Roadmap

Overview (16 min)

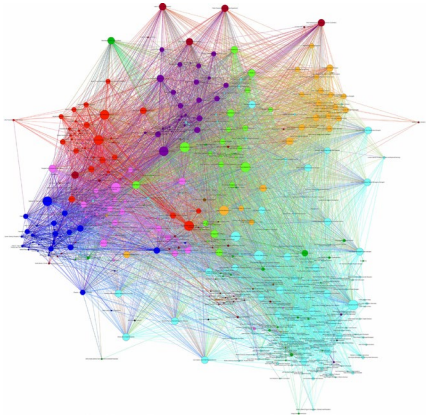
Pan: Graph NNs (8 min)

Dylan: HLS4ML (8 min)

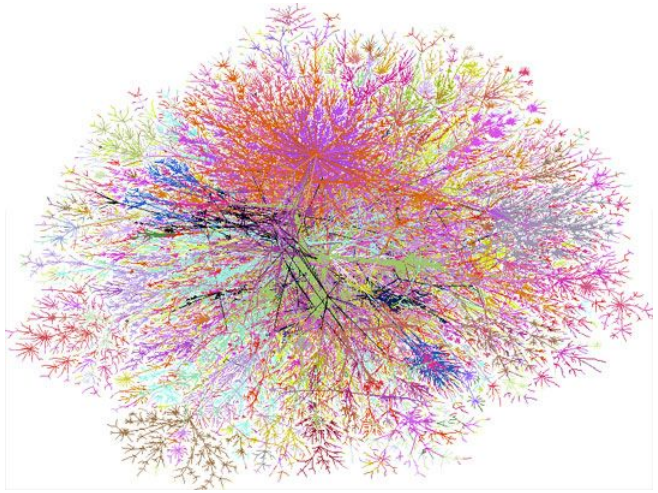
Deming: PyLog, ScaleHLS (8 min)

Song: PVCNN, SPVNAS, PointAcc (8 min)

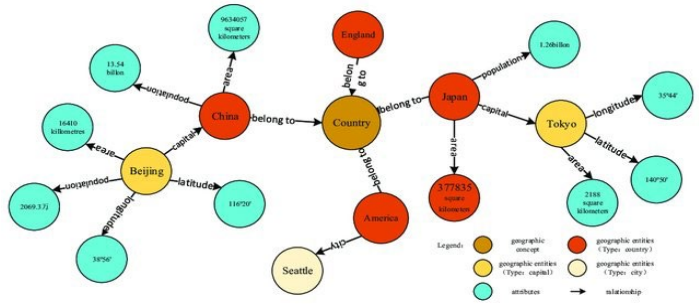
Graph Structured Data is Everywhere…



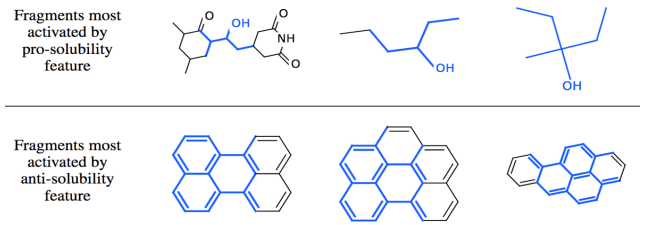
Social networks



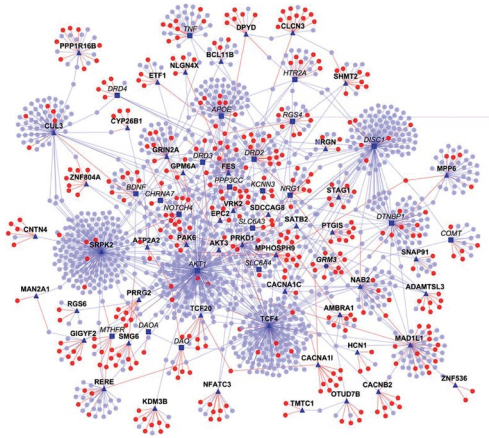
Internet networks



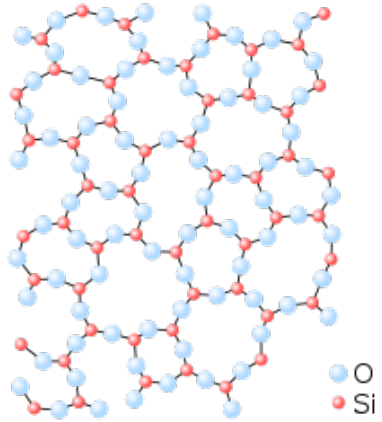
Knowledge graphs



Drug molecules



Protein-protein Interaction

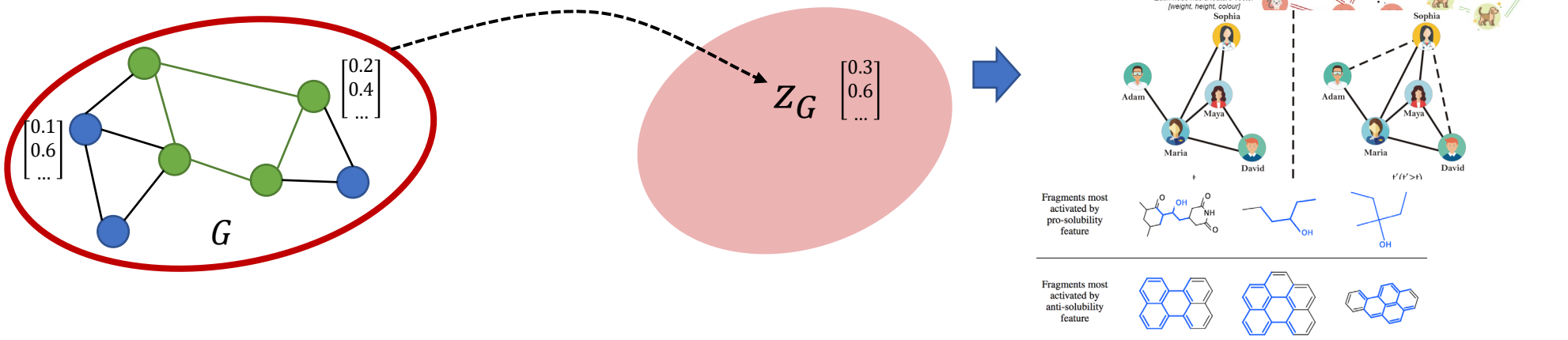


Glass structure

AI Algorithms for Graphs

- Given a graph or a collection of graphs, possibly with node or edge attributes
- AI algorithms for graphs are to extract the patterns of
  - the graph structure
  - the node/edge attributes

  for certain downstream applications.

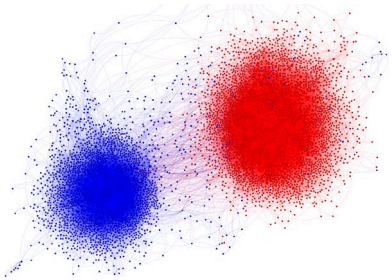


Applications

- Community detection

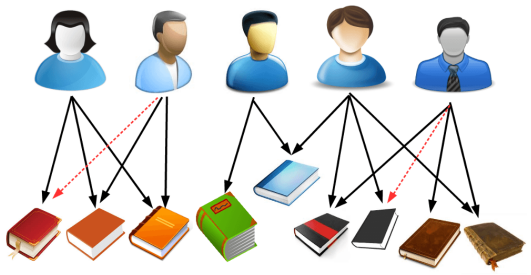
[Girvan and Newman, PNAS 2012]

A retweet network on Twitter: The colors reflect different political attitudes, conservative (red) and progressive (blue)



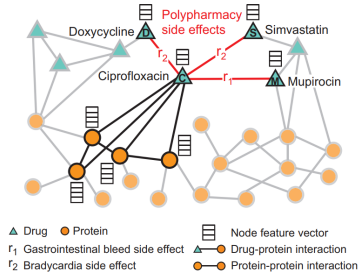
- Link prediction and recommendation

[Nowell and Kleinberg, J. Am. Soc. Inf. Sci., 2007]



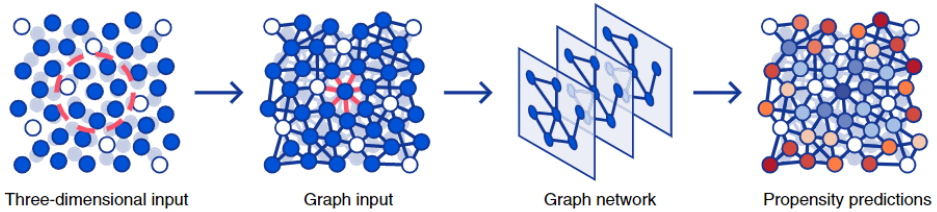
- Drug side-effect prediction

[Zitnik et al, Bioinformatics 2018]



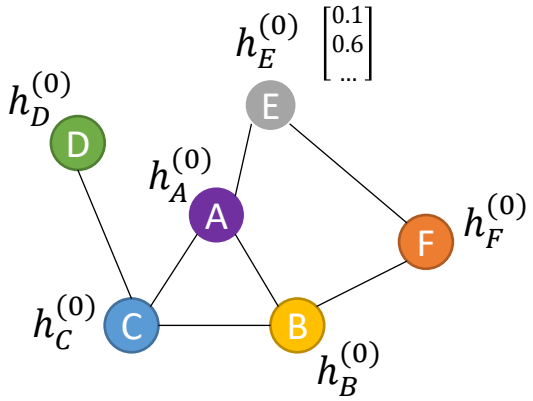
- Glass dynamics simulation

[Baspt et al, Nature Physics 2021]



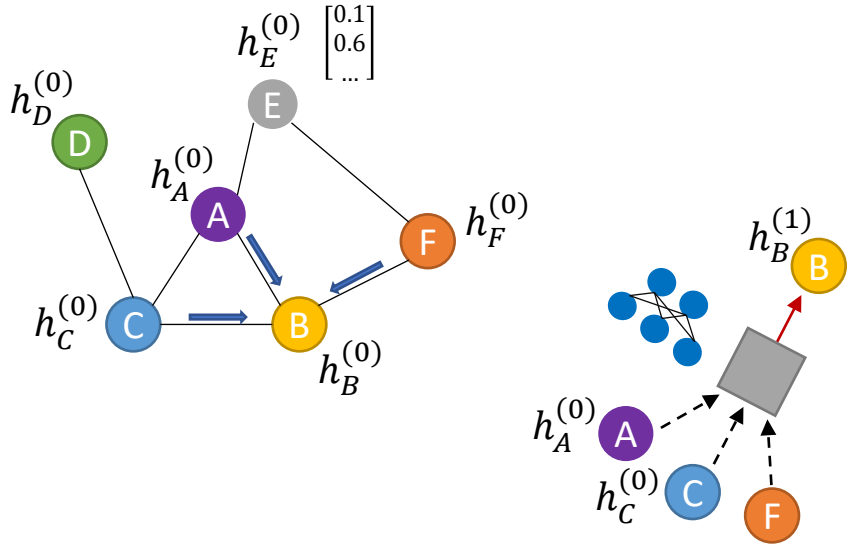
Graph Neural Networks

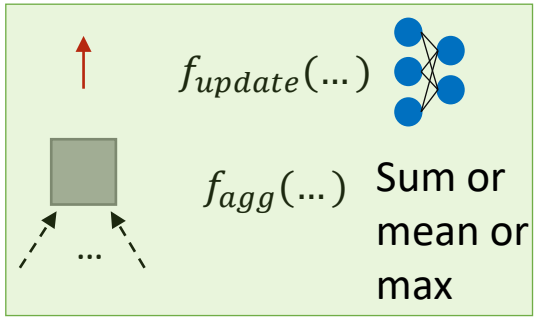
Input Data: A graph G, possibly with node/edge attributes.



Graph Neural Networks

Input Data: A graph G, possibly with node/edge attributes.





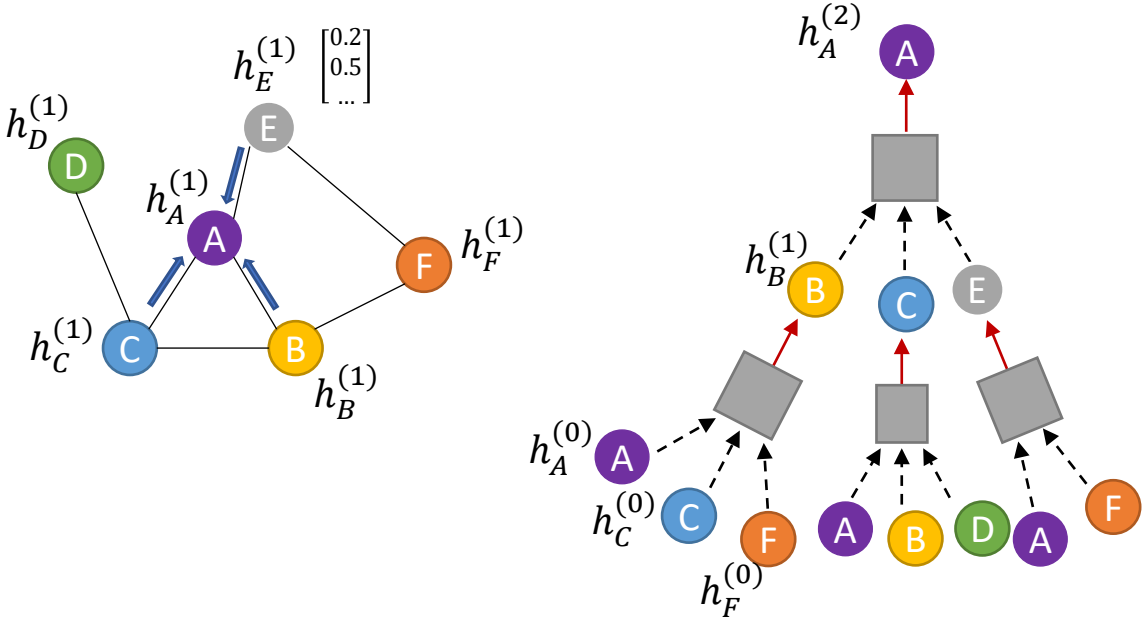
Graph neural network: one layer

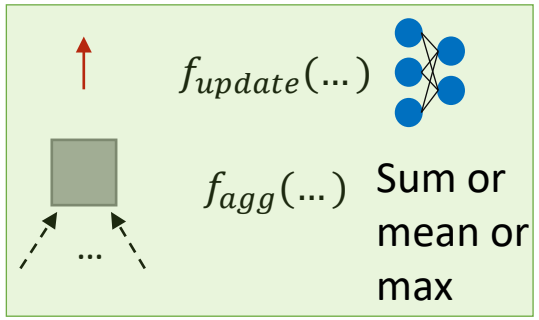
$$h_v^{(t+1)} = f_{update}\left(h_v^{(t)}, f_{agg}\left(\{h_u^{(t)} \middle| u \in N_v\}\right)\right),$$

where N_v denotes the set of the neighbors of node v.

Graph Neural Networks

Input Data: A graph G, possibly with node/edge attributes.





Graph neural network: one layer

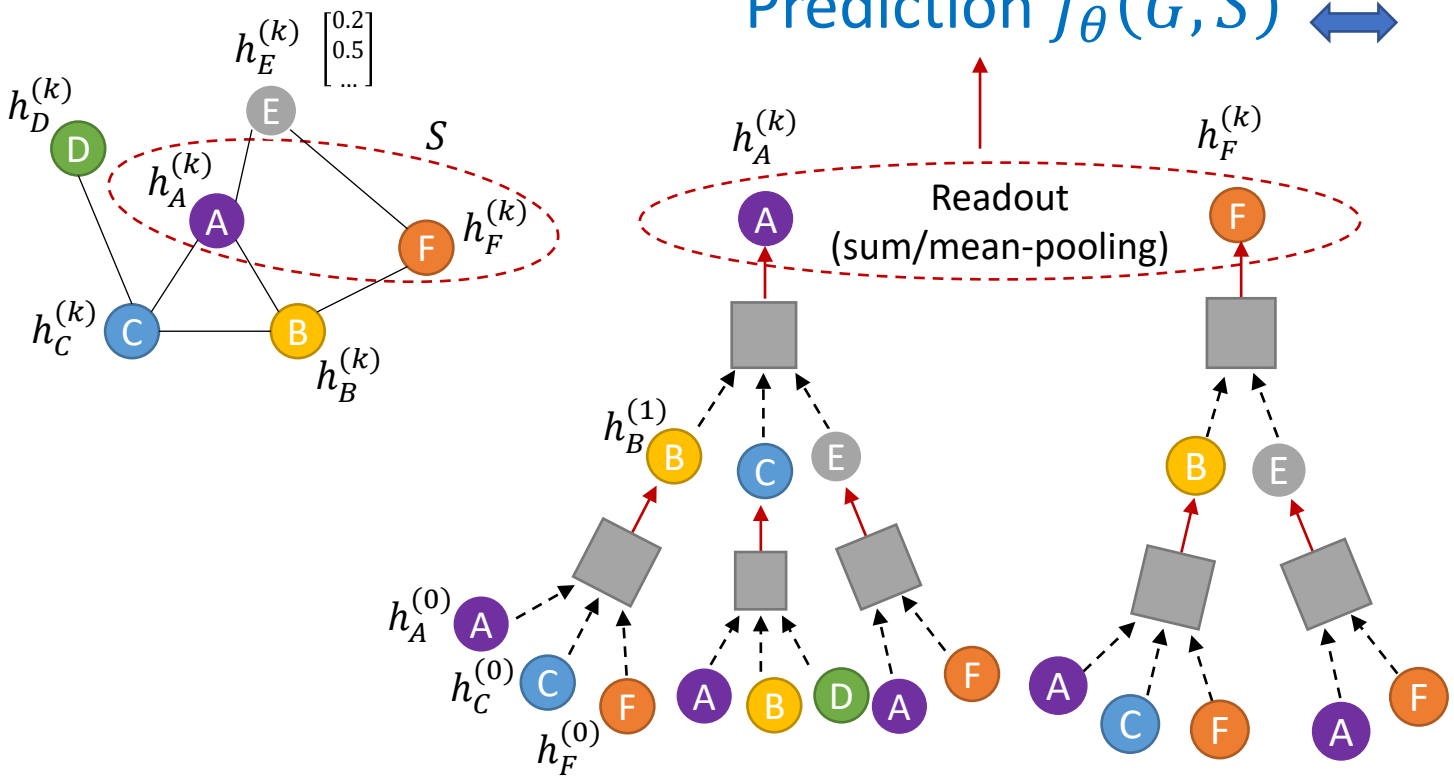
$$h_v^{(t+1)} = f_{update}\left(h_v^{(t)}, f_{agg}\left(\{h_u^{(t)} \middle| u \in N_v\}\right)\right),$$

where N_v denotes the set of the neighbors of node v.

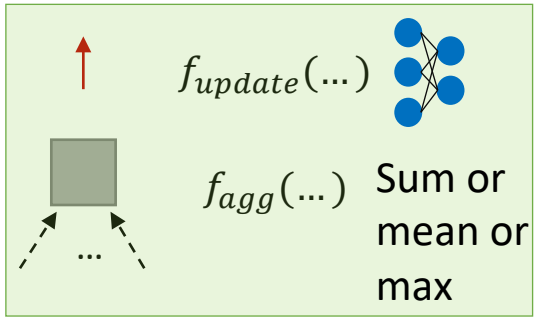
Graph Neural Networks

Input Data: A graph G, possibly with node/edge attributes.

Query: Make prediction for a set of nodes $S \subseteq V$



Compared with ground-truth labels to compute the loss and optimize the parameters



Graph neural network: one layer

Why Graph Neural Networks?

- Computationally efficient (Highly in parallel)
  - Single machine: Implemented by sparse matrix-vector multiplication

PyTorch Geometric [Fey & Lenssen 2019]

DGL [Wang 2019]

  - Distributed system (extremely large-scale graphs): Supported by the MapReduce scheme

    PinSage [Ying et al. KDD 2018]

    AliGraph [Zhu et al. VLDB 2019]

    …

Why Graph Neural Networks?

- Computationally efficient (Highly in parallel)
  - Single machine: Implemented by sparse matrix-vector multiplication

PyTorch Geometric [Fey & Lenssen 2019]

DGL [Wang 2019]

  - Distributed system (extremely large-scale graphs): Supported by the MapReduce scheme

    PinSage [Ying et al. KDD 2018]

    AliGraph [Zhu et al. VLDB 2019]

    …

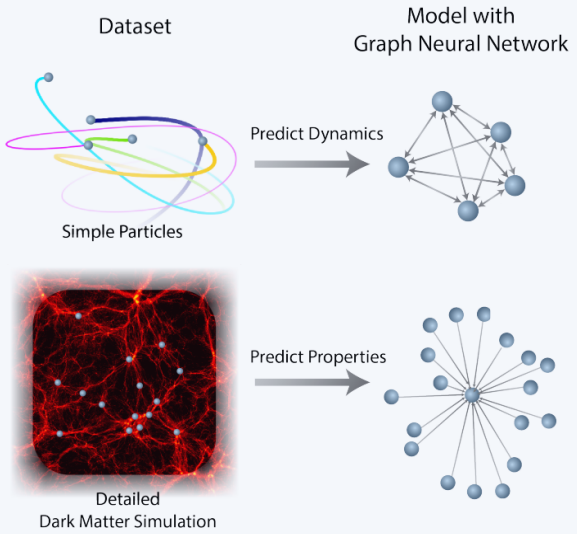
However, current system support is mainly for industry applications (1B-sized graphs, latency requirement : 0.1s ~1s)

Scientific applications: 1k-sized graphs, latency req. 1us~1ms

Why Graph Neural Networks?

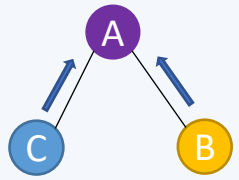
- Empirically good generalization
  - Capture inductive biases

Use graph neural functions to model physical interactions directly



[Battaglia et al. 2018]
[Cranmer et al. NeurIPS 2020]

A NN-based alternative of probabilistic models



→ Belief propagation?

B Whether is the ground wet?

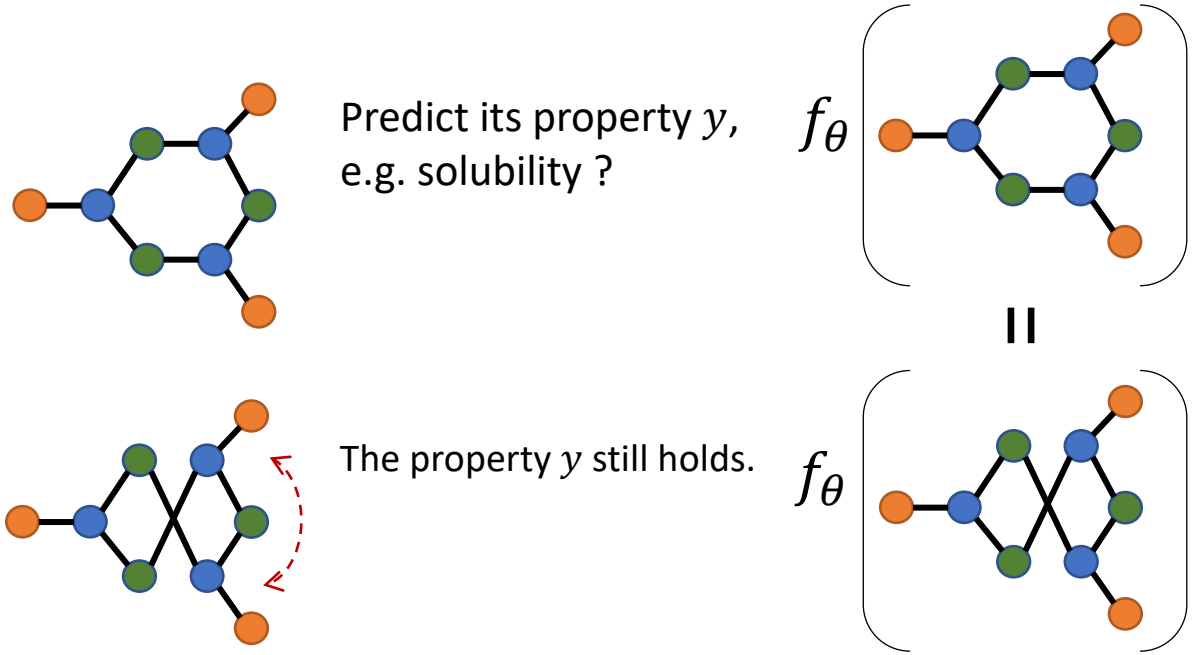
C Whether is the sky dark?

A Is it raining?

[Jia & Benson, SIMOD 2021]

Why Graph Neural Networks

- Empirically good generalization
  - Keep permutation invariance



Mathematically, $f_\theta(PAP^T) = f_\theta(A)$, A is the adjacency matrix of the input graph and P is any permutation matrix. Reduce the dimension of parameter space significantly…

Current Projects

GNN Capacity:

- Why does non-linearity help?
- What class of functions can be approximated with finite neurons?

GNN Generalization:

- When graph structures shift, can GNNs be well generalized?

System:

- Build a system for subgraph-based graph machine learning.

Applications:

- Pileup mitigation
- Tau to 3 mu
- Point cloud NN interpretation