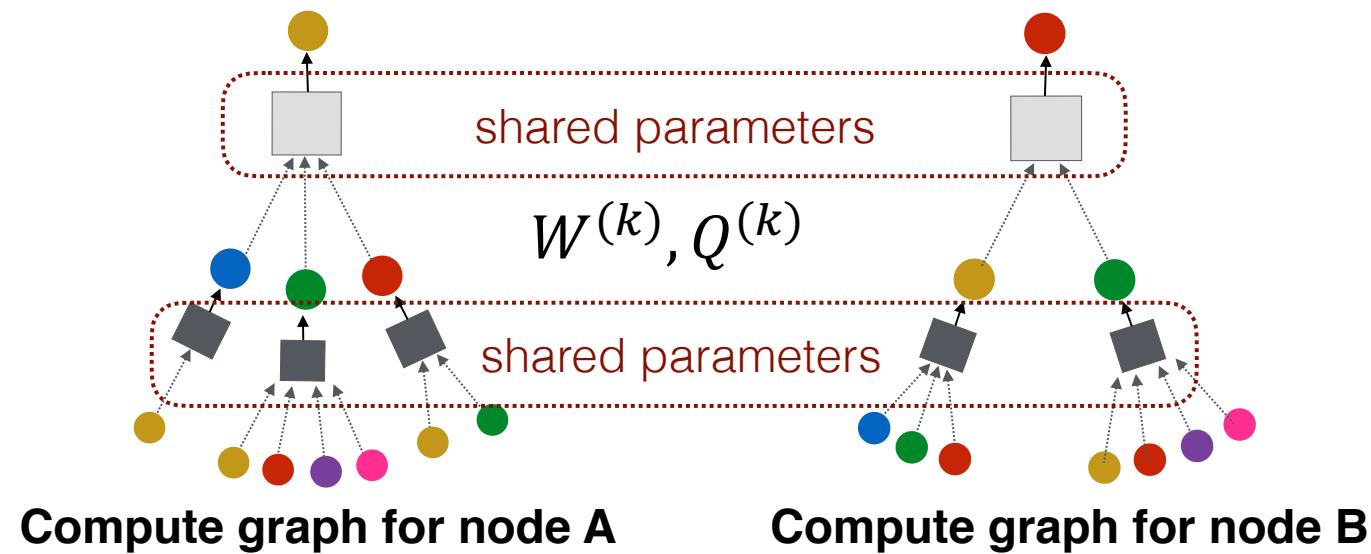
■ $h_A^{(0)} = \text{attributes } X_A \text{ of node } A$

GraphSAGE: Training

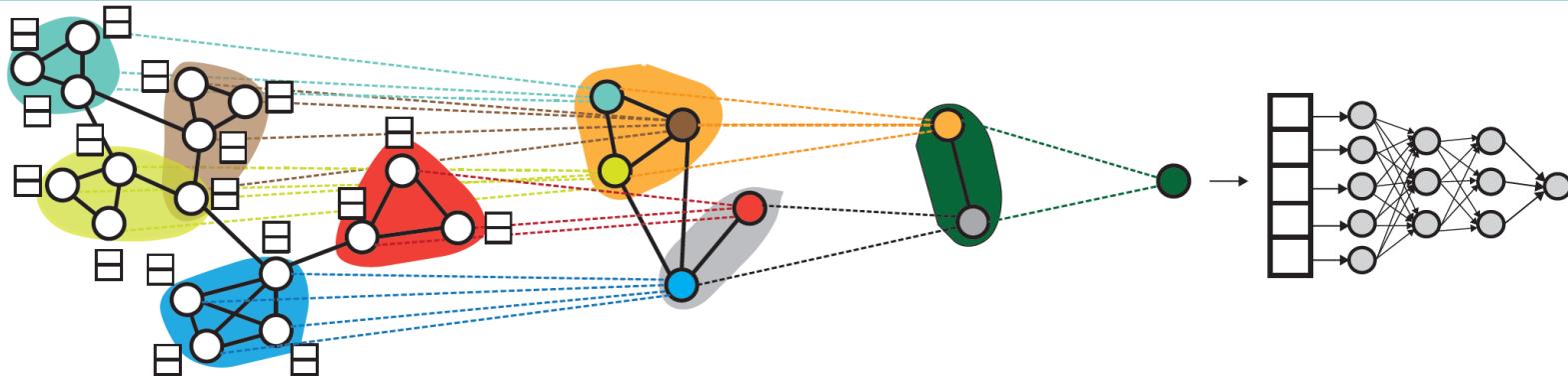
- Aggregation parameters are shared for all nodes
- The number of model parameters is sublinear in $|V|$
- Can use different loss functions:
 - Classification/Regression: $\mathcal{L}(h_A) = \|y_A - f(h_A)\|^2$
 - Pairwise Loss: $\mathcal{L}(h_A, h_B) = \max(0, 1 - dist(h_A, h_B))$



INPUT GRAPH



Pooling for GNNs



Don't just embed individual nodes. **Embed the entire graph.**

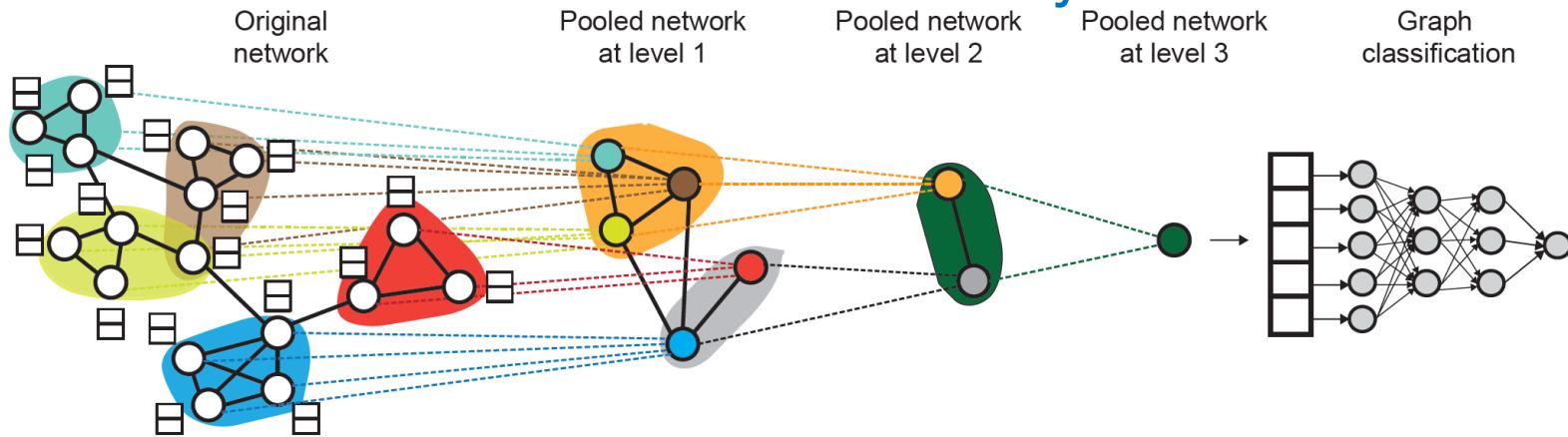
Problem: Learn how to hierarchical pool the nodes to embed the entire graph

Our solution: DIFFPOOL

- Learns hierarchical pooling strategy
- Sets of nodes are pooled hierarchically
- Soft assignment of nodes to next-level nodes

DIFFPOOL Architecture

A different GNN is learned at every level of abstraction

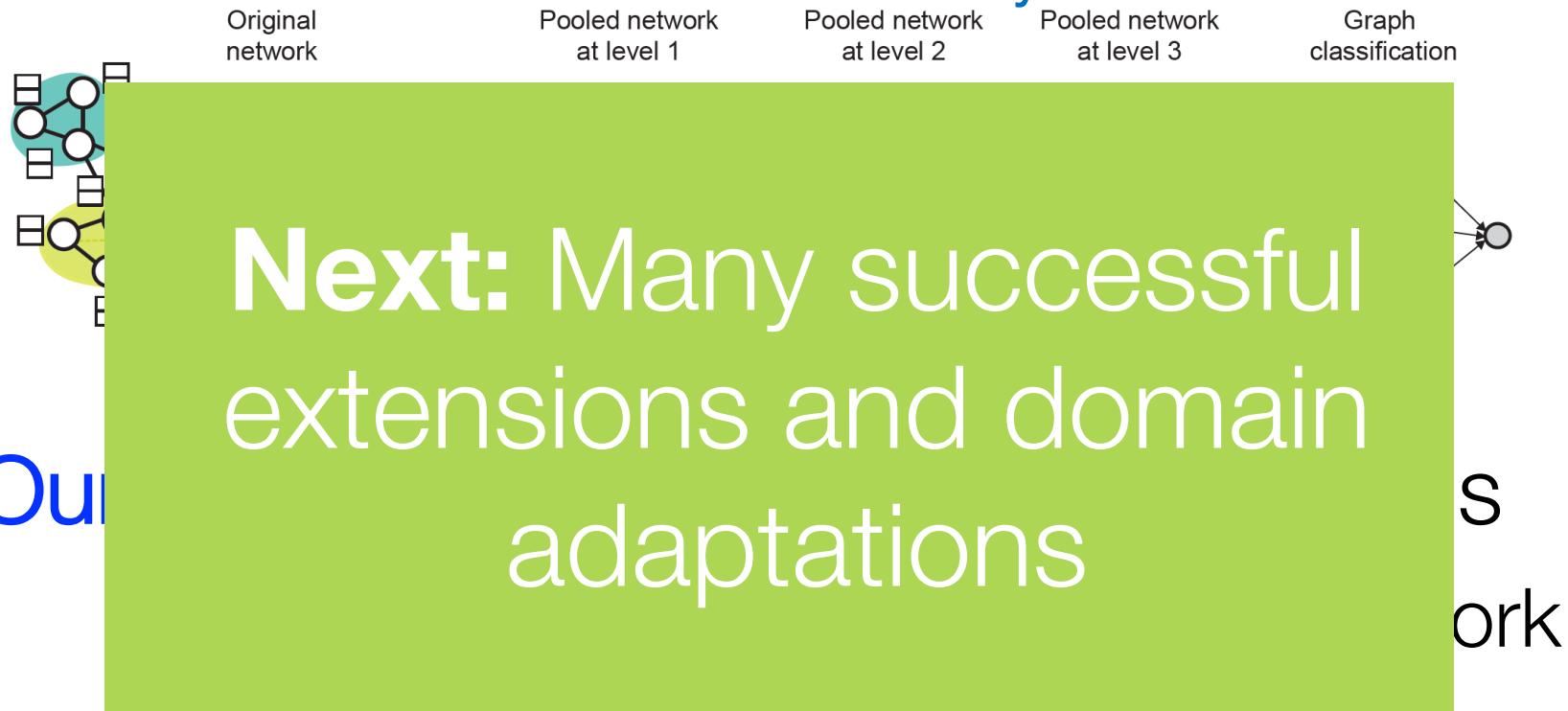


Our approach: Use two sets of GNNs

- GNN1 to learn how to pool the network
 - Learn cluster assignment matrix
- GNN2 to learn the node embeddings

DIFFPOOL Architecture

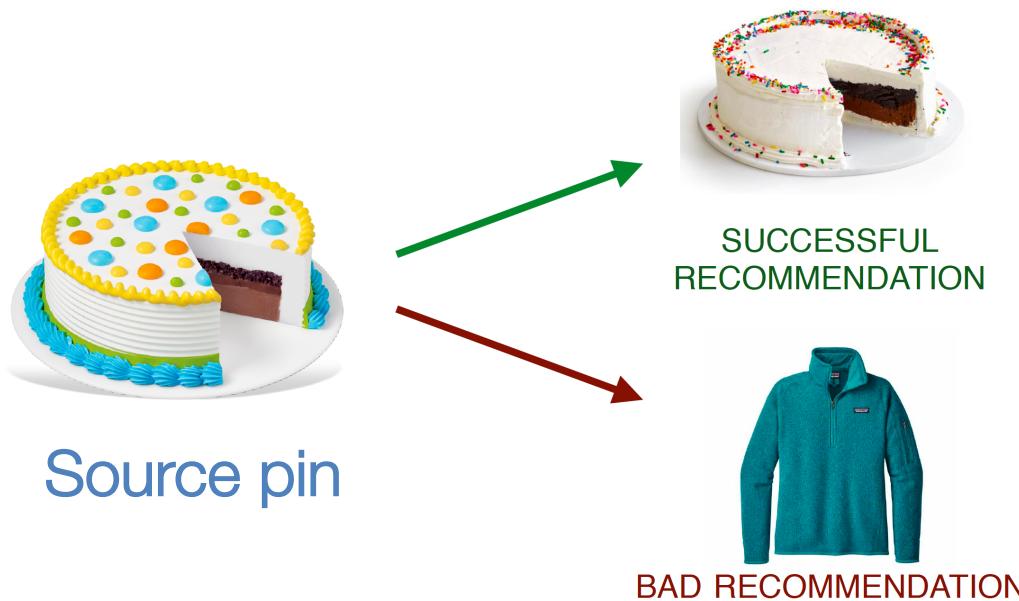
A different GNN is learned at every level of abstraction



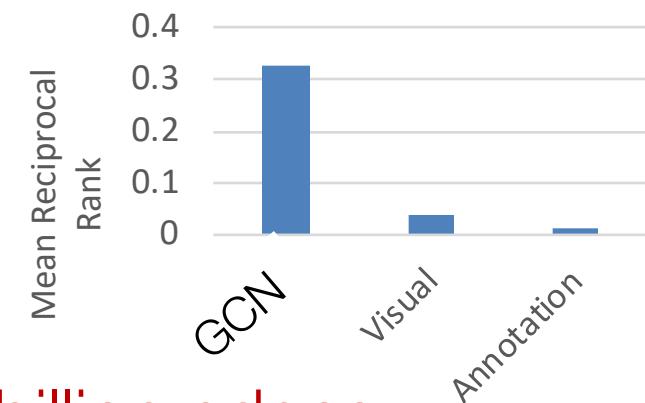


Application (1): Pinterest

Task: Recommend related pins



Task: Learn node embeddings z_i s.t.

$$d(z_{cake1}, z_{cake2}) < d(z_{cake1}, z_{sweater})$$


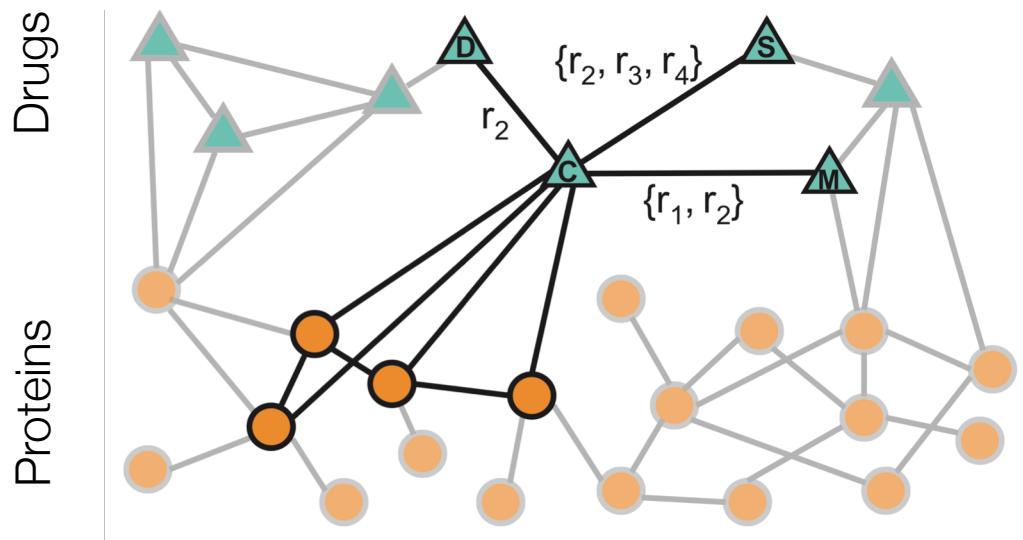
- **Challenges:**
 - Massive size: 3 billion nodes, 20 billion edges
 - Heterogeneous data: Rich image and text features

(2) Drug Side Effects

- **Task: Given a pair of drugs predict adverse side effects**

46% of people ages 70-79 take >5 drugs

- Link prediction on a multimodal graph



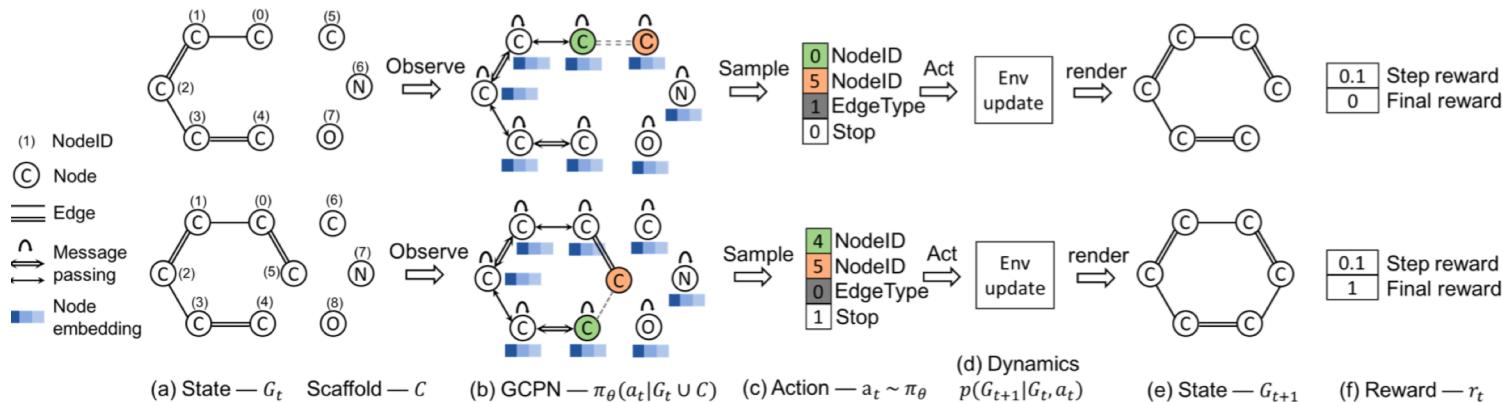
36% improvement
in AP@50 over
state of the art

(3) Targeted Molecule Generation

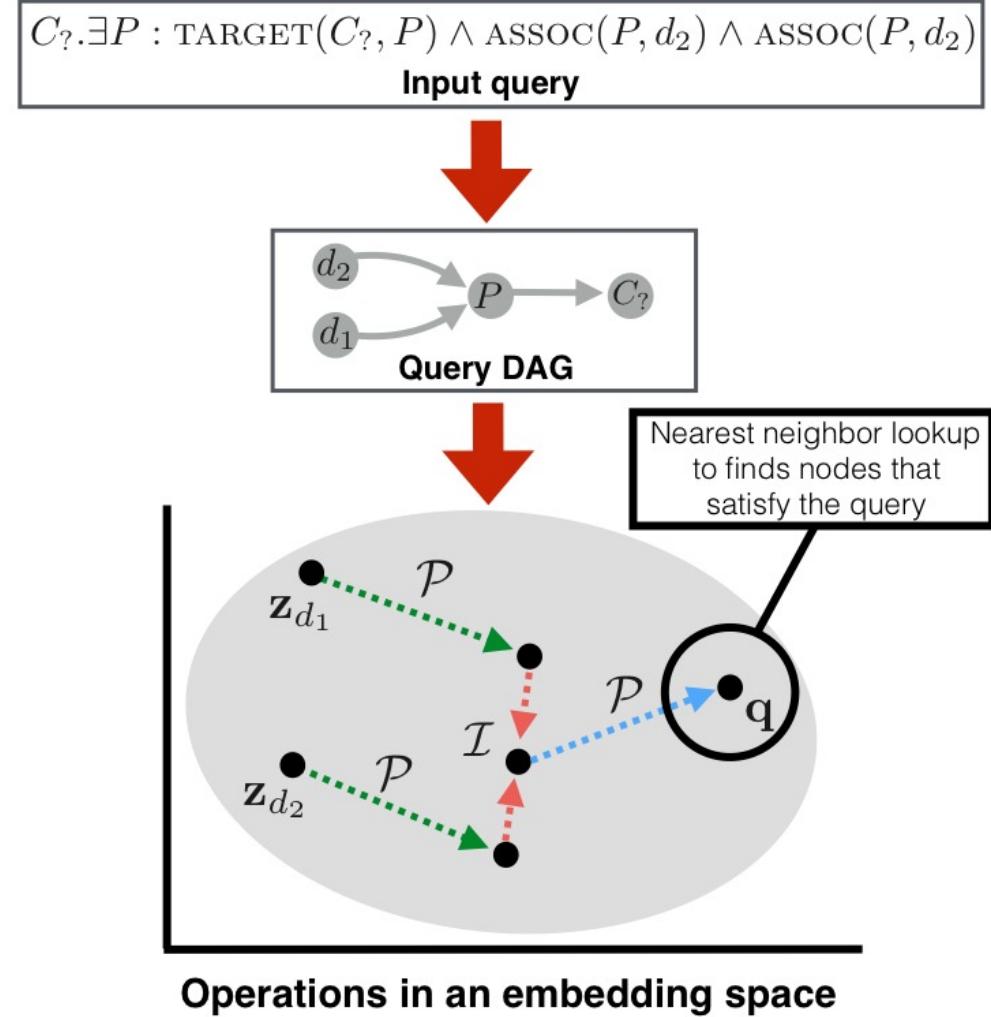
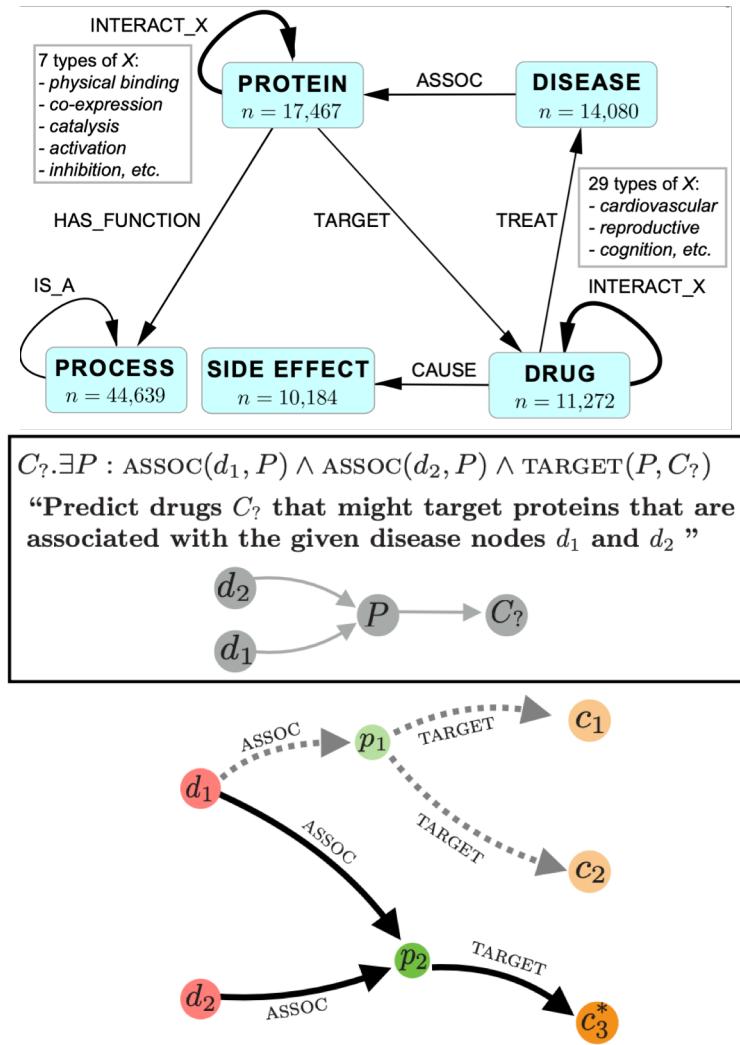
Goal: Generate molecules that optimize a given property (Quant. energy, solubility)

Solution: Combination of

- Graph representation learning
- Adversarial training
- Reinforcement learning



(4) Knowledge Graphs

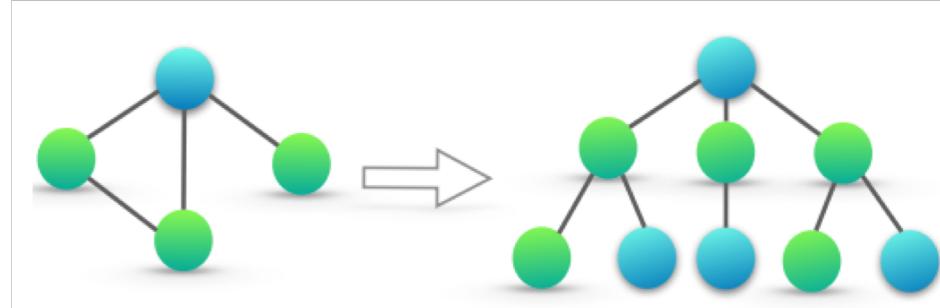


How expressive are GNNs?

Theoretical framework: Characterize GNN's discriminative power by analyzing the discriminative power of the multiset function:

- Characterize upper bound of the discriminative power of GNNs
- Propose a maximally powerful GNN
- Characterize discriminative power of popular GNNs that use non-injective multiset functions

Discriminative Power of GNNs



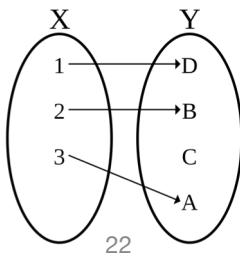
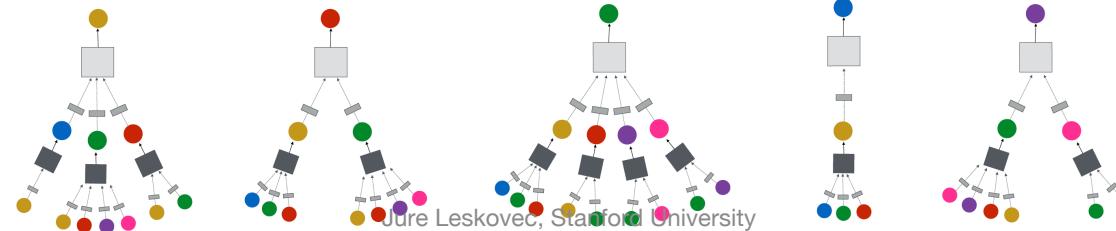
$$\phi(\gamma_{x \in \mathcal{N}(A)}(h_x))$$

ϕ ... combine
 γ ... aggregate

Theorem: The most discriminative GNN uses injective multiset function for neighbor aggregation

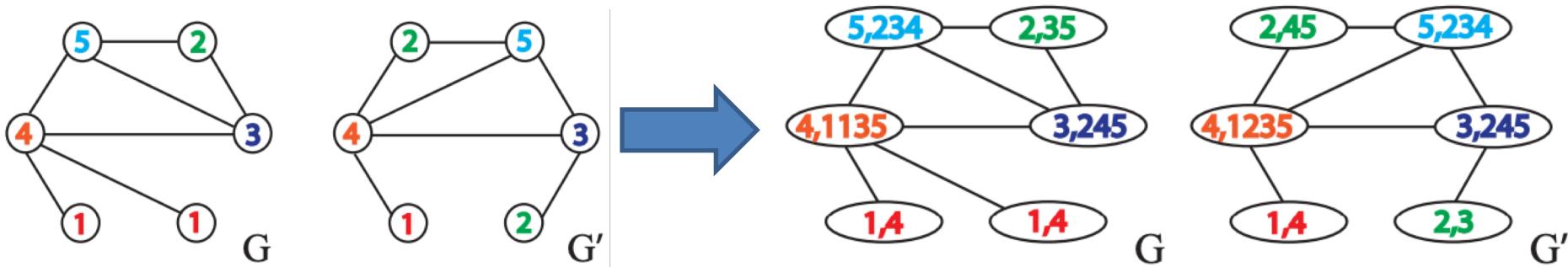
If the aggregation function is injective, GNN can fully capture/distinguish the rooted subtree structures:

Injective



Discriminative Power of GNNs

Theorem: GNNs can be at most as powerful as the Weisfeiler-Lehman graph isomorphism test (a.k.a. canonical labeling or color refinement)



What GNN achieves this upper bound?

Injective functions over multiset

Idea: If GNN functions are injective, GNN can capture/distinguish the rooted subtree structures

Theorem: Any injective multiset function can be expressed as $\phi(\sum_{x \in S} f(x))$

- Solution: Model $\phi()$ and $f()$ with MLP

Consequence: Sum aggregator is the most expressive

Expressive Power of GNNs

Many popular aggregators:

$\phi(\text{MEAN } \{f(x) : x \in S\})$ cannot distinguish:



Theorem (Informal):

Mean aggregator only captures distribution information

$\phi(\text{MAX } \{f(x) : x \in S\})$ cannot distinguish:

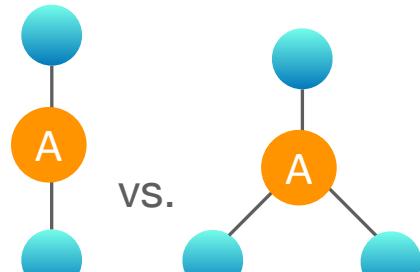


Theorem (Informal):

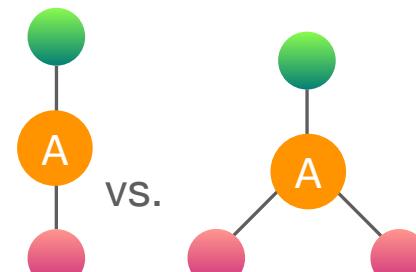
Max aggregator only captures set information (no multiplicity)

Power of Aggregators

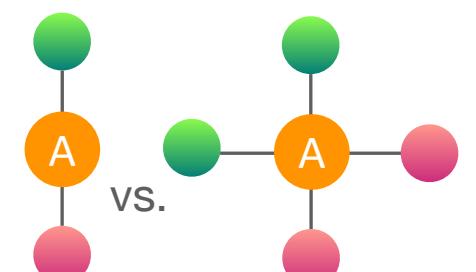
Failure cases for mean and max agg.



(a) Mean and Max both fail

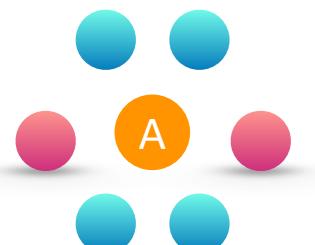


(b) Max fails

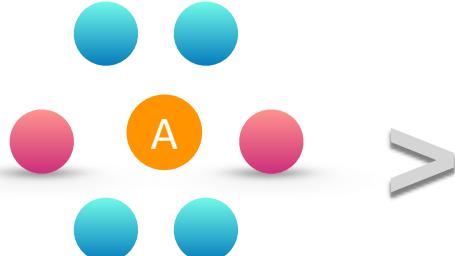


(c) Mean and Max both fail

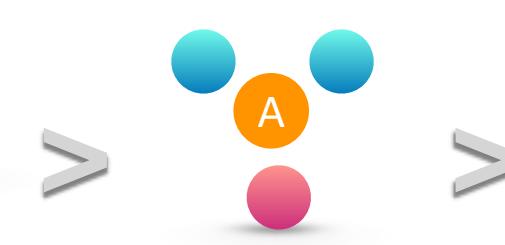
Ranking by discriminative power



Input



sum - multiset



mean - distribution



max - set

Summary

- Graph Convolution Networks
 - Generalize beyond simple convolutions
- Fuses node features & graph info
 - State-of-the-art accuracy for node classification and link prediction.
- Model size independent of graph size; can scale to billions of nodes
 - Largest embedding to date (3B nodes, 17B edges)
- Leads to significant performance gains

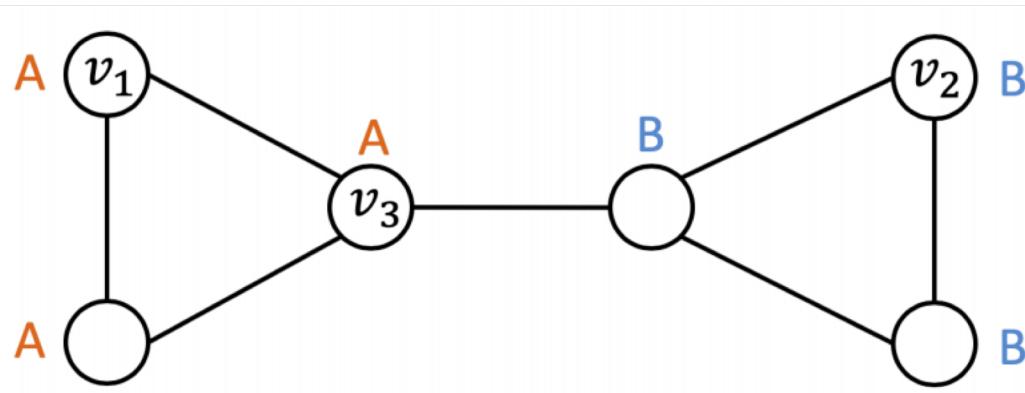
Conclusion

Results from the past 2-3 years have shown:

- Representation learning paradigm can be extended to graphs
- No feature engineering necessary
- Can effectively combine node attribute data with the network information
- State-of-the-art results in a number of domains/tasks
- Use end-to-end training instead of multi-stage approaches for better performance

Future Work

- Understand other/new GNN variants
- Understand optimization and generalization of GNNs
- Towards more powerful DL on graphs
 - **GNNs can only distinguish rooted trees:**



References

- Tutorial on Representation Learning on Networks at WWW 2018 <http://snap.stanford.edu/proj/embeddings-www/>
- [Inductive Representation Learning on Large Graphs](#).
W. Hamilton, R. Ying, J. Leskovec. NIPS 2017.
- [Representation Learning on Graphs: Methods and Applications](#). W. Hamilton, R. Ying, J. Leskovec.
IEEE Data Engineering Bulletin, 2017.
- [Graph Convolutional Neural Networks for Web-Scale Recommender Systems](#). R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec. KDD, 2018.
- [Modeling Polypharmacy Side Effects with Graph Convolutional Networks](#). M. Zitnik, M. Agrawal, J. Leskovec. Bioinformatics, 2018.
- [Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation](#). J. You, B. Liu, R. Ying, V. Pande, J. Leskovec, NeurIPS 2018.
- [Embedding Logical Queries on Knowledge Graphs](#). W. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, J. Leskovec. NeurIPS, 2018.
- [How Powerful are Graph Neural Networks?](#) K. Xu, W. Hu, J. Leskovec, S. Jegelka. ICLR 2019.
- **Code:**
 - <http://snap.stanford.edu/graphsage>
 - <http://snap.stanford.edu/decagon/>
 - https://github.com/bowenliu16/rl_graph_generation
 - <https://github.com/williamleif/graphgembd>
 - <https://github.com/snap-stanford/GraphRNN>

Industry Partnerships



YAHOO!



Funding



Collaborators

Dan Jurafsky, Linguistics, Stanford University

Christian Danescu-Miculescu-Mizil, Information Science, Cornell University

Stephen Boyd, Electrical Engineering, Stanford University

David Gleich, Computer Science, Purdue University

VS Subrahmanian, Computer Science, University of Maryland

Sarah Kunz, Medicine, Harvard University

Russ Altman, Medicine, Stanford University

Jochen Profit, Medicine, Stanford University

Eric Horvitz, Microsoft Research

Jon Kleinberg, Computer Science, Cornell University

Sendhill Mullainathan, Economics, Harvard University

Scott Delp, Bioengineering, Stanford University

Jens Ludwig, Harris Public Policy, University of Chicago



PhD Students



Alexandra
Porter



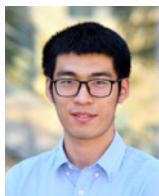
Camilo
Ruiz



Claire
Donnat



Emma
Pierson



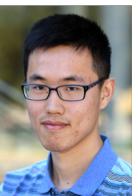
Jiaxuan
You



Mitchell
Gordon



Mohit
Tiwari



Rex
Ying

Post-Doctoral Fellows



Baharan
Mirzasoleiman



Marinka
Zitnik



Michele
Catasta



Robert
Palovics

Research Staff



Srijan
Kumar



Adrijan
Bradaschia

Rok
Sosic