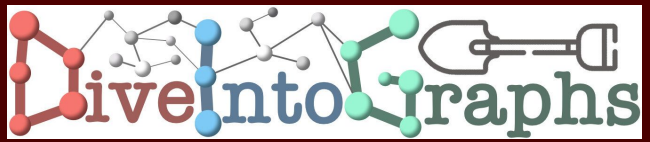
TEXAS A&M UNIVERSITY
Engineering



Frontiers of Graph Neural Networks with DIG

https://github.com/divelab/DIG/

Shuiwang Ji, Yaochen Xie, Zhao Xu, Haiyang Yu

Schedule

Graph neural networks	20 min	Shuiwang Ji
Self-supervised learning	35 min	Yaochen Xie
Practical session: *dig.sslgraph*	15 min	Zhao Xu
Q&A	10 min	All
Explainability	35 min	Haiyang Yu
Practical session: *dig.xgraph*	15 min	Haiyang Yu
Q&A	10 min	All

Success of Deep Learning

TEXAS A&M UNIVERSITY
Engineering

Deep learning shines on many tasks

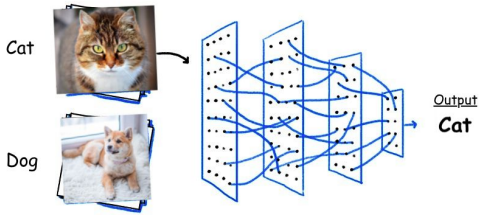
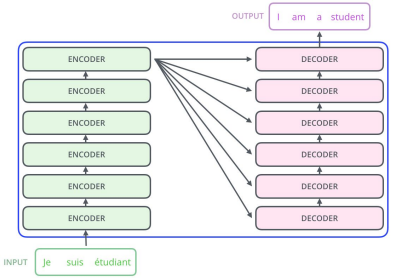
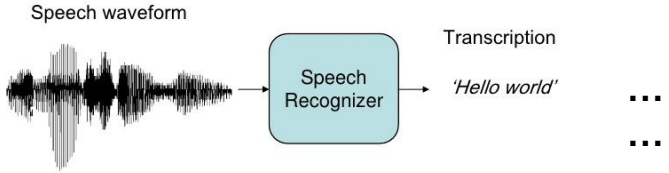


Image Classification



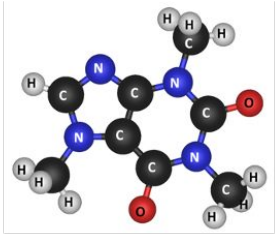
Machine Translation

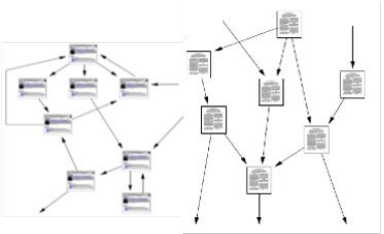


Speech recognition

Graphs

Objects + Relationships = Graph





.....

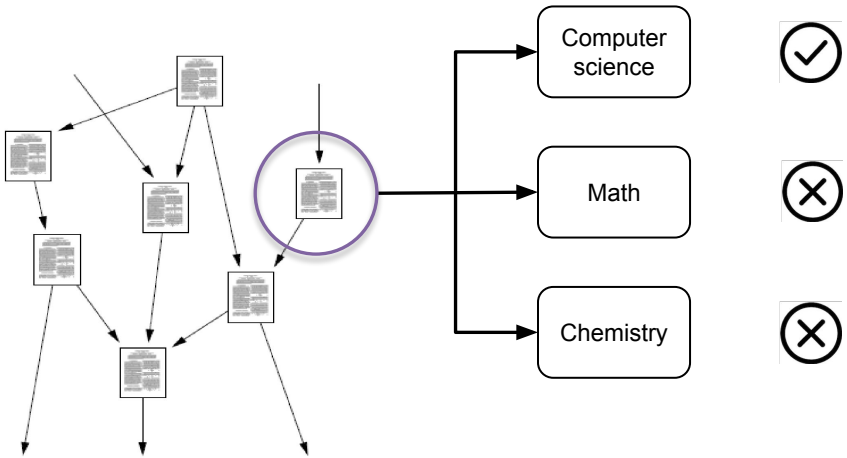
Tasks over Graphs

- Node-level classification/regression
- Graph-level classification/regression
- Link prediction

- Generation
- Explainability
- .....

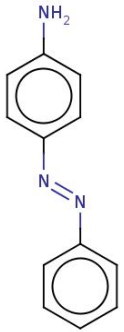
Tasks on Graph Data

- e.g., Node classification
  - Document classification in a citation network

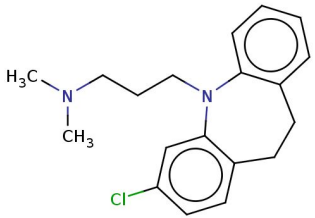


Tasks on Graph Data

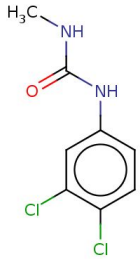
- e.g., Graph classification
  - Molecular property prediction



Toxic



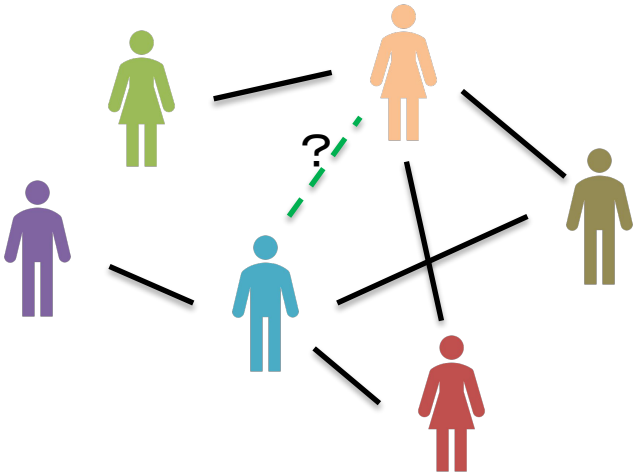
Non-toxic



?

Tasks on Graph Data

- e.g., Link prediction
  - Friend suggestion in a social network



TEXAS A&M UNIVERSITY
Engineering

Graph Neural Networks:
Basic Concepts

https://github.com/divelab/DIG/

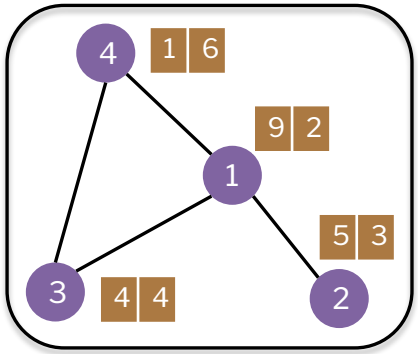
Notations

- $G = (X, A)$
- Adjacency matrix

$$A: \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \in R^{N \times N}$$

- Feature matrix

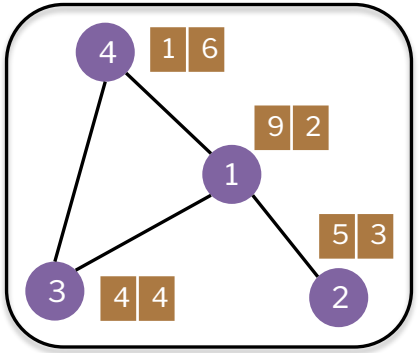
$$X: \begin{bmatrix} 9 & 2 \\ 5 & 3 \\ 4 & 4 \\ 1 & 6 \end{bmatrix} \in R^{N \times d}$$



- We will consider edge features later

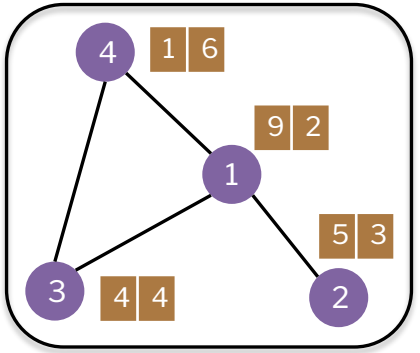
Permutation

$$X: \begin{bmatrix} 9 & 2 \\ 5 & 3 \\ 4 & 4 \\ 1 & 6 \end{bmatrix} \in R^{N \times d} \qquad A: \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \in R^{N \times N}$$

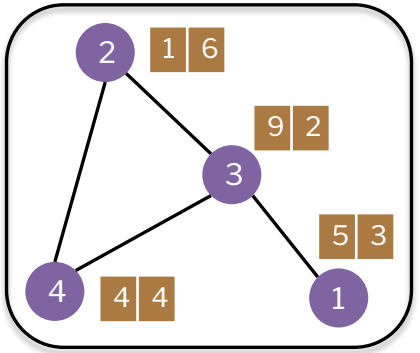


Permutation

$$X: \begin{bmatrix} 9 & 2 \\ 5 & 3 \\ 4 & 4 \\ 1 & 6 \end{bmatrix} \in R^{N \times d} \qquad A: \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \in R^{N \times N}$$



$$X': \begin{bmatrix} 5 & 3 \\ 1 & 6 \\ 9 & 2 \\ 4 & 4 \end{bmatrix} \in R^{N \times d} \qquad A': \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \in R^{N \times N}$$



Permutation

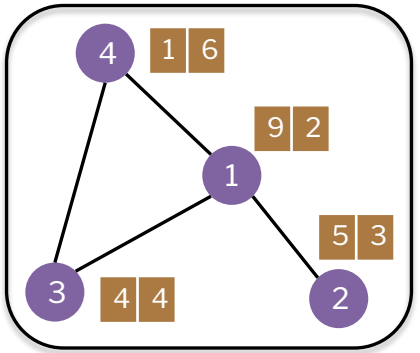
$$X: \begin{bmatrix} 9 & 2 \\ 5 & 3 \\ 4 & 4 \\ 1 & 6 \end{bmatrix} \in R^{N \times d} \qquad A: \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \in R^{N \times N}$$

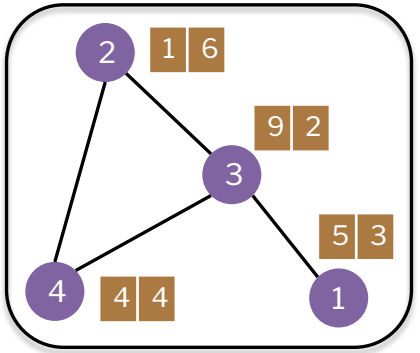
$$X' = PX; A' = PAP^T$$

Permutation matrix:

$$P: \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in \{0,1\}^{N \times N}$$

$$X': \begin{bmatrix} 5 & 3 \\ 1 & 6 \\ 9 & 2 \\ 4 & 4 \end{bmatrix} \in R^{N \times d} \qquad A': \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \in R^{N \times N}$$



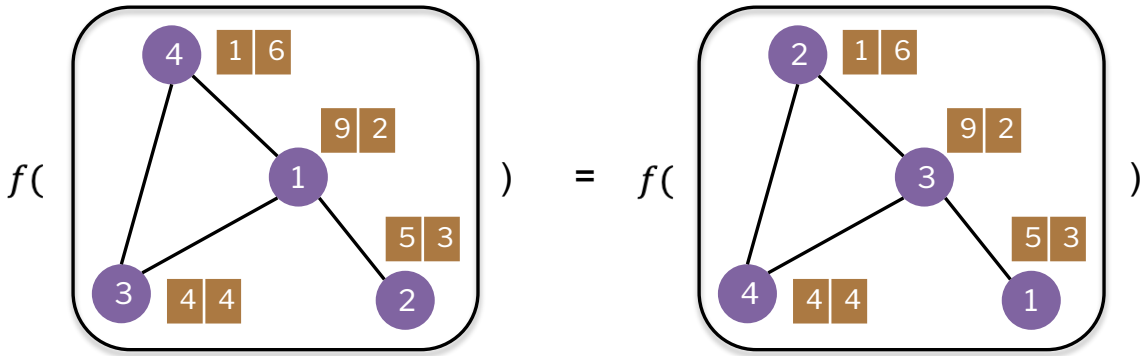


Permutation Invariance

- For graph-level representations
  - We learn a function f that maps a graph $G = (X, A)$ to a representation vector $\in \mathbb{R}^{d'}$

$$f(X, A) = f(PX, PAP^T)$$

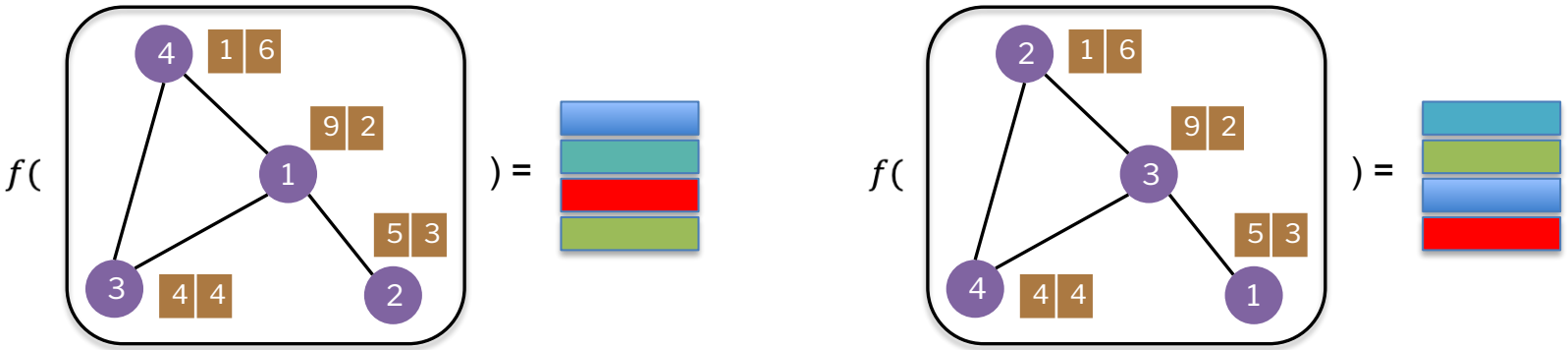
  - P is any permutation matrix



Permutation Equivariance

- For node-level representations
  - We learn a function f that maps a graph $G = (X, A)$ to a node representation matrix $\in \mathbb{R}^{N \times d'}$

$$Pf(X, A) = f(PX, PAP^T)$$



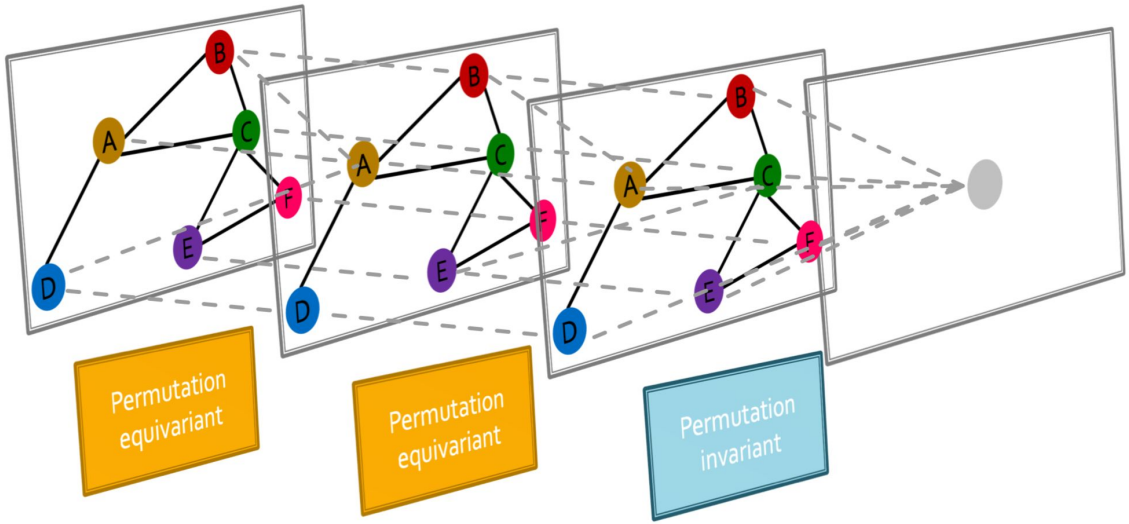
TEXAS A&M UNIVERSITY
Engineering

Graph Neural Networks

Partially based on materials at:
https://geometricdeeplearning.com/lectures/

Overview

- GNNs are composed of multiple permutation equivariant/invariant layers/functions

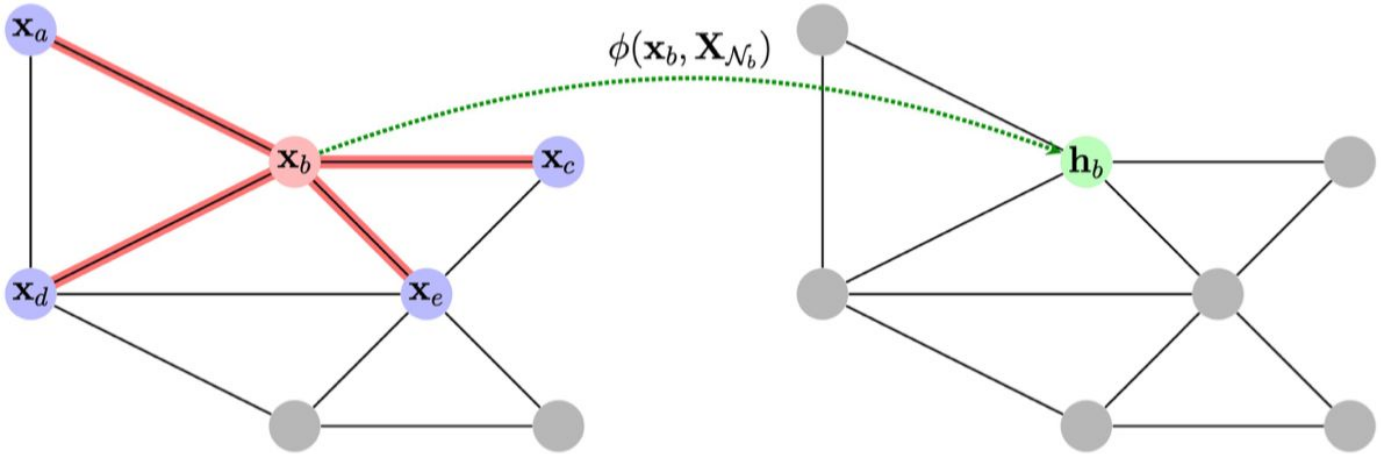


Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

Idea

- Neighborhood aggregation!
  - Neighborhood features $X_{\mathcal{N}_i} = \{\{x_j : j \in \mathcal{N}_i\}\}$
  - Define a local function $\phi(x_i, X_{\mathcal{N}_i})$

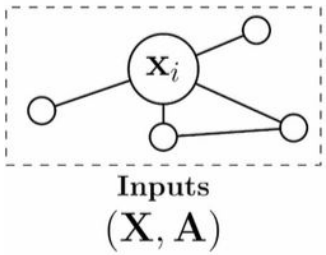
$$H = f(X, A) = \begin{bmatrix} \phi(x_1, X_{\mathcal{N}_1}) \\ \dots \\ \phi(x_N, X_{\mathcal{N}_N}) \end{bmatrix}$$



Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

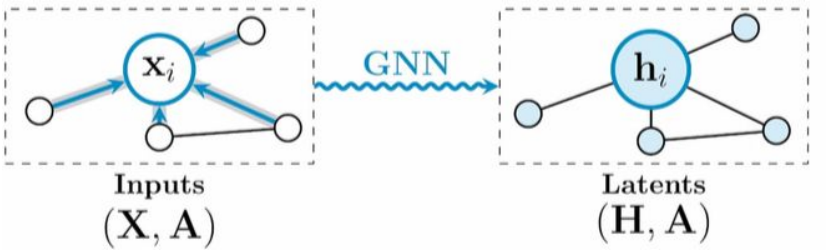
Blueprint for Learning on Graphs

TEXAS A&M UNIVERSITY
Engineering



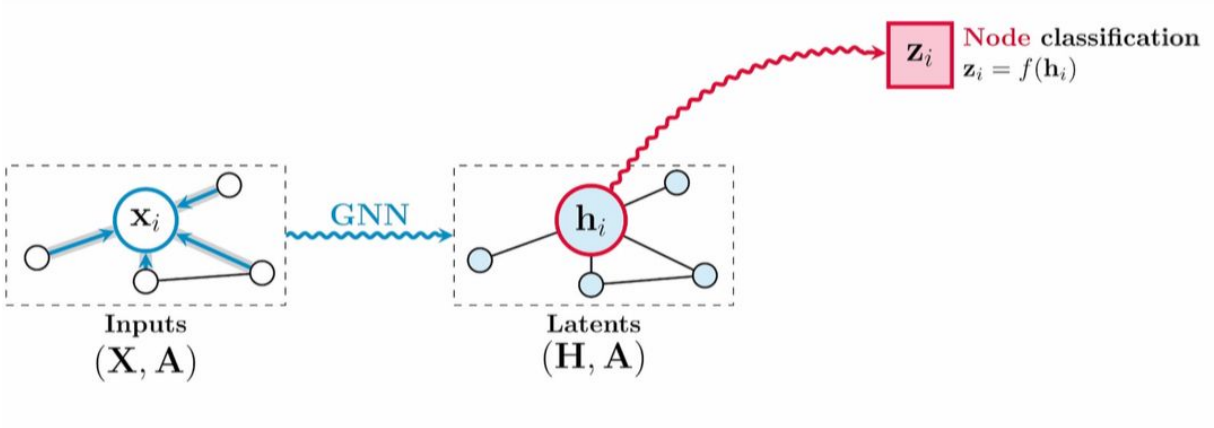
Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

Blueprint for Learning on Graphs



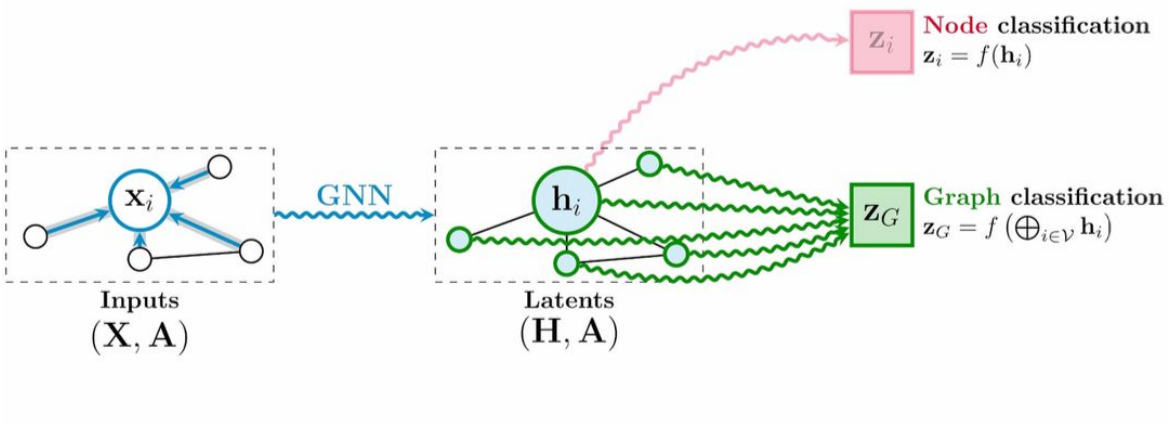
Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

Blueprint for Learning on Graphs



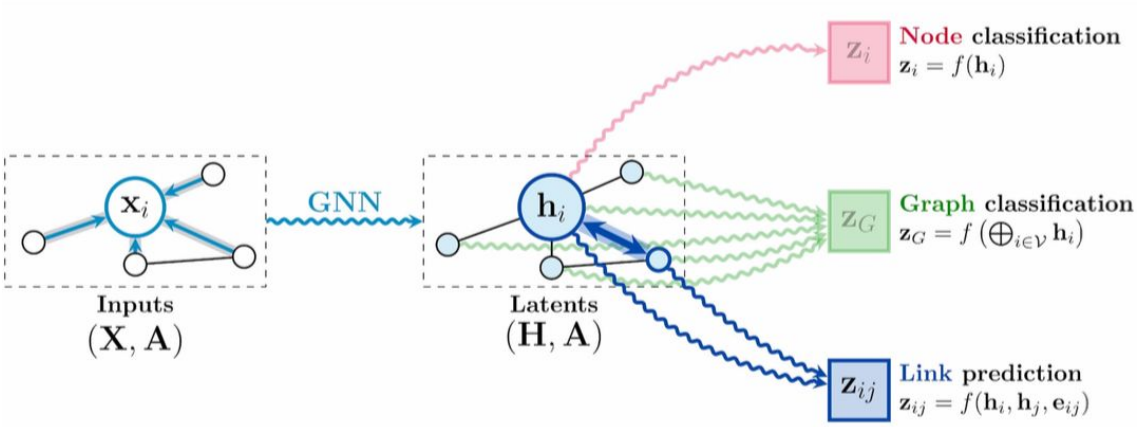
Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

Blueprint for Learning on Graphs



Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

Blueprint for Learning on Graphs

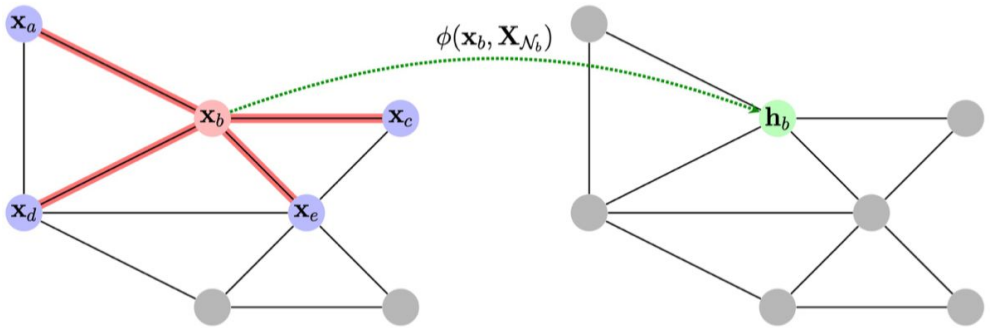


Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

What's in a GNN layer

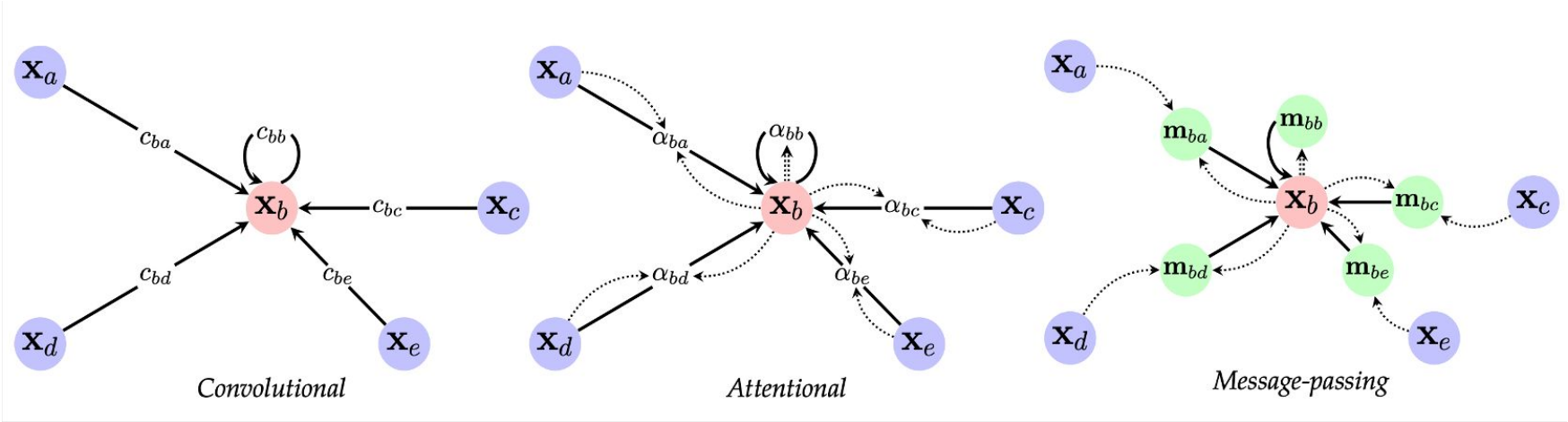
- Neighborhood aggregation!
  - We build permutation equivariant functions $f(X, A)$ on graphs by sharing the local permutation invariant function $\phi(x_i, X_{\mathcal{N}_i})$
  - How to implement ϕ?

$$H = f(X, A) = \begin{bmatrix} \phi(x_1, X_{\mathcal{N}_1}) \\ \dots \\ \phi(x_N, X_{\mathcal{N}_N}) \end{bmatrix}$$



Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

Three “flavours” of GNN layers



$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} c_{uv}\psi(\mathbf{x}_v)\right)$$

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} a(\mathbf{x}_u, \mathbf{x}_v)\psi(\mathbf{x}_v)\right)$$

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v)\right)$$

Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

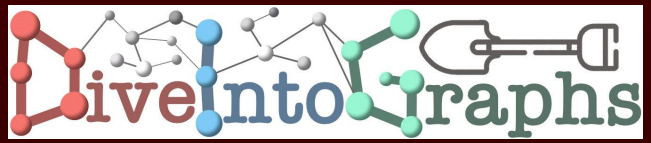
Frontier Topics of GNN

- Self-supervised learning of GNNs
  - Studies how to train GNNs with unlabeled graph data
  - Applications: pre-training in drug discovery, node representation learning for industrial large-scale graphs.
- GNN explainability
  - Studies the cause of GNN predictions
  - Applications: building trustworthy and transparent GNN models
- Graph generation, 3D geometric GNNs, etc.

TEXAS A&M UNIVERSITY
Engineering

Self-Supervised Learning on Graphs

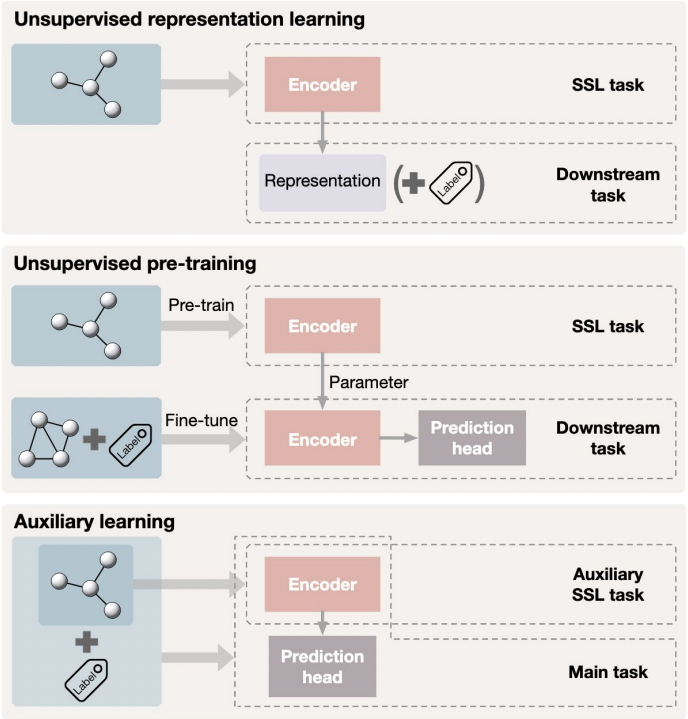
https://github.com/divelab/DIG/



Self-supervised Learning

- Training models with self-supervision.
- Success in text and image data. We see an explosion of graph SSL papers.

- Three paradigms of using self-supervised learning
  - Unsupervised representation learning
  - Unsupervised pre-training
  - Auxiliary learning

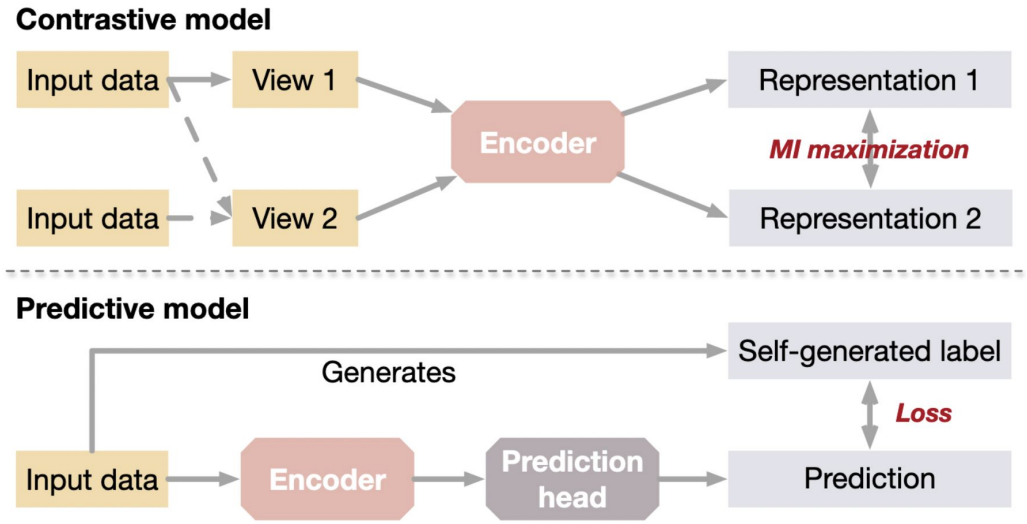


Xie et al. Self-Supervised Learning of Graph Neural Networks: A Unified Review. TPAMI, 2022.

Taxonomy of SSL Methods

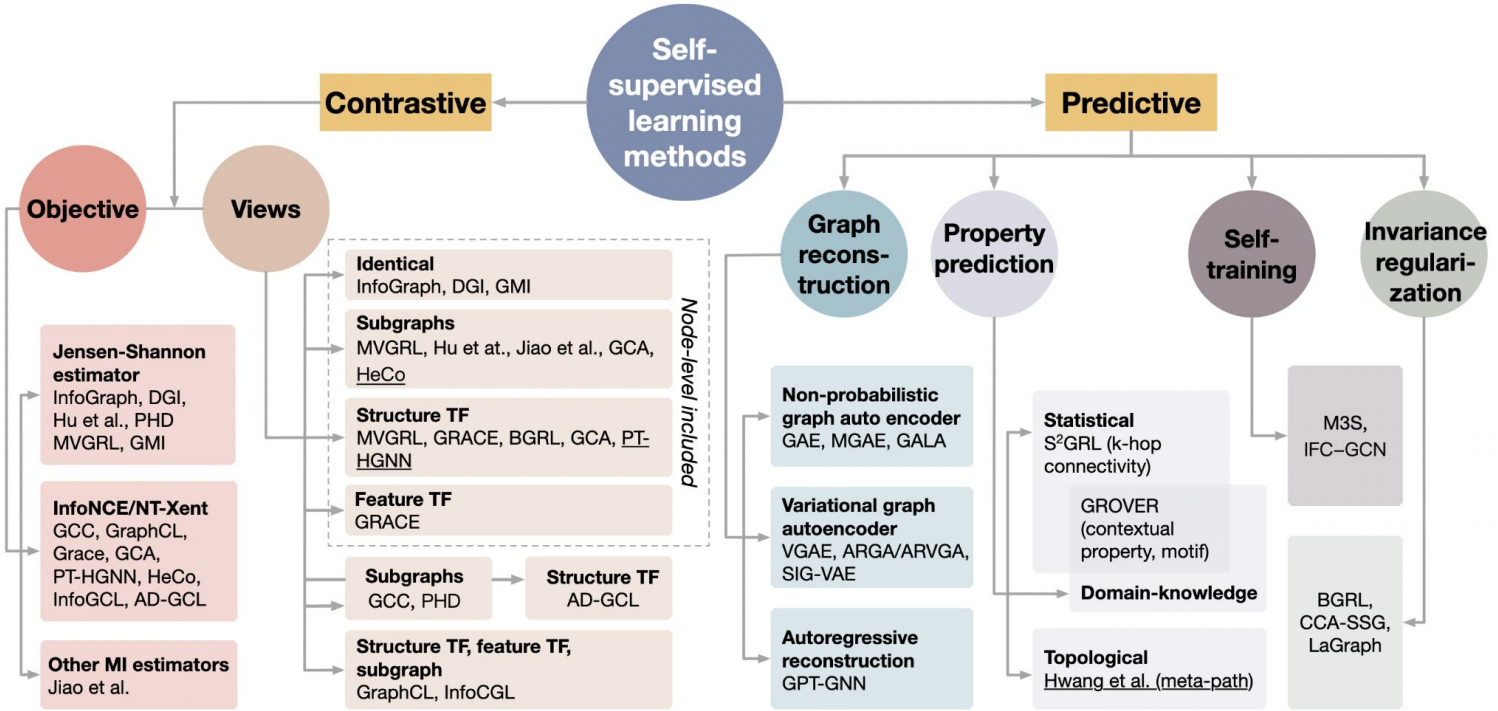
Contrastive vs Predictive: are negative pairs required?

- Contrastive
  - Maximizes mutual information
  - Focuses on designing view generation
- Predictive
  - Ad-hoc pretext-tasks / information bottleneck
  - Focuses on designing task/objective



Xie et al. Self-Supervised Learning of Graph Neural Networks: A Unified Review. TPAMI, 2022.

Taxonomy of SSL Methods



Xie et al. Self-Supervised Learning of Graph Neural Networks: A Unified Review. TPAMI, 2022.

Contrastive Methods

- Representations of “similar” graph data to be closer, and that of “dissimilar” graph data to be further from each other.

- In practice, we do not have ground-truth or defined measurement for similarity. We construct augmented graphs (or subgraphs) from the same graph to be “similar”, and different graph samples are “dissimilar”.

Contrastive Methods

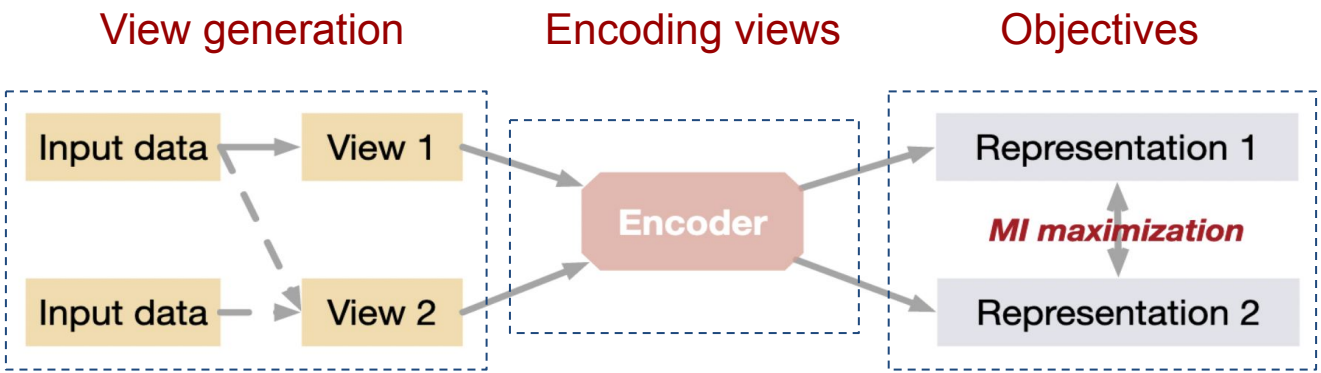
- Representations of “similar” graph data to be closer, and that of “dissimilar” graph data to be further from each other.

- Theoretical grounding: mutual information maximization. Jointly sampled graphs are “similar”, while independently sampled graphs are “dissimilar”

$$\mathcal{I}(\boldsymbol{x}, \boldsymbol{y}) = D_{KL}(p(\boldsymbol{x}, \boldsymbol{y}) || p(\boldsymbol{x})p(\boldsymbol{y}))$$

  - We consider a graph as a random variable and obtain two views of the graph.
  - Representations of the two views are x and y. A good GNN should encode two views into representation that share as much mutual information as possible.

Contrastive Methods



Xie et al. Self-Supervised Learning of Graph Neural Networks: A Unified Review. TPAMI, 2022.

Contrastive Objectives

- Derived as lower bounds of mutual information
  - Jensen-Shannon Estimator (DGI, InfoGraph, MVGRL, etc.)

$$\widehat{\mathcal{I}}^{(JS)}(\boldsymbol{h}_i,\boldsymbol{h}_j) = \mathbb{E}_{(\boldsymbol{A},\boldsymbol{X})\sim\mathcal{P}}\left[\log(\mathcal{D}(\boldsymbol{h}_i,\boldsymbol{h}_j))\right] + \mathbb{E}_{[(\boldsymbol{A},\boldsymbol{X}),(\boldsymbol{A}',\boldsymbol{X}')]\sim\mathcal{P}\times\mathcal{P}}\left[\log(1-\mathcal{D}(\boldsymbol{h}_i,\boldsymbol{h}'_j))\right]$$

  - InfoNCE (GraphCL, GRACE, etc.)

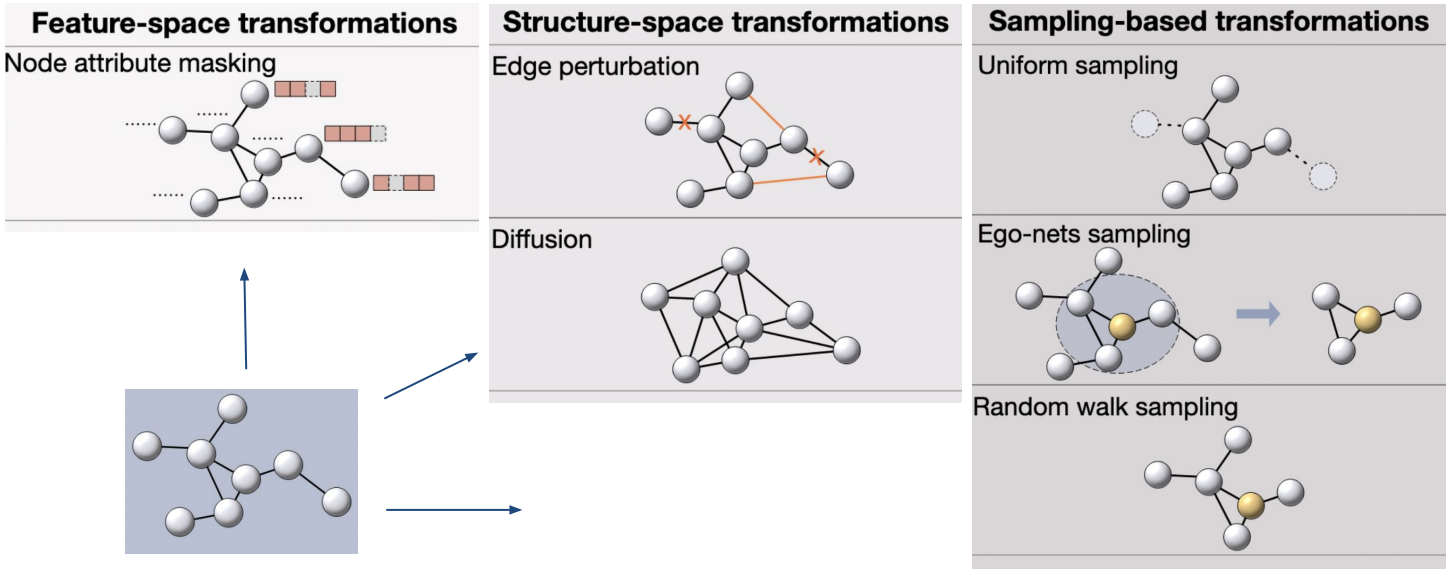
$$\widehat{\mathcal{I}}^{(\mathrm{NCE})}(\boldsymbol{h}_i,\boldsymbol{h}_j) = \mathbb{E}_{(\boldsymbol{A},\boldsymbol{X})\sim\mathcal{P}}\left[\mathcal{D}(\boldsymbol{h}_i,\boldsymbol{h}_j) - \mathbb{E}_{\boldsymbol{K}\sim\mathcal{P}^N}\left[\log\sum_{(\boldsymbol{A}',\boldsymbol{X}')\in\boldsymbol{K}} e^{\mathcal{D}(\boldsymbol{h}_i,\boldsymbol{h}'_j)}/N \,\middle|\, (\boldsymbol{A},\boldsymbol{X})\right]\right]$$

$$= \mathbb{E}_{[(\boldsymbol{A},\boldsymbol{X}),\boldsymbol{K}]\sim\mathcal{P}\times\mathcal{P}^N}\left[\log\frac{e^{\mathcal{D}(\boldsymbol{h}_i,\boldsymbol{h}_j)}}{\sum_{(\boldsymbol{A}',\boldsymbol{X}')\in\boldsymbol{K}} e^{\mathcal{D}(\boldsymbol{h}_i,\boldsymbol{h}'_j)}}\right] + \log N$$

Xie et al. Self-Supervised Learning of Graph Neural Networks: A Unified Review. TPAMI, 2022.

View generation

- Three types of graph transformations for view generation

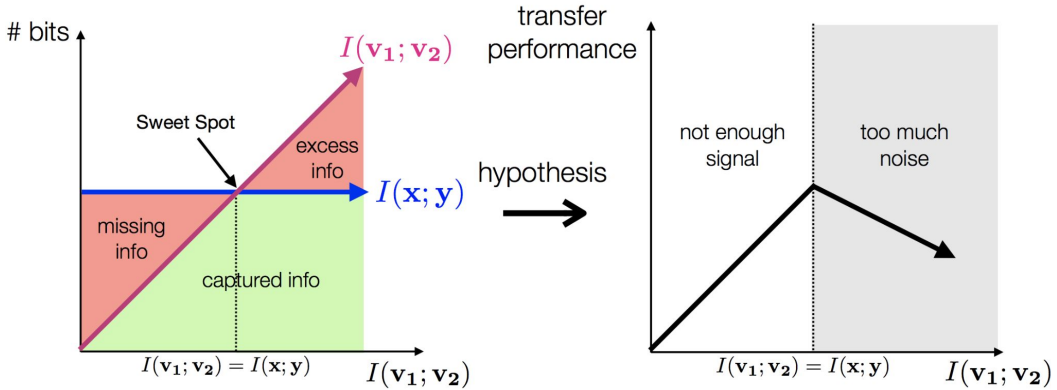


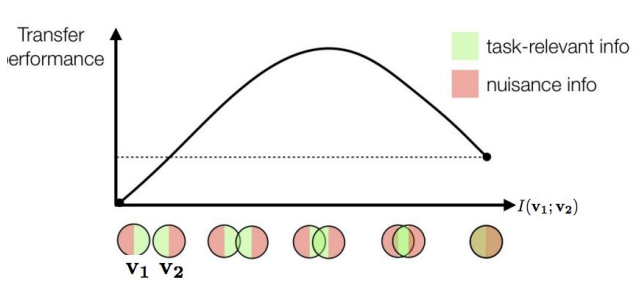
Xie et al. Self-Supervised Learning of Graph Neural Networks: A Unified Review. TPAMI, 2022.

What are good views for graphs?

TEXAS A&M UNIVERSITY Engineering

- Ideally, good views (v_1, v_2) generated for contrastive learning should
  - Have their mutual information $I(v_1, v_2)$ minimized
  - Subject to $I(v_1, y) = I(v_2, y) = I(x, y)$





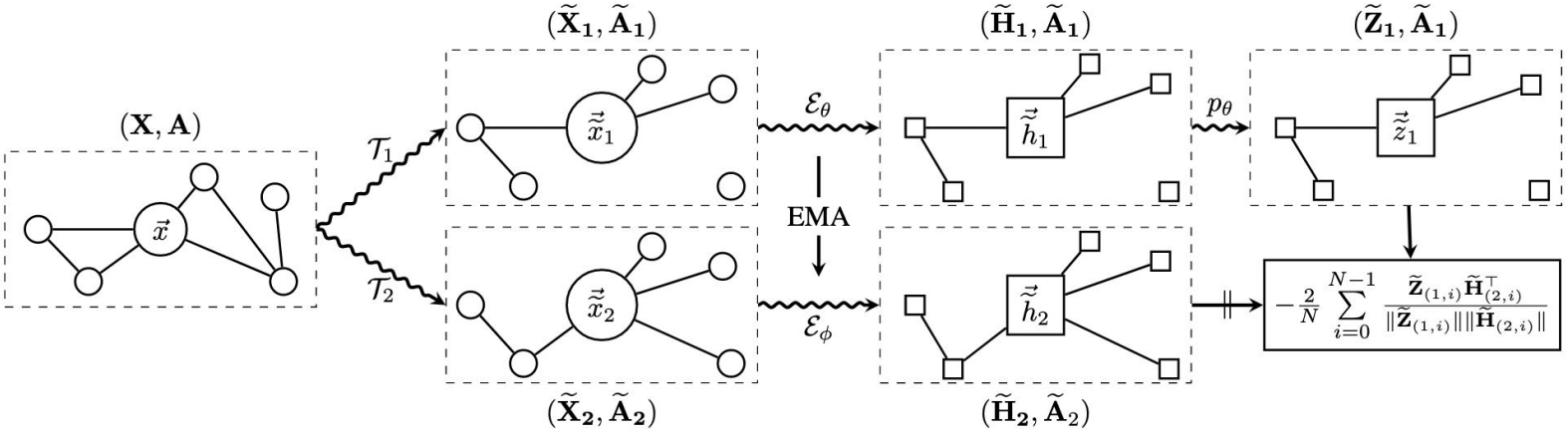
- The principle is even more important for graphs.

Tian et al. What Makes for Good Views for Contrastive Learning? NeurIPS 2020

Predictive - Invariance Regularization

TEXAS A&M UNIVERSITY Engineering

- BGRL



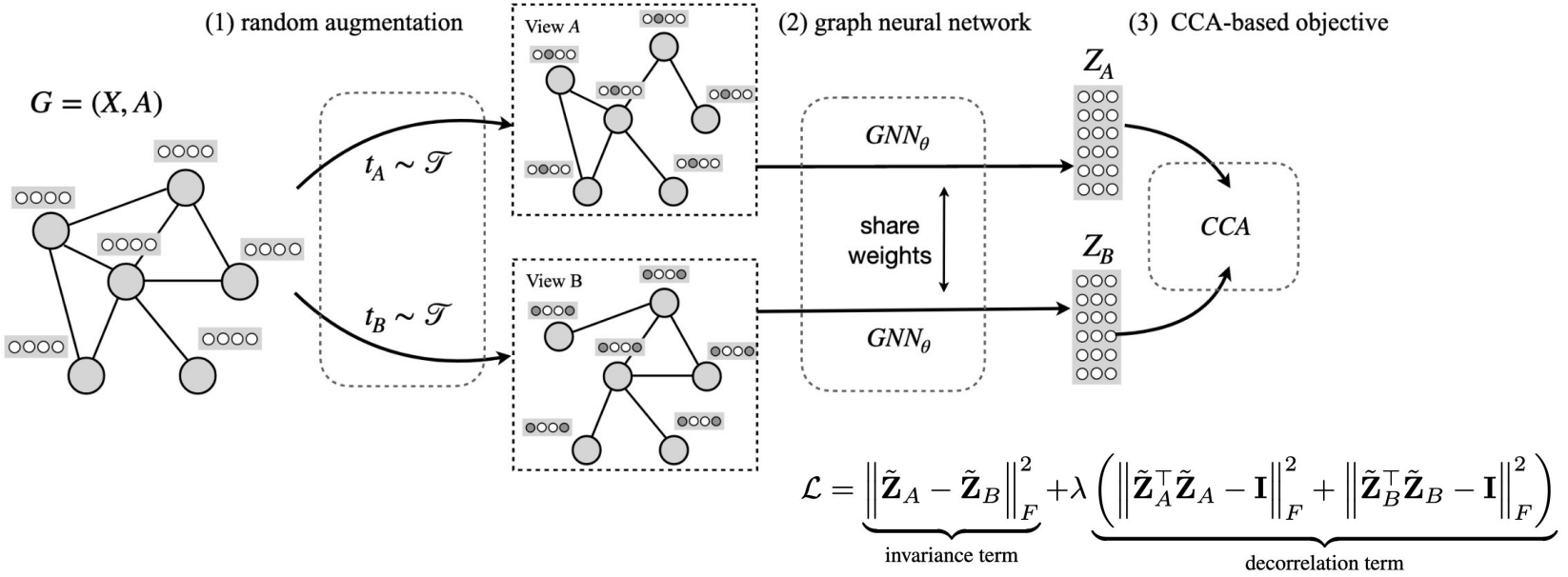
$\theta \leftarrow \text{optimize}(\theta,\ \eta,\ \partial_\theta \ell(\theta, \phi))$ EMA update: $\phi \leftarrow \tau\phi + (1-\tau)\theta$

Thakoor et al., Large-Scale Representation Learning on Graphs via Bootstrapping. ICLR 2022.

Predictive - Invariance Regularization

TEXAS A&M UNIVERSITY Engineering

- CCA-SSG

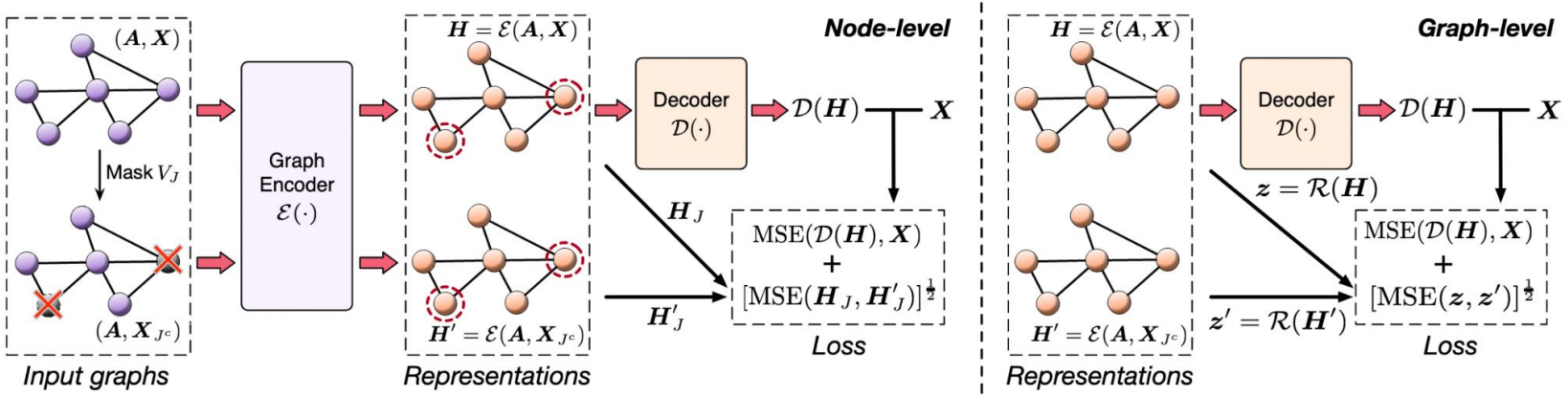


Zhang et al., From Canonical Correlation Analysis to Self-supervised Graph Neural Networks. NeurIPS 2021.

Predictive - Invariance Regularization

TEXAS A&M UNIVERSITY Engineering

- LaGraph



$$L_{node}(\mathcal{E}, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} \|\mathcal{D}(\boldsymbol{A}_i, \boldsymbol{H}_i) - \boldsymbol{X}_i\|^2 + \alpha \left[\frac{\sum_i \|\mathbb{1}_{J_i} \odot \boldsymbol{H}_i - \mathbb{1}_{J_i} \odot \boldsymbol{H}'_i\|^2}{\sum_i |J_i|}\right]^{1/2} + \alpha' \left[\sum_i \|\boldsymbol{z}_i - \boldsymbol{z}'_i\|^2 / \sum_i |J_i|\right]^{1/2}$$

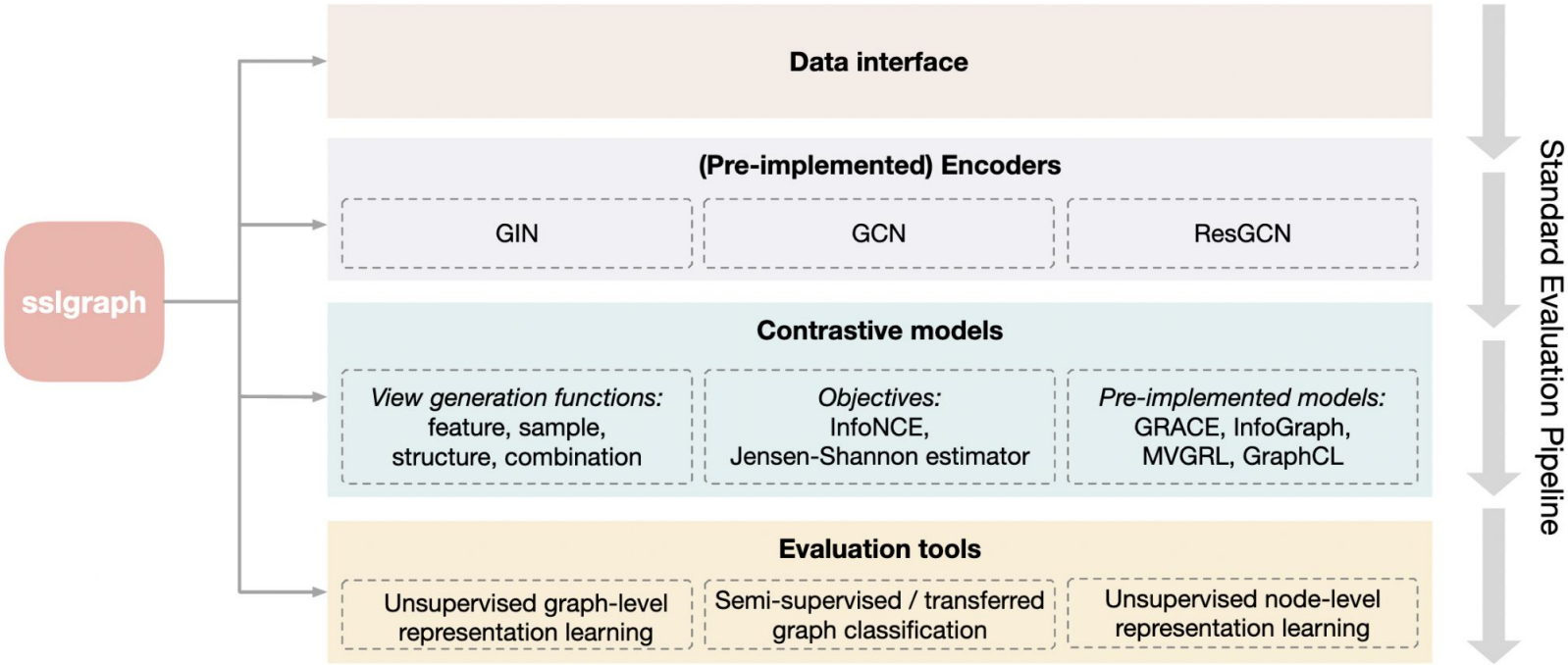
Xie et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML 2022.

Predictive - Invariance Regularization

- Similarity
  - Do not requires negative samples.
  - Better scalability.
  - Their objectives include a term to regularize the invariance of representations to different views (minimize their distance)
  - Effectiveness can be justified by the Information Bottleneck Principle

- Difference
  - The formulation of entire objectives are different.
  - Based on or derived from different theoretical grounding/insights
  - Different technique to guarantee informative representations

DIG.sslgraph



TEXAS A&M UNIVERSITY
Engineering

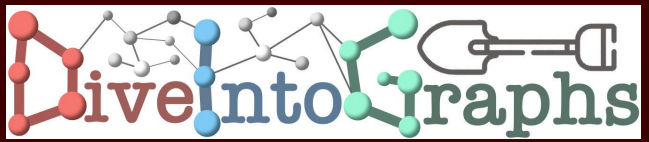
Q&A

Next session:
Explainability of GNNs

TEXAS A&M UNIVERSITY
Engineering

Explainability of GNNs

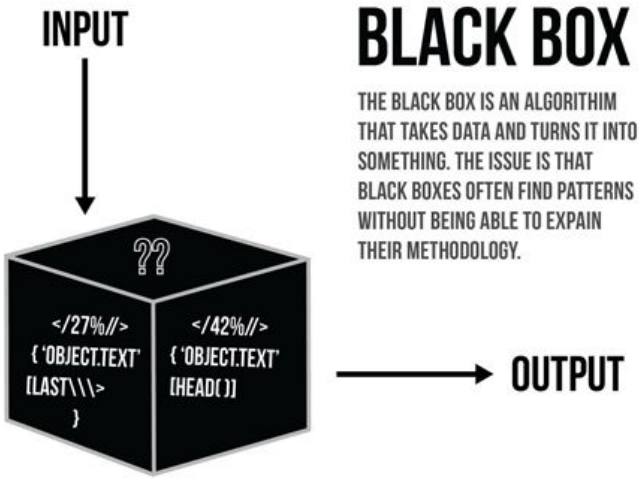
https://github.com/divelab/DIG/



Why Explainability

Good performance but

- People don't understand deep model
  - taking it as black box
- People don't trust deep model
  - when the model fails



Explainability:

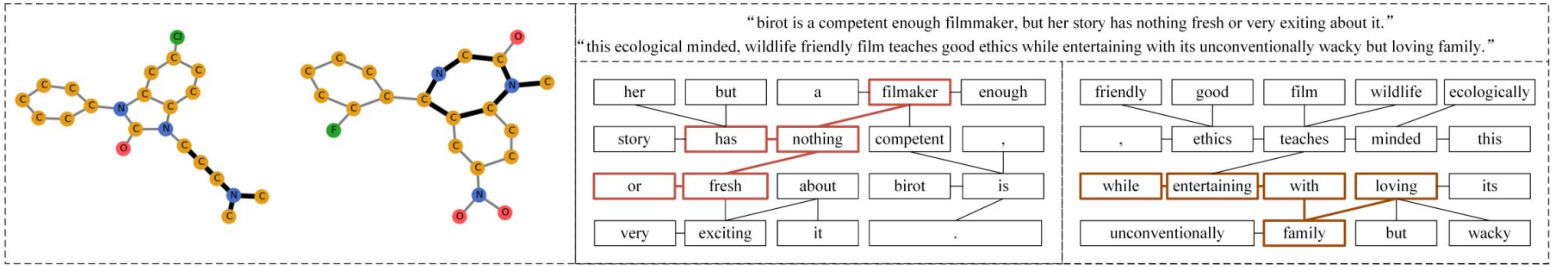
- Provide human-intelligible explanations to build trustworthy AI

Reference: https://towardsdatascience.com/guide-to-interpretable-machine-learning-d40e8a64b6cf

Challenges

- How to explain GNNs with edge information
  - Instance-level: edge-based, subgraph-based
  - Model-level: xgnn

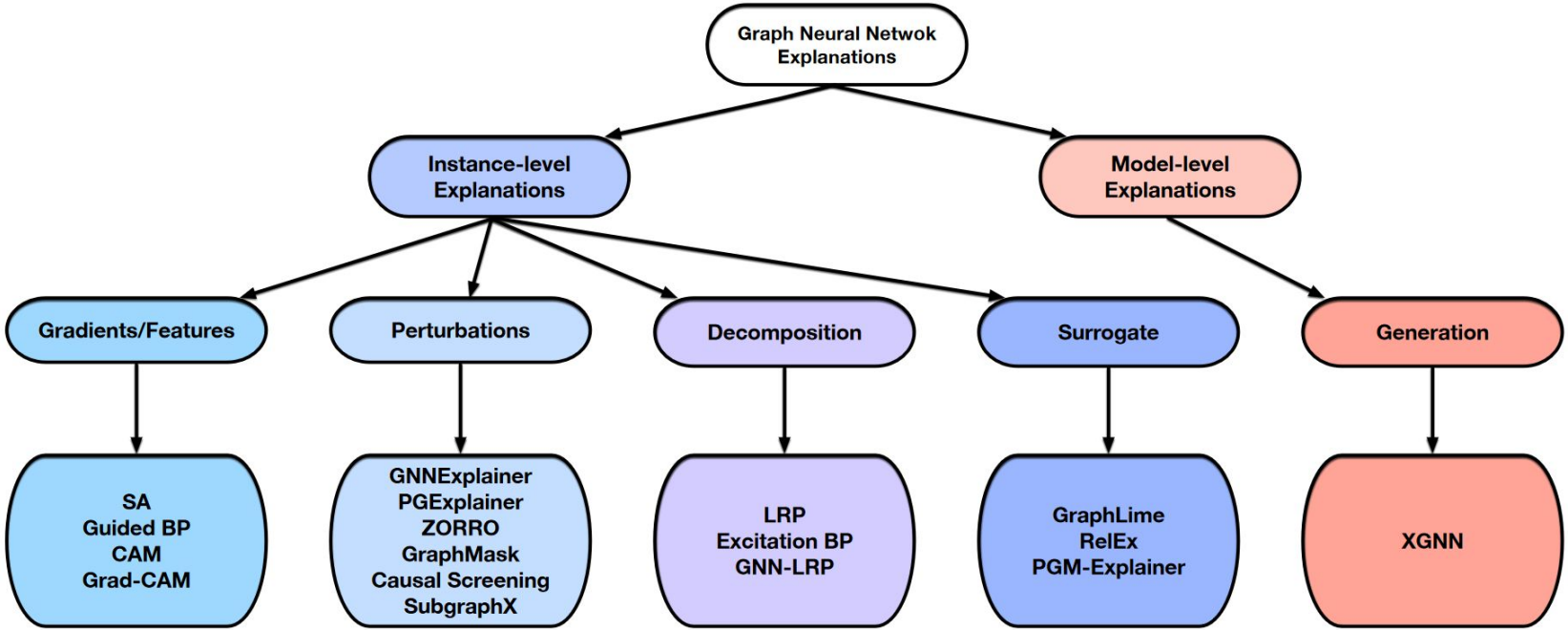
- How to get human-understandable explanation results
  - Graph Sentiment dataset



- How to compare explainability methods systematically
  - Quantitative metrics: Fidelity+, Fidelity-, Sparsity

Yuan et al. Explainability in Graph Neural Networks: A Taxonomic Survey. TPAMI 2022.

Overview of methods

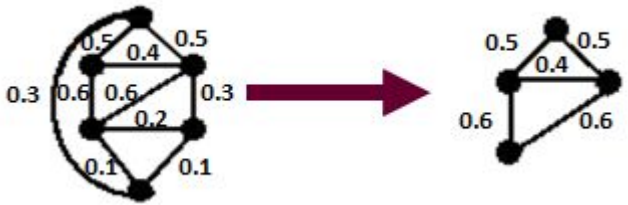


Yuan et al. Explainability in Graph Neural Networks: A Taxonomic Survey. TPAMI, 2022

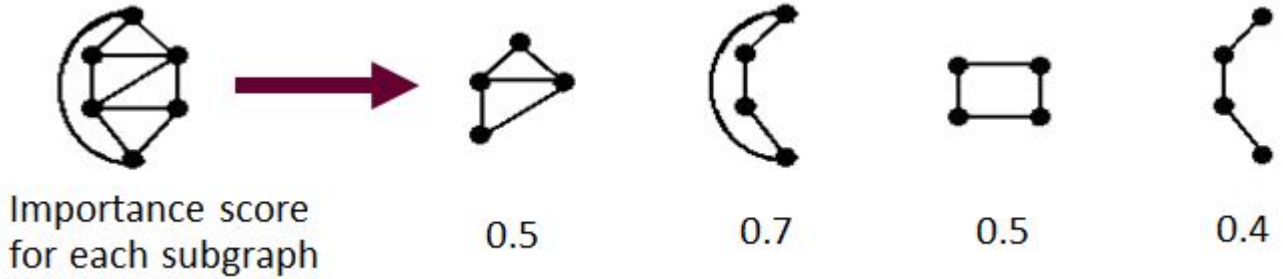
Explanation on GNNs

TEXAS A&M UNIVERSITY
Engineering

Assigning importance scores on edges

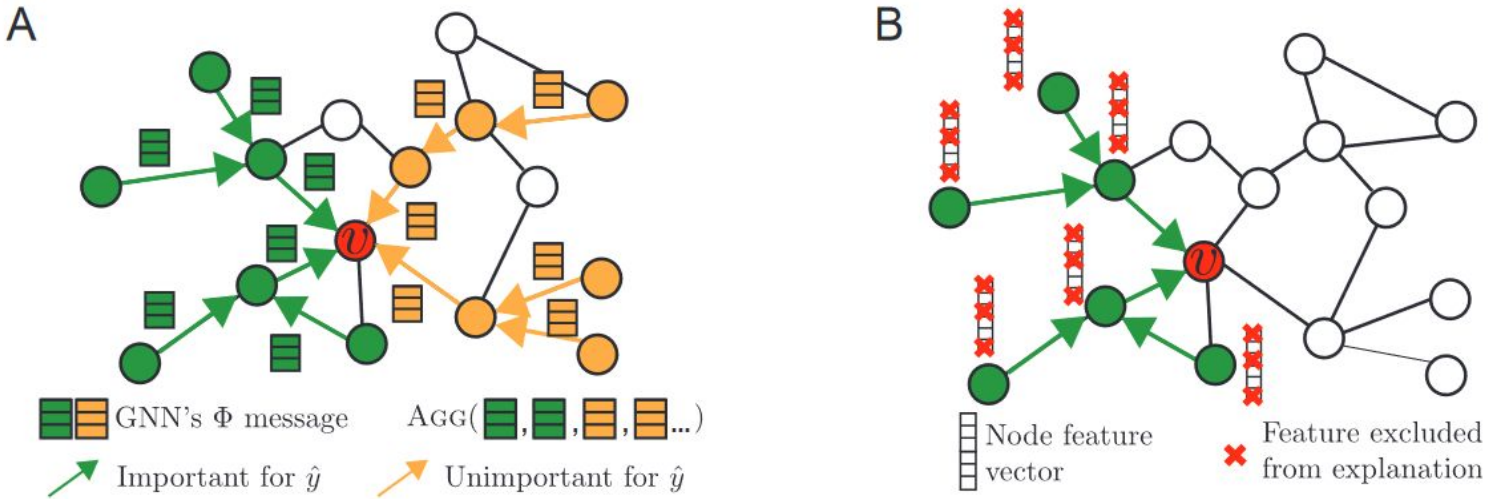


Assigning importance scores on connected subgraphs



GNNExplainer

A: Edge masks for class-distinguish important edges
B: Node masks for important node feature dimension.



Ying et al. GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS, 2019.

GNNExplainer

Explanation based on edges

$$\min_M -\sum_{c=1}^{C} \mathbf{1}[y=c] \log P_\Phi(Y=y \mid G=A_c \odot \sigma(M_E), X=X_c \odot \sigma(M_X))$$

- It learns soft masks on graph edges and node features.
- The masks are randomly initialized and updated to maximum the mutual information between the original predictions and the perturbed graphs.

Ying et al. GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS, 2019.

SubgraphX

Explanation based on subgraphs

$$\mathcal{G}^* = \underset{|\mathcal{G}_i| \leq N_{\min}}{\operatorname{argmax}} \operatorname{Score}(f(\cdot), \mathcal{G}, \mathcal{G}_i)$$

- Explore the important subgraphs with Monte Carlo tree search algorithm

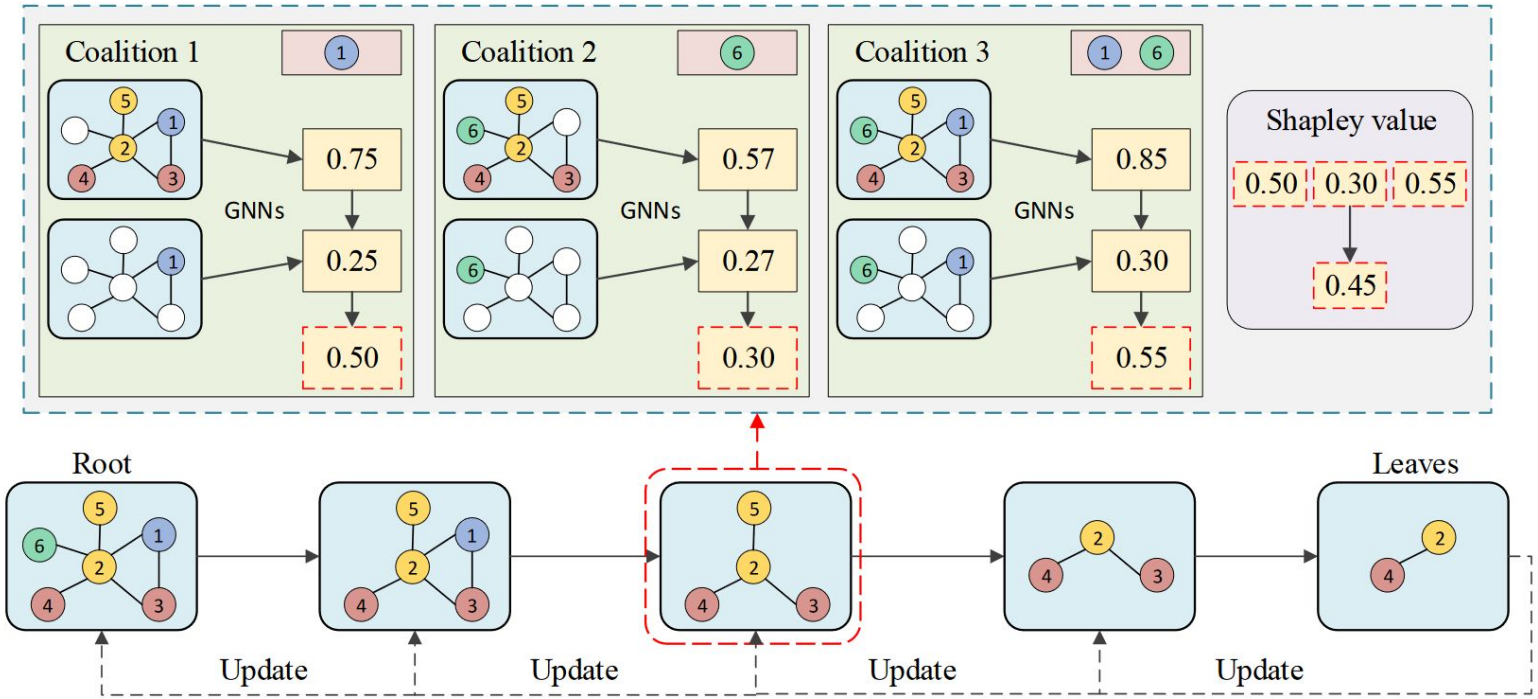
$$a^* = \underset{a_j}{\operatorname{argmax}} Q(\mathcal{N}_i, a_j) + U(\mathcal{N}_i, a_j)$$

- Take Shapley value as the subgraph importance scores

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S))$$

Yuan et al. On Explainability of Graph Neural Networks via Subgraph Explorations. ICML 2021.

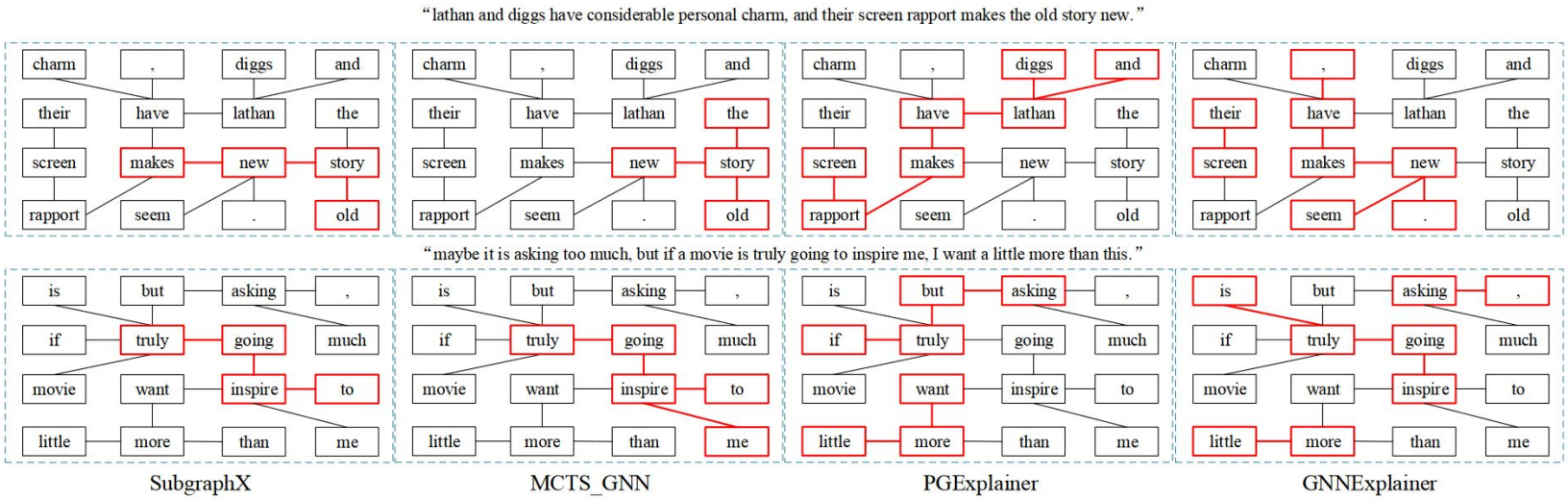
SubgraphX



Yuan et al. On Explainability of Graph Neural Networks via Subgraph Explorations. ICML 2021.

Sentiment Graph Dataset

- Better human evaluations for the explainabiltiy results
  - Graph-SST2, Graph-SST5, Graph-Twitter

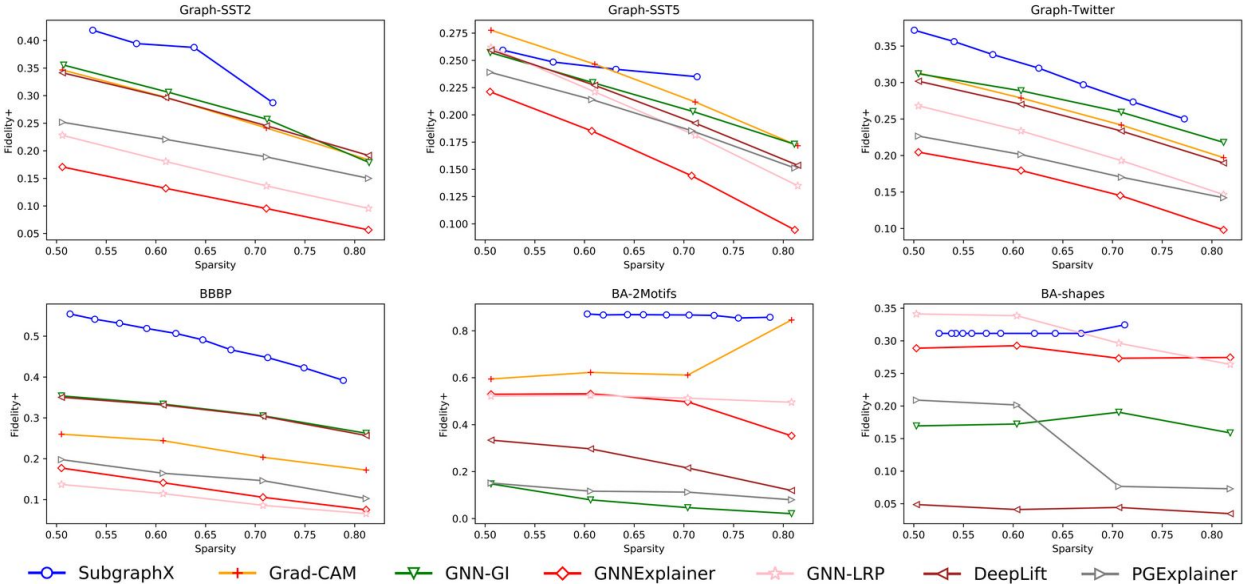


Yuan et al. Explainability in Graph Neural Networks: A Taxonomic Survey. TPAMI 2022.
Yuan et al. On Explainability of Graph Neural Networks via Subgraph Explorations. ICML 2021.

Evaluation Metrics

- Compare different explainability methods systematically
  - Fidelity+, Fidelity-, Sparisty

Benchmark of explainability methods using Fidelity and Sparsity



Yuan et al. Explainability in Graph Neural Networks: A Taxonomic Survey. TPAMI 2022.

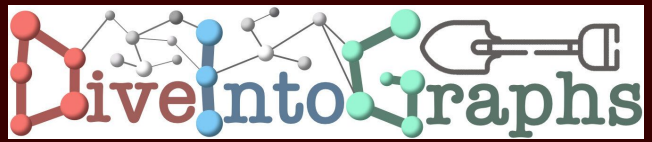
TEXAS A&M UNIVERSITY
Engineering

Q&A

DIG

xgraph
benchmark

Thank you!



TEXAS A&M UNIVERSITY
Engineering

Q&A