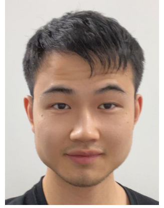
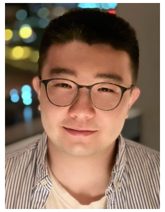
Data Quality-Aware Graph Machine Learning


Yu Wang[1,6]


Yijun Tian[2]


Tong Zhao[3]


Xiaorui Liu[4]

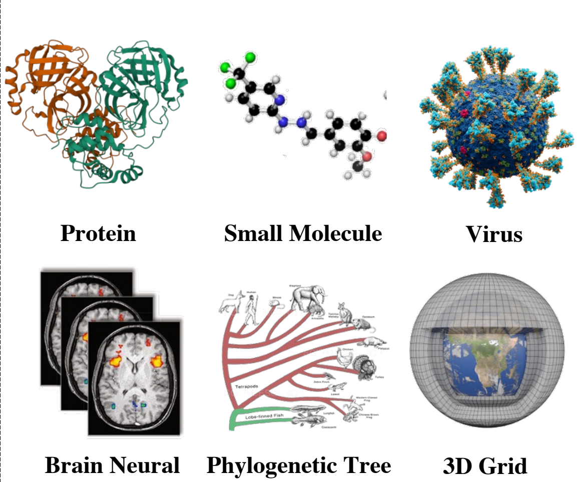

Jian Kang[5]


Tyler Derr[1]

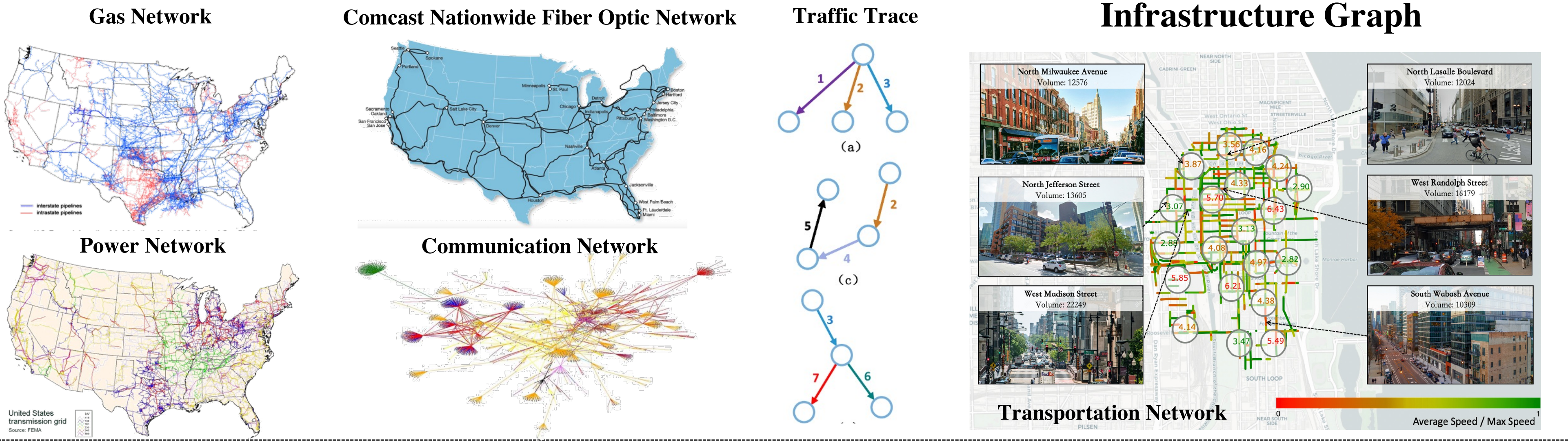
Vanderbilt University[1]
Univeresity of Notre Dame[2]
Snap Research[3]
North Carolina State University[4]
University of Rochester[5]
University of Oregon[6]

Introduction and Background - Graph-Structured Data is Everywhere

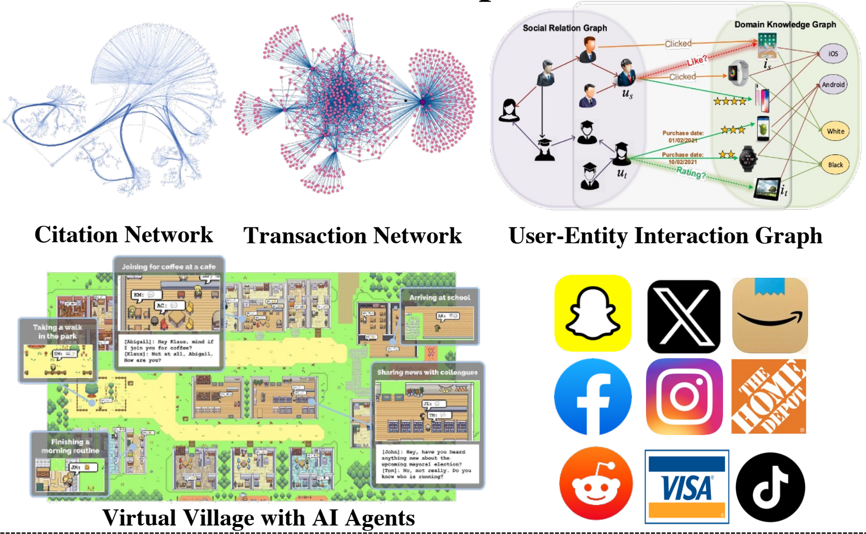
Scientific Graph



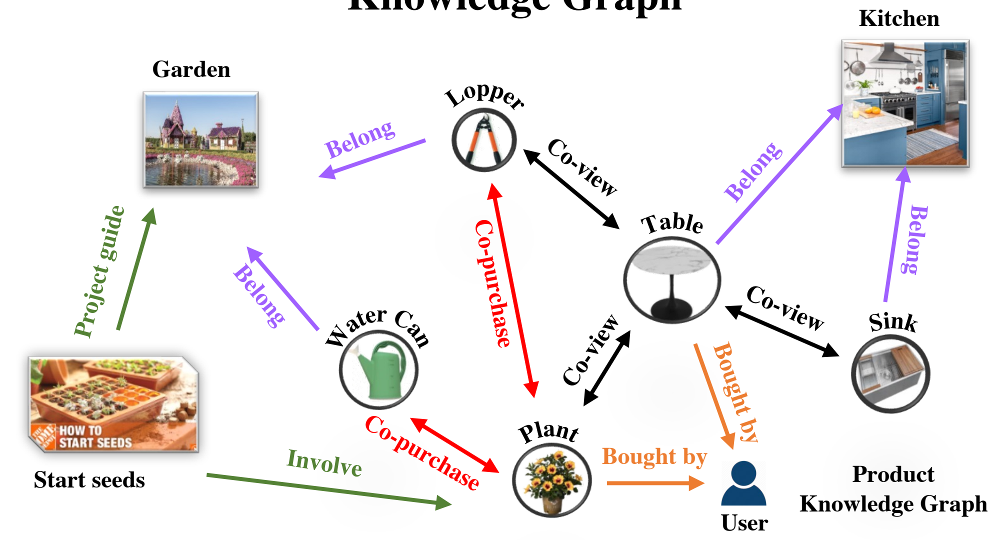
Infrastructure Graph



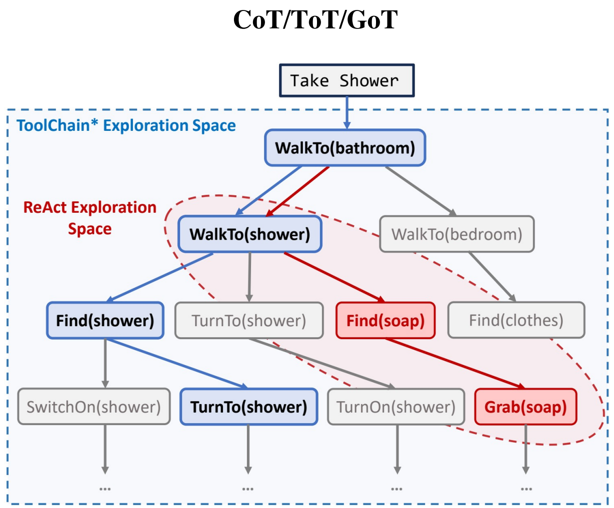
Social Interaction Graph



Knowledge Graph

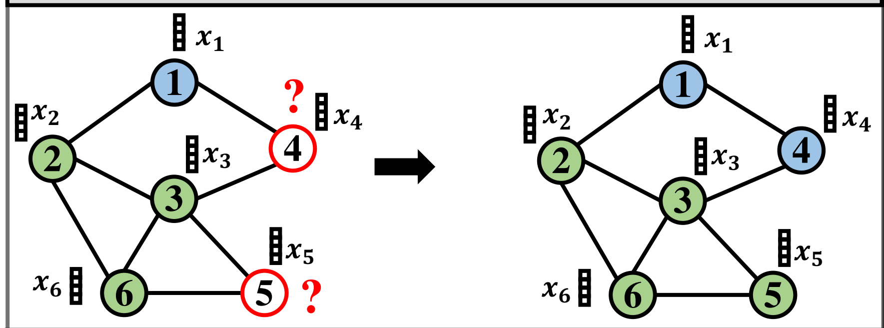


Reasoning / Planning Graph

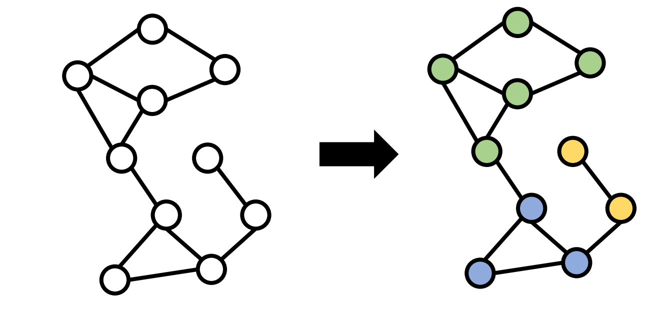


Introduction and Background - Graph-based Tasks and Graph Machine Learning

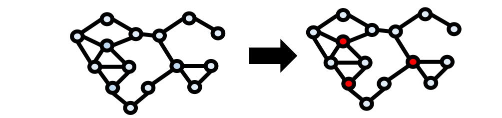
Node Classification/Regression



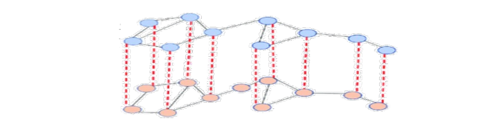
Community Detection



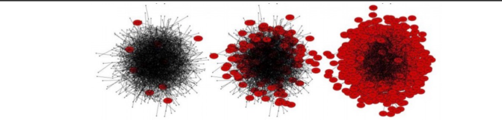
Network Dismantling



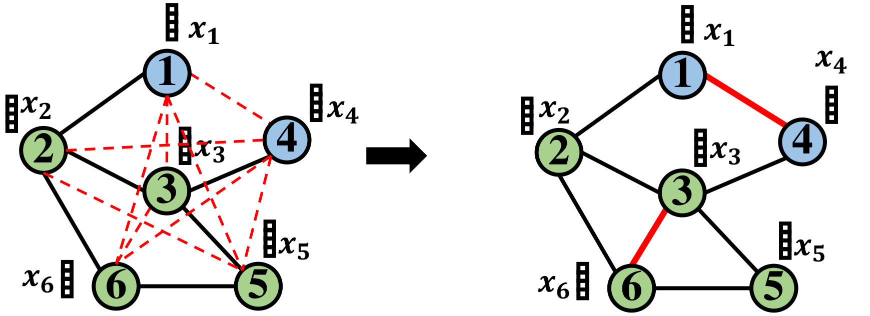
Network Alignment



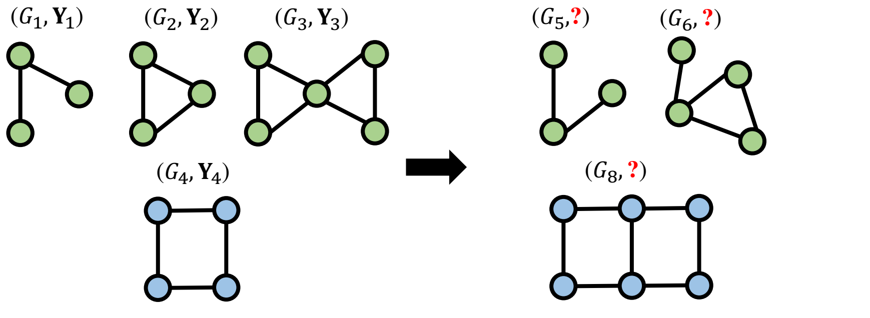
Influence Maximization

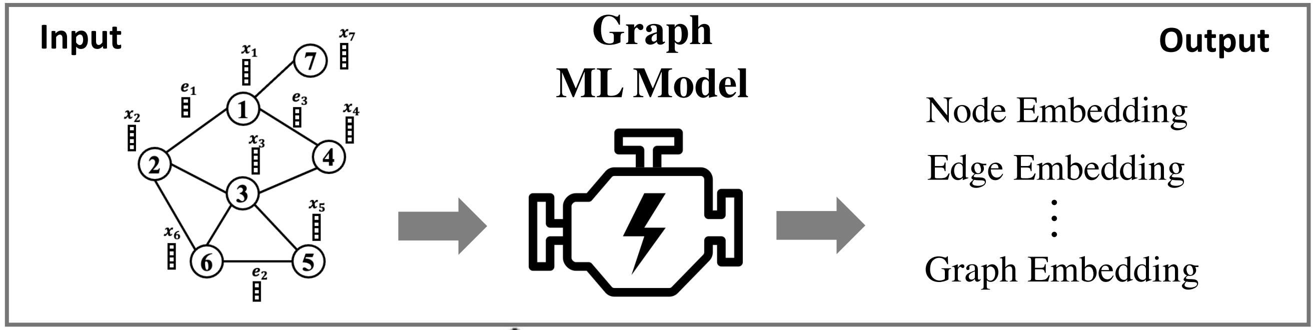


Link Prediction/Edge Classification



Graph Classification/Regression



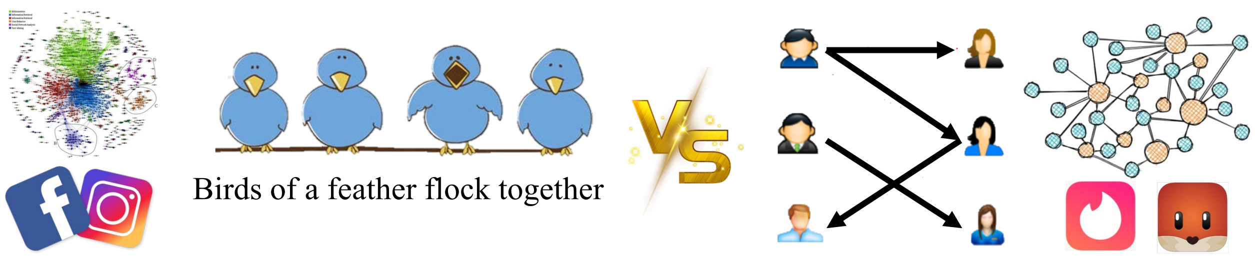


Real-world graph data can have data quality challenges…

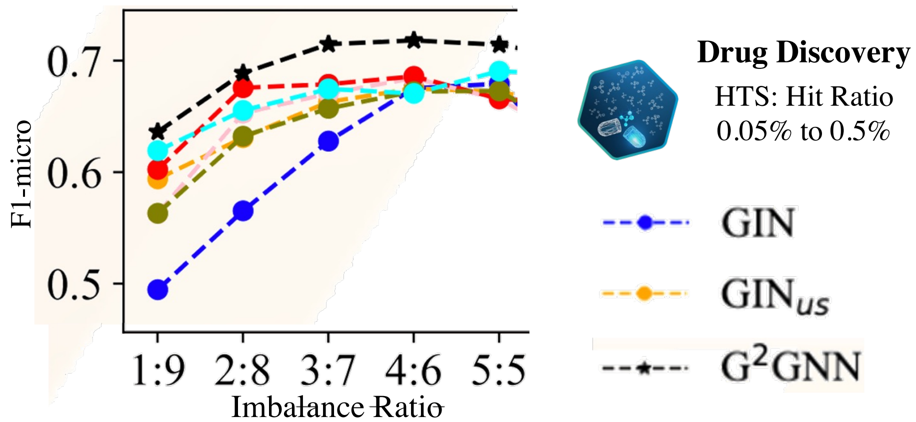
Garbage in, garbage out

Introduction and Background – Real-world Graphs have Data Quality Issues

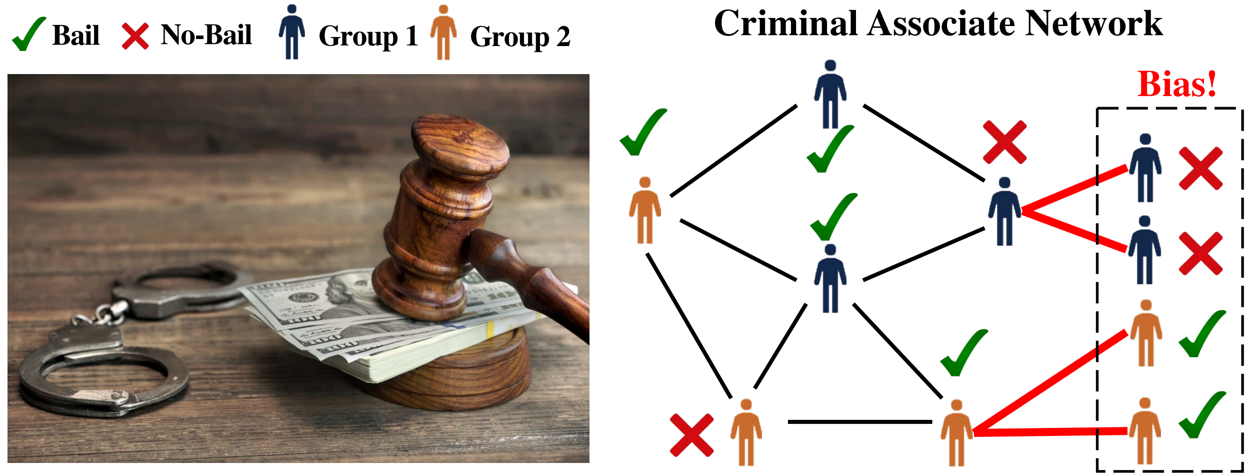
Topological Issues
e.g., Homophily vs Heterophily



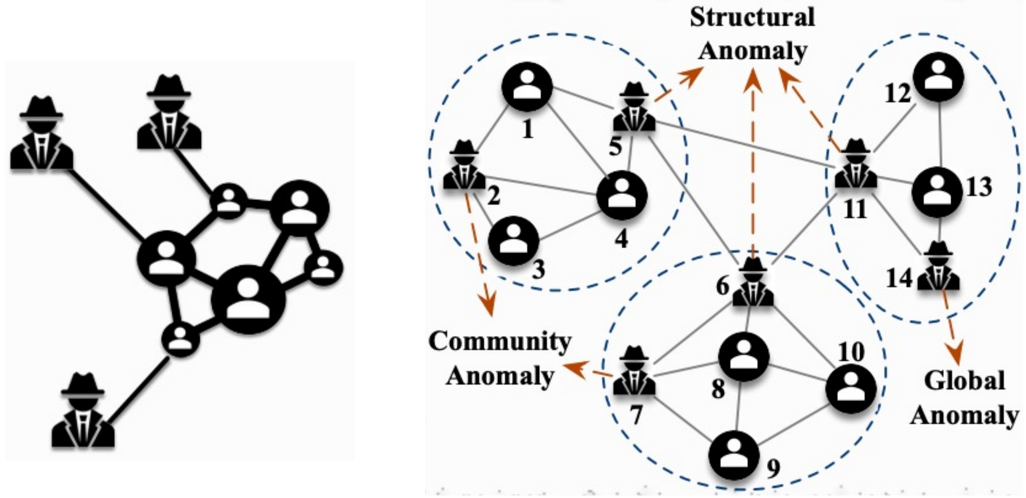
Imbalance Issues
e.g., labeled data in chemistry



Bias Issues
e.g., bail decision making



Abnormal Graph Data



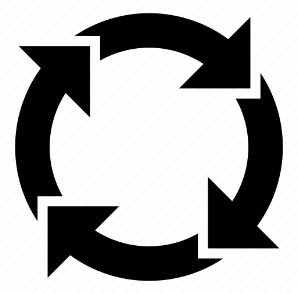
Introduction and Background – Model- vs. Data-Centric Methods

Find the *best model* for the given *fixed dataset*

Realize the *best dataset* for the given *prediction task*

Model-Centric

- Model architectures
- Loss functions/ constraints
- Hyperparameter tuning
- …
- etc.



Data-Centric

- Data Organization: Constructing graphs
- Data Integration: Improving node/edge features
- Data Cleaning: Confident learning
- …
- etc.

Introduction and Background – Model- vs. Data-Centric Methods

Credit: MIT Introduction to Data-Centric AI course & Inspired by XKCD 2494 “Flawed Data”

Data Quality-Aware Graph Machine Learning

- Introduction and Background
- Topology Issues
- Imbalance Issues
- Short Break
- Bias and Fairness Issues
- Limited Labeled Data Issues
- Abnormal Graph Data Issues
- Summary

Outline

- **Introduction and Background**
- Topology Issues
- Imbalance Issues
- Short Break
- Bias and Fairness Issues
- Limited Labeled Data Issues
- Abnormal Graph Data Issues
- Summary

Outline

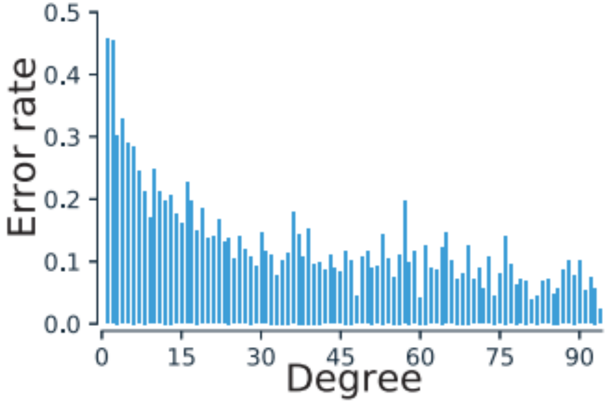
- Introduction and Background
- **Topology Issues**
- Imbalance Issues
- Short Break
- Bias and Fairness Issues
- Limited Labeled Data Issues
- Abnormal Graph Data Issues
- Summary

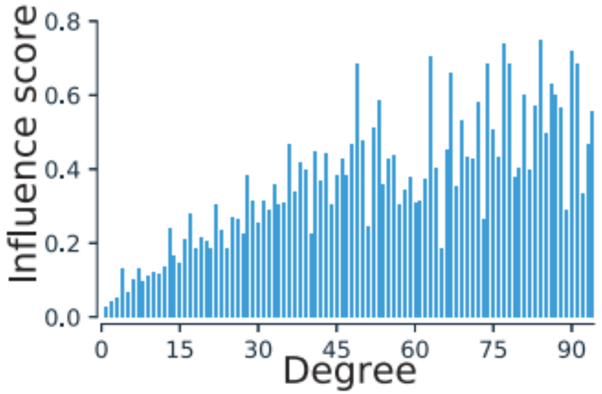
Topology Issues

- Global Positional Issues
- Local Topology Issues
- Missing Graph Issues
- Future Directions and Q&A

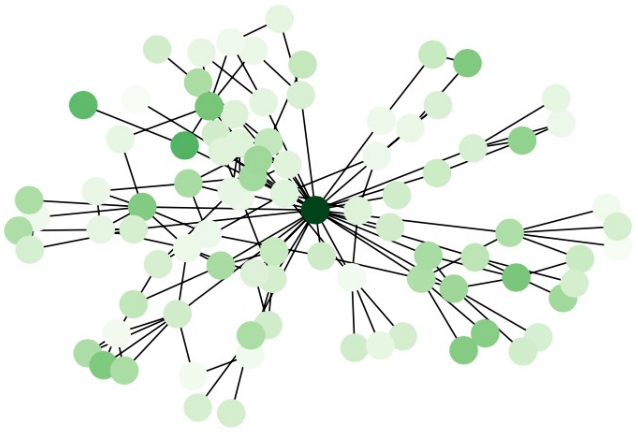
Topology Issues – Global Topology Issues – Labeled Node Influence

Degree -> Influence





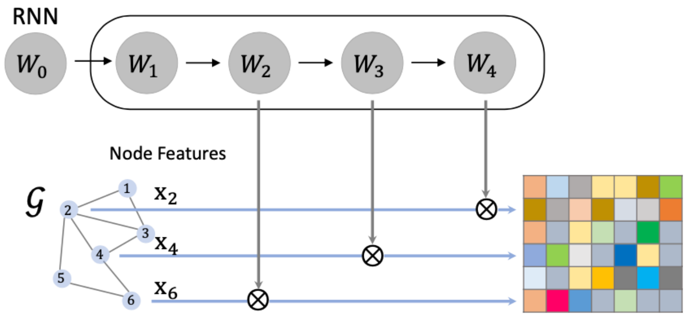
If $d_i > d_j$, v_i has higher influence than v_j on training GNNs



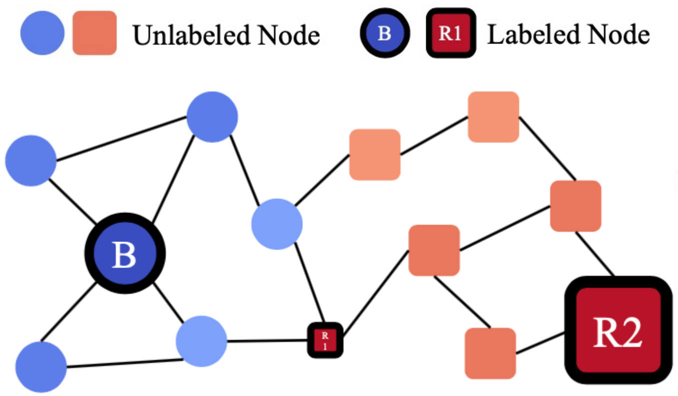
Darker colors - Higher influences.

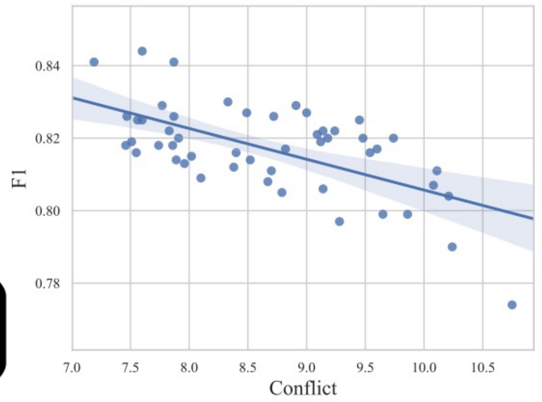
$$\mathbf{x}_i^{l+1} = \sigma\Big(\sum_{j \in \mathcal{N}_i} a_{ij} \left(\mathbf{W}^l + \boxed{\mathbf{W}^l_{d_j}}\right) \mathbf{x}_j^l\Big)$$

Degree-dependent!



Position -> Influence





$$\mathbf{P} = \alpha(\mathbf{I} - (1-\alpha)\mathbf{A}')^{-1}$$

$$\boxed{\mathbf{T}_v} = \mathbb{E}_{x \sim \mathbf{P}_{v:}}\Big(\sum_{j \in [1,k], j \neq y_v} |\mathcal{C}_j|^{-1} \sum_{i \in \mathcal{C}_j} \mathbf{P}_{i,x}\Big)$$

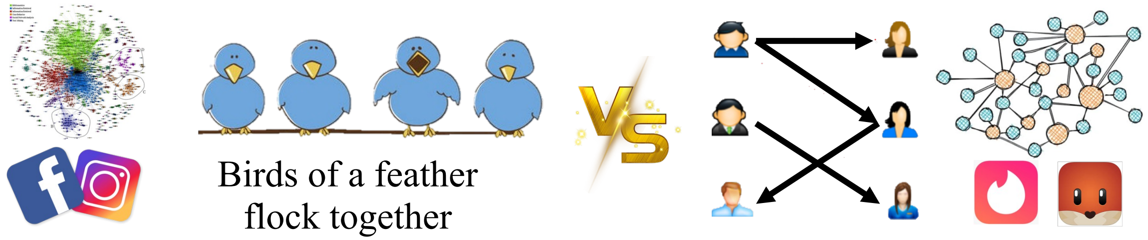
$$L = -|\mathcal{L}|^{-1} \sum_{v \in \mathcal{L}} \boxed{w_v} \sum_{c=1}^{k} y_v^c \log \mathrm{p}_v^c$$

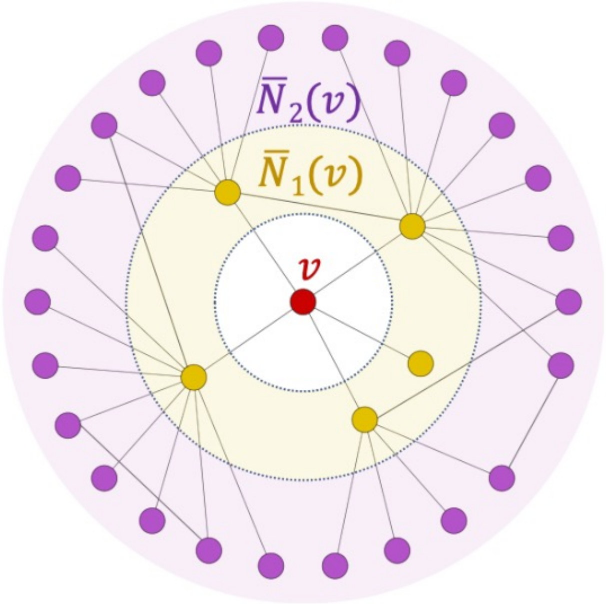
High T
Hight Conflicts,
low weight

Tang, Xianfeng, et al. "Investigating and mitigating degree-related biases in graph convoltuional networks." CIKM 2020
Chen, Deli, et al. "Topology-imbalance learning for semi-supervised node classification." NeurIPS 2021

Topology Issue – Local Topology Issues – Heterophily/Homophily

Homophily vs Heterophily





- Ego
- 1-order
- 2-order

Graph-level Homophily

$$h(\mathcal{G}, \{y_i; i \in \mathcal{V}\}) = \frac{1}{|\mathcal{E}|} \sum_{(j,k) \in \mathcal{E}} \mathbb{1}(y_j = y_k)$$

Ego-Neighbor Separation

$$\mathbf{r}_v^k = \mathrm{COMBINE}(\mathrm{r}_v^{k-1}, \mathrm{AGGR}(\{\mathbf{r}_u^{k-1}: u \in \mathcal{N}_v\}))$$

Higher-order Neighbor

$$\mathbf{r}_v^k = \mathrm{COMBINE}(\mathrm{r}_v^{k-1}, \mathrm{AGGR}_1(\{\mathbf{r}_u^{k-1}: u \in \mathcal{N}_v^1\}, \mathrm{AGGR}_2(\{\mathbf{r}_u^{k-1}: u \in \mathcal{N}_v^2\} \ldots))$$

Combination of Intermediate Representation

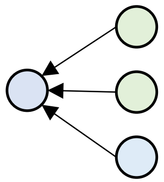
$$\mathbf{r}_v^k = \mathrm{COMBINE}(\mathrm{r}_v^1, \mathrm{r}_v^2, \ldots, \mathrm{r}_v^K)$$

Class belief propagation

$$\mathbf{B}^k = \mathbf{B}^0 + \underbrace{\mathbf{A}}_{\text{Transition among Graph}}\mathbf{B}^{k-1}\mathbf{H}$$

$\mathbf{H} \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$

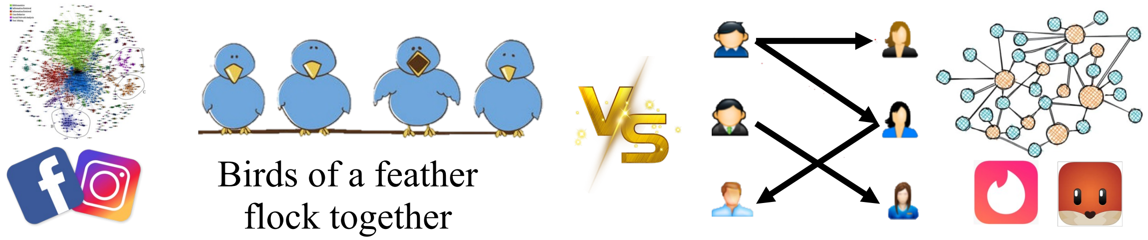
$\mathbf{B} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{Y}|}$

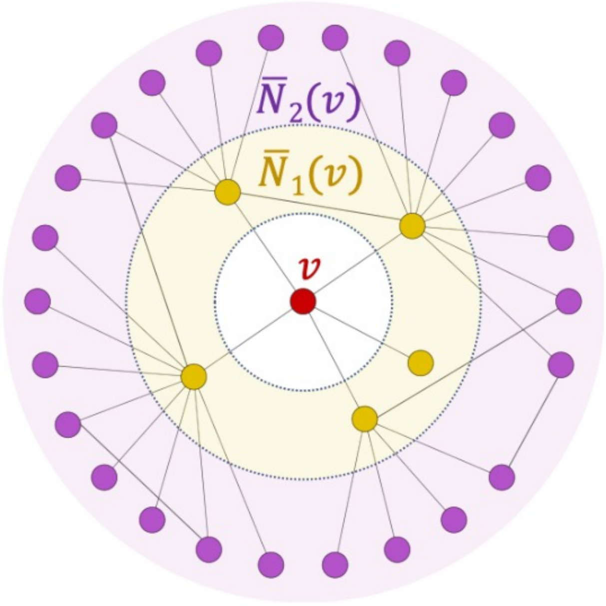


Zhu, Jiong, et al. "Beyond homophily in graph neural networks: Current limitations and effective designs." NeurIPS 2020
Zhu, Jiong, et al. "Graph Neural Networks with Heterophily." AAAI 2021

Topology Issue – Local Topology Issues – Heterophily/Homophily

Homophily vs Heterophily





Ego

1-order

2-order

Graph-level Homophily

$$h(\mathcal{G}, \{y_i; i \in \mathcal{V}\}) = \frac{1}{|\mathcal{E}|} \sum_{(j,k)\in\mathcal{E}} \mathbb{1}(y_j = y_k)$$

Ego-Neighbor Separation

$$\mathbf{r}_v^k = \mathrm{COMBINE}(\mathrm{r}_v^{k-1}, \mathrm{AGGR}(\{\mathbf{r}_u^{k-1}: u \in \mathcal{N}_v\}))$$

Higher-order Neighbor

$$\mathbf{r}_v^k = \mathrm{COMBINE}(\mathrm{r}_v^{k-1}, \mathrm{AGGR}_1(\{\mathbf{r}_u^{k-1}: u \in \mathcal{N}_v^1\}), \mathrm{AGGR}_2(\{\mathbf{r}_u^{k-1}: u \in \mathcal{N}_v^2\} \ldots))$$

Combination of Intermediate Representation

$$\mathbf{r}_v^k = \mathrm{COMBINE}(\mathrm{r}_v^1, \mathrm{r}_v^2, \ldots, \mathrm{r}_v^K)$$

Class belief propagation

$$\mathbf{B}^k = \mathbf{B}^0 + \underbrace{\mathbf{A}}_{\text{Graph Transition}}\mathbf{B}^{k-1}\underbrace{\mathbf{H}}_{\text{Class Transition}}$$

$\mathbf{H} \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$

$\mathbf{B} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{Y}|}$

$\mathbf{AB}^{k-1}$

[0.1|0.1|0.8]

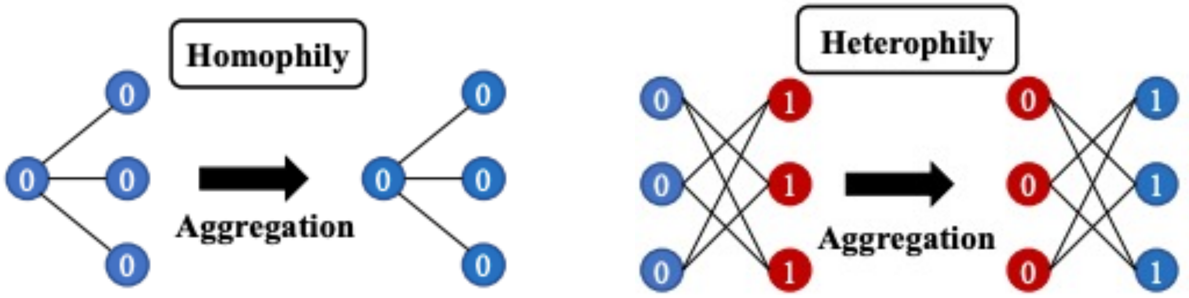
$$\begin{bmatrix} 0.1 & 0.3 & 0.6 \\ 0.8 & 0.1 & 0.1 \\ 0.7 & 0.3 & 0 \end{bmatrix}$$

H

Zhu, Jiong, et al. "Beyond homophily in graph neural networks: Current limitations and effective designs." NeurIPS 2020
Zhu, Jiong, et al. "Graph Neural Networks with Heterophily." AAAI 2021

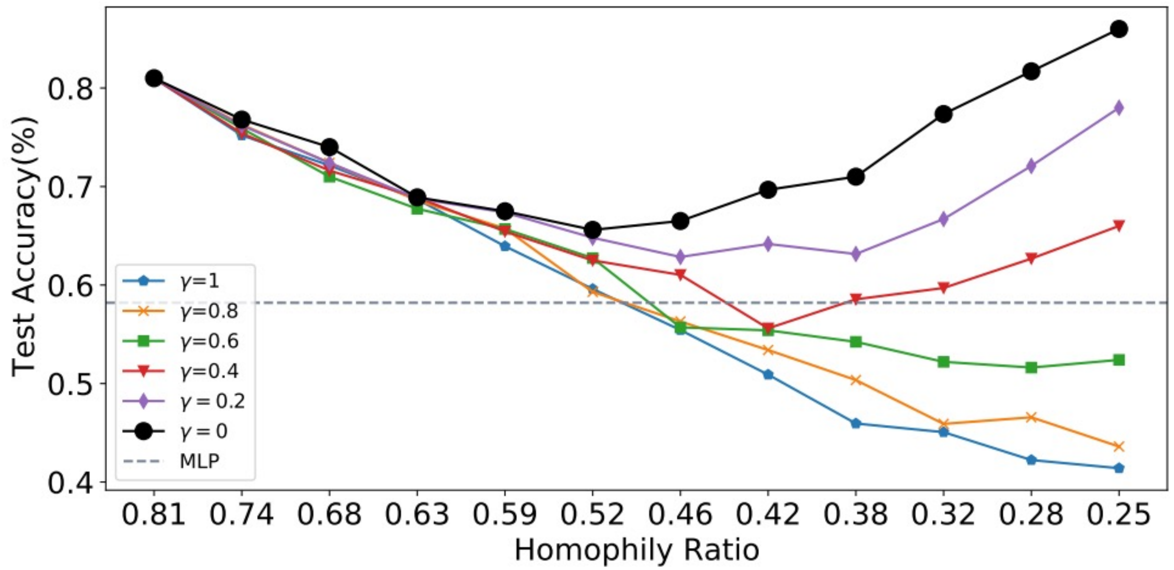
Topology Issue – Local Topology Issues – Heterophily/Homophily

Across Different Graphs

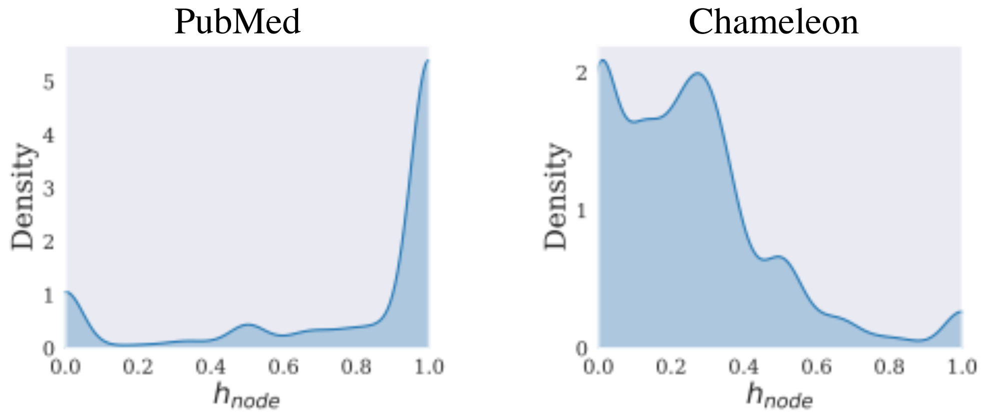


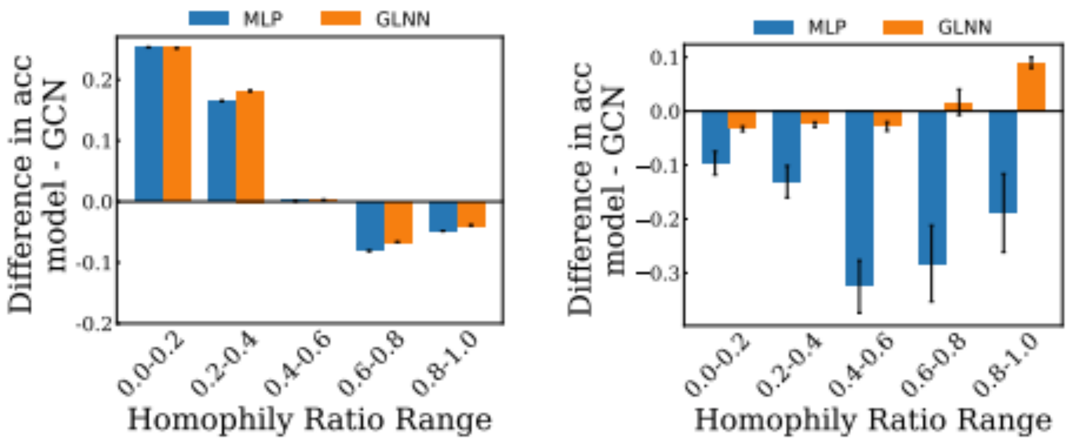
Class Neighborhood Distribution Distinguishability

High **Low** **High**



Within the Same Graph



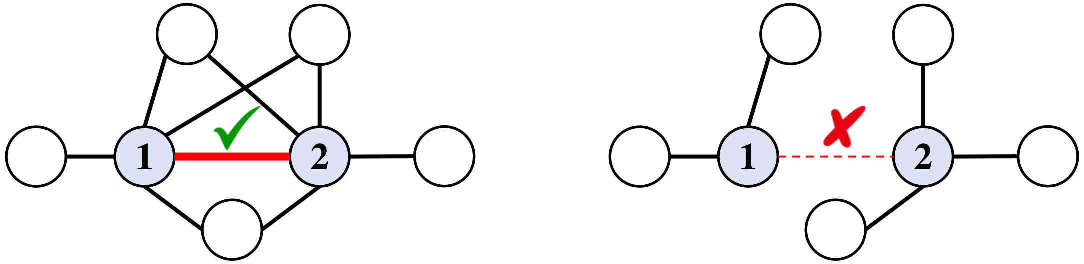


In **homophily** graph, GNNs > MLP on **homophily** nodes

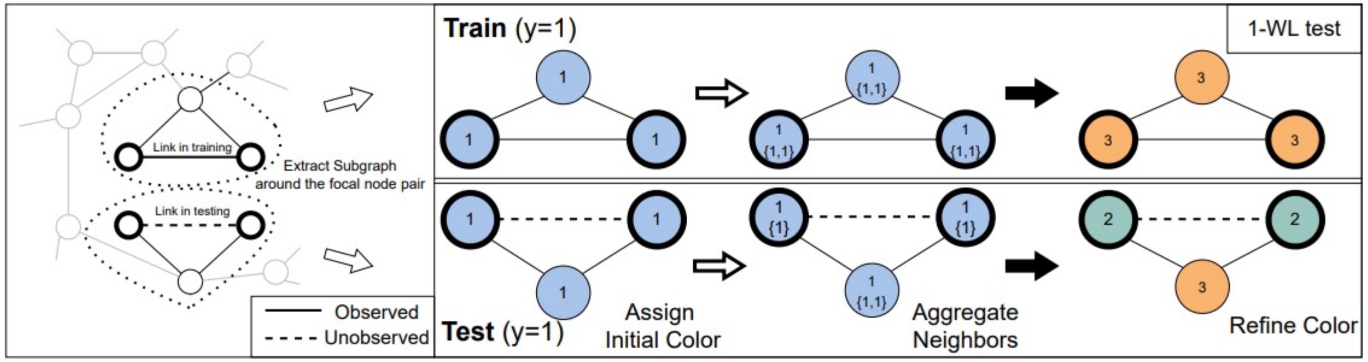
In **heterophily** graph, GNNs > MLP on **heterophily** nodes

Topology Issue – Local Topology Issues – Training-to-Testing Topology Shift

Link Prediction

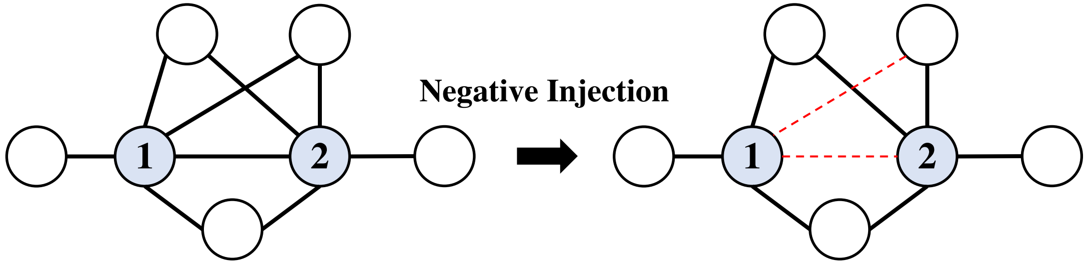


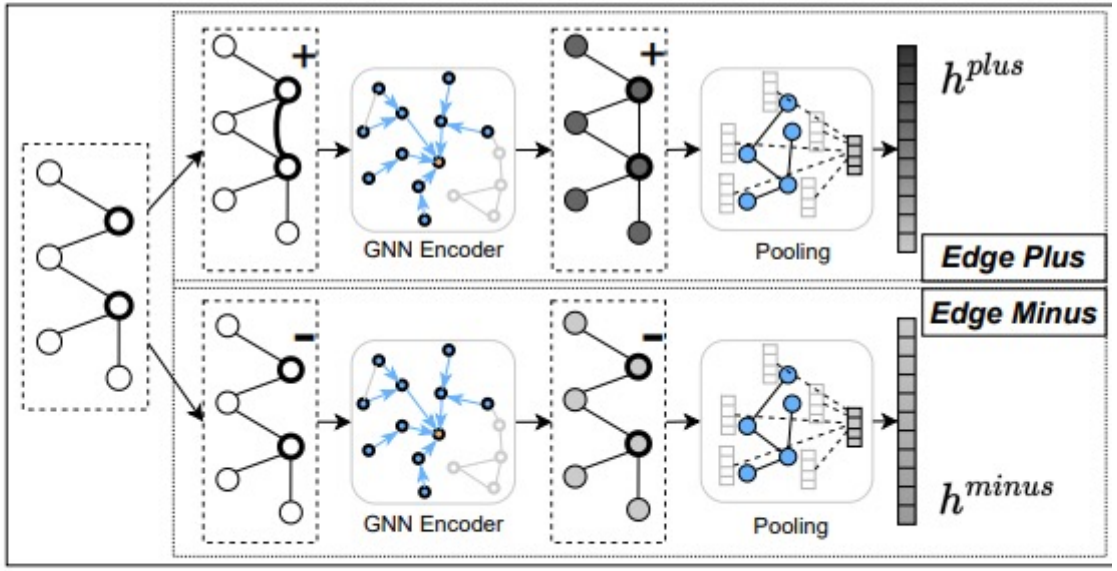
Local Subgraph → **Predictor** → Link



Focal Link is missing from training subgraph to testing subgraph

Distribution Shift





Edge Mean

$$\mathbf{h}^{\text{mean}} = \frac{(\mathbf{h}^{+} + \mathbf{h}^{-})}{2}$$

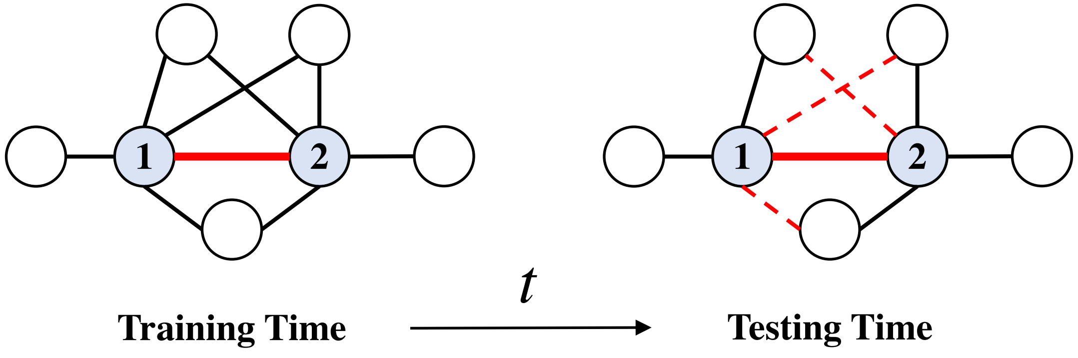
Edge Attention

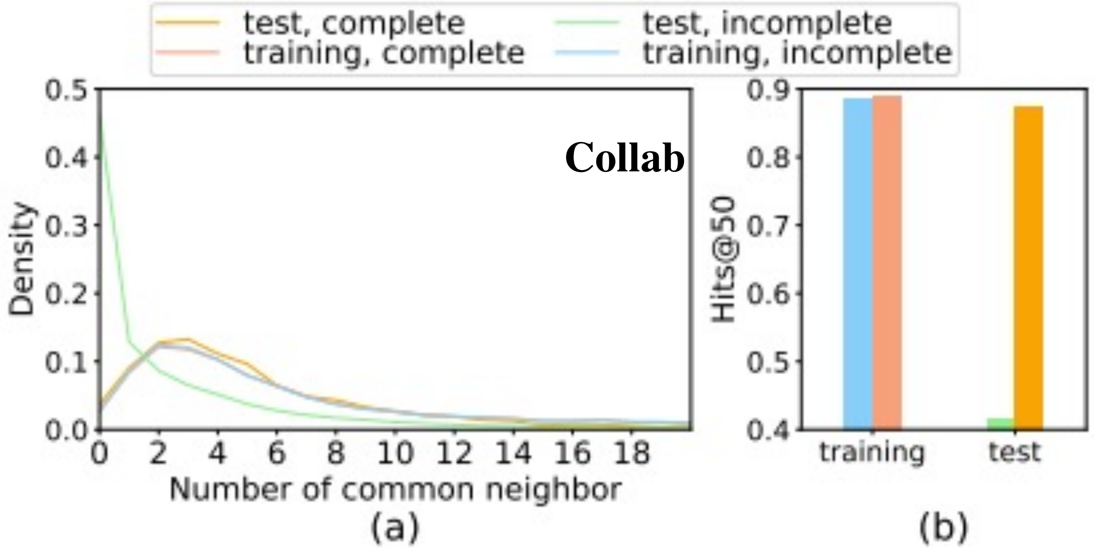
$$\mathbf{h}^{\text{mean}} = w^{+}\mathbf{h}^{+} + w^{-}\mathbf{h}^{-}$$
$$w^{+} = \sigma(\mathbf{q}^{\mathrm{T}}\tanh(\mathbf{W}\mathbf{h}^{+} + \mathbf{b}))$$

Zhang, Muhan et al. "Link prediction based on graph neural networks." NeurIPS, 2018
Dong, Kaiwen, et al. "Fakeedge: Alleviate dataset shift in link prediction." LOG, 2022

Topology Issue – Local Topology Issues – # of Common Neighbor Shift

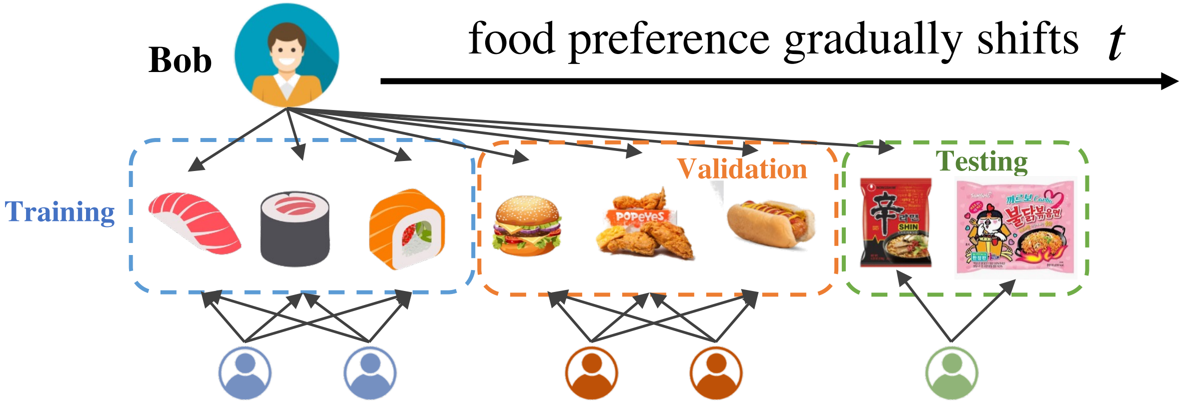
Link-centric Perspective





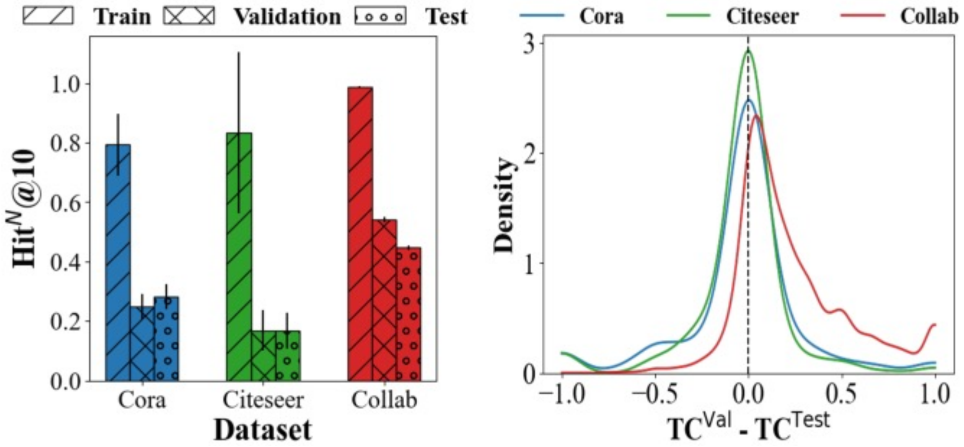
Time-based Split
Testing edges have more testing edges around

Node-centric Perspective



$\mathbf{TC}^{\mathrm{Val}}$: Common interaction between training and validation

$\mathbf{TC}^{\mathrm{Test}}$: Common interaction between training and validation



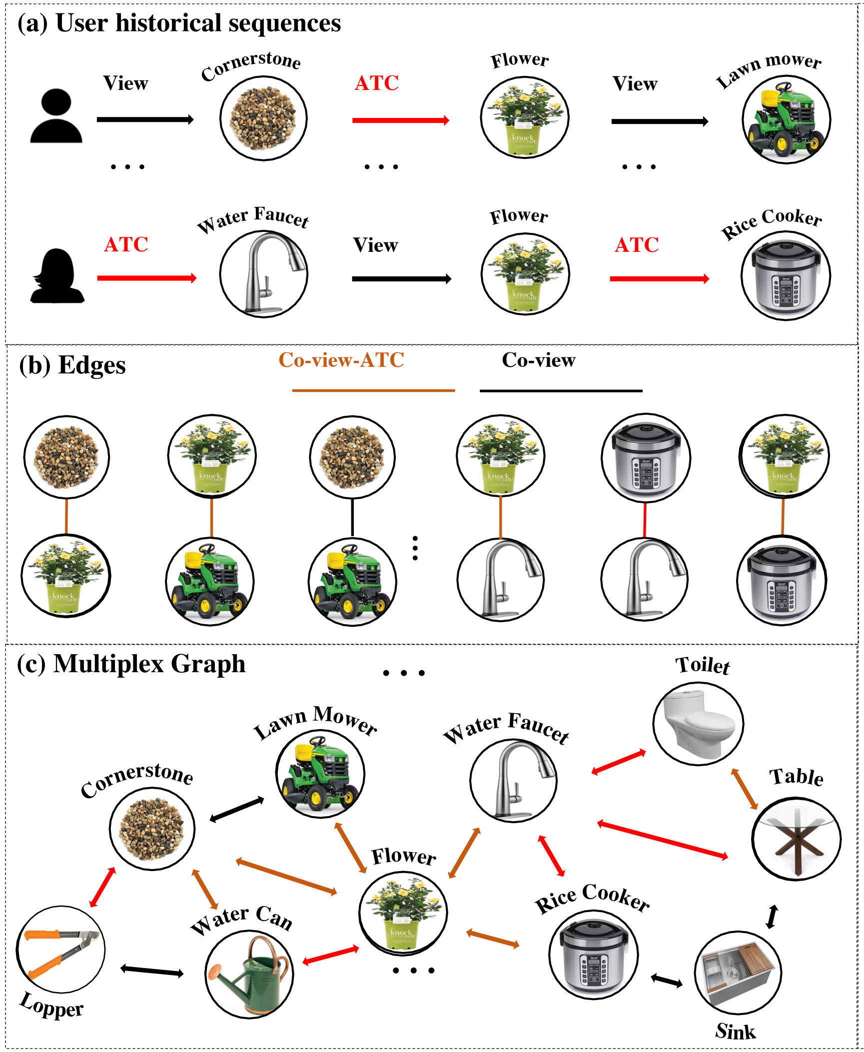
Wang, Xiyuan et al. "Neural Common Neighbor with Completion for Link Prediction." ICLR, 2024
Wang, Yu, et al. "A Topological Perspective on Demystifying GNN-based Link Prediction Performance" ICLR, 2024

Topology Issue – Missing Topology Issues

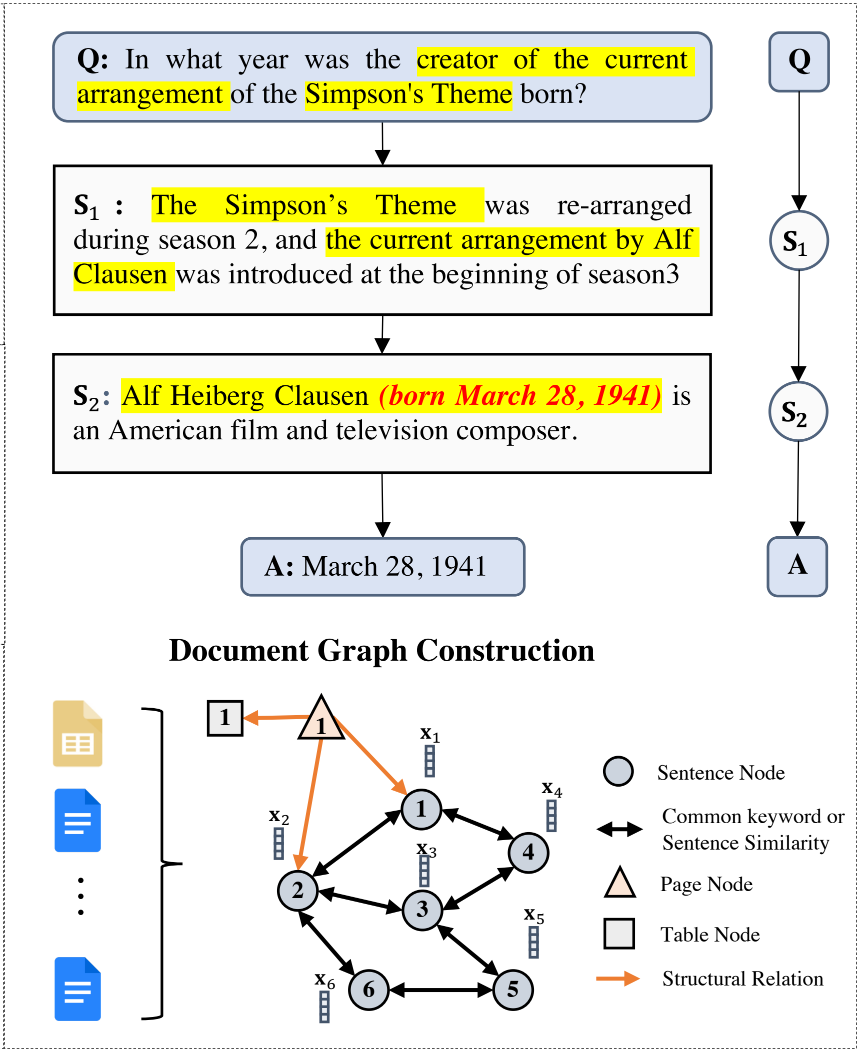
Sometimes Real-world Applications do not have Graphs!

But Graph can actually encode some useful information

User/Item Interaction

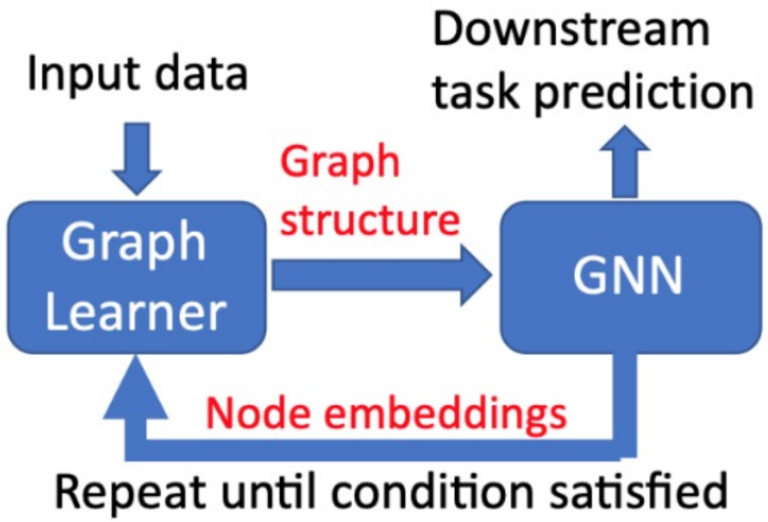


Multi-hop Reasoning



Wang, Yu, et al. "Knowledge graph prompting for multi-document question answering." AAAI, 2024
Wang, Yu, et al. "Knowledge Graph-based Session Recommendation with Adaptive Propagation." WWW, 2024

Topology Issue – Missing Topology Issues



Better Graph Structure

Better Node Embeddings

GNN embeddings

$$\mathbf{A}_{ij}^{p} = \cos\left(\mathbf{w}_p \odot \boxed{\mathbf{v}_i,} \mathbf{w}_p \odot \mathbf{v}_j\right), \mathbf{A}_{ij} = m^{-1}\sum_{p=1}^{m} a_{ij}^{p}$$

Real-world Graph is sparse!

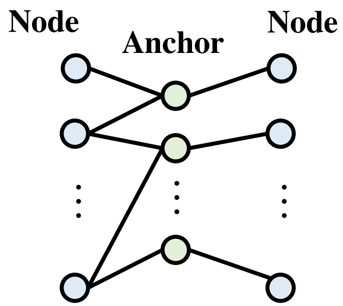
$$\mathbf{a}_{ij} = \begin{cases} \mathbf{A}_{ij}, \mathbf{a}_{ij} < \epsilon \\ 0, \ \mathbf{a}_{ij} > \epsilon \end{cases} \qquad \mathbf{A}^{t} = \lambda \mathbf{L}^{0} + (1-\lambda)(\eta f(\mathbf{A}^{t}) + (1-\eta) f(\mathbf{A}^{1}))$$

$$\mathbf{L}^{0} = (\mathbf{D}^{0})^{-0.5}\mathbf{A}^{0}(\mathbf{D}^{0})^{-0.5}$$

Quadratic Computation $\mathcal{O}(n^2)$ $\quad f(\mathbf{A})_{ij} = \mathbf{A}_{ij} / \sum_{\mathbf{j}} \mathbf{A}_{ij}$

$$\mathbf{r}_{ik}^{p} = \cos\left(\mathbf{w}_p \odot \mathbf{v}_i, \mathbf{w}_p \odot \mathbf{u}_k\right), \mathbf{r}_{ik} = m^{-1}\sum_{p=1}^{m} \mathbf{r}_{ik}^{p}$$

Anchor Selection $\mathcal{O}(\mathrm{nK}), \mathrm{K} \ll \mathrm{n}$

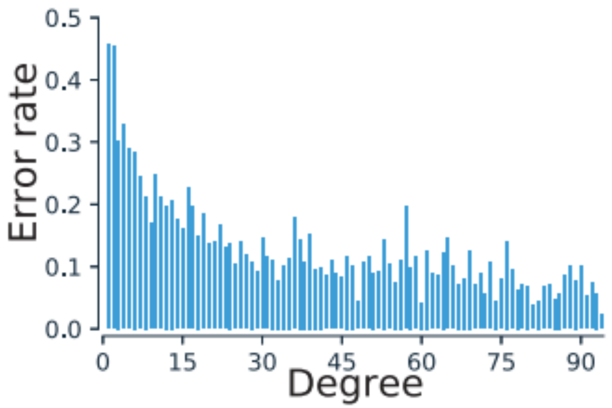


$$\mathbf{F}^{0,\prime} = \mathbf{\Lambda}^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{F}^{0}$$ Node -> Anchor

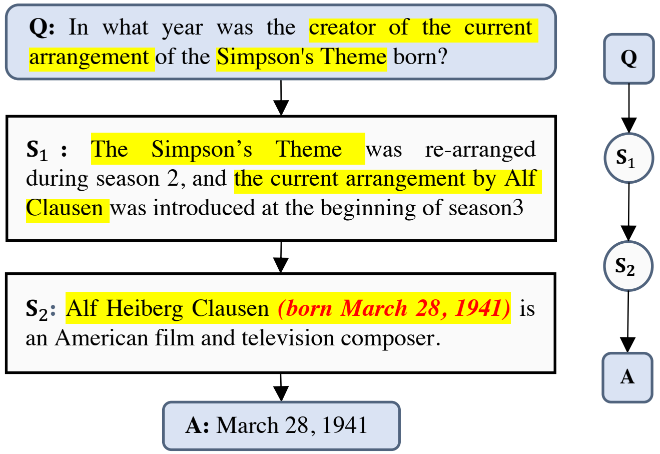
$$\mathbf{F}^{1} = \mathbf{\Lambda}^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{F}^{0,\prime}$$ Anchor -> Node

Q&A and Future Work – Topology Issue

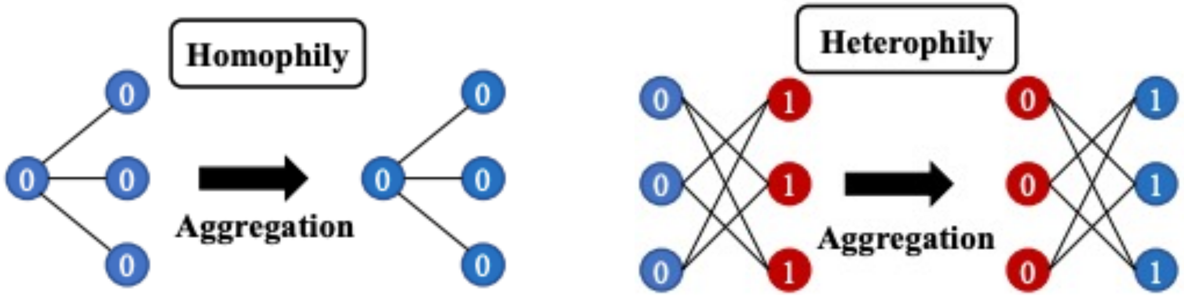
Global Topology Issue



Missing Topology Issue



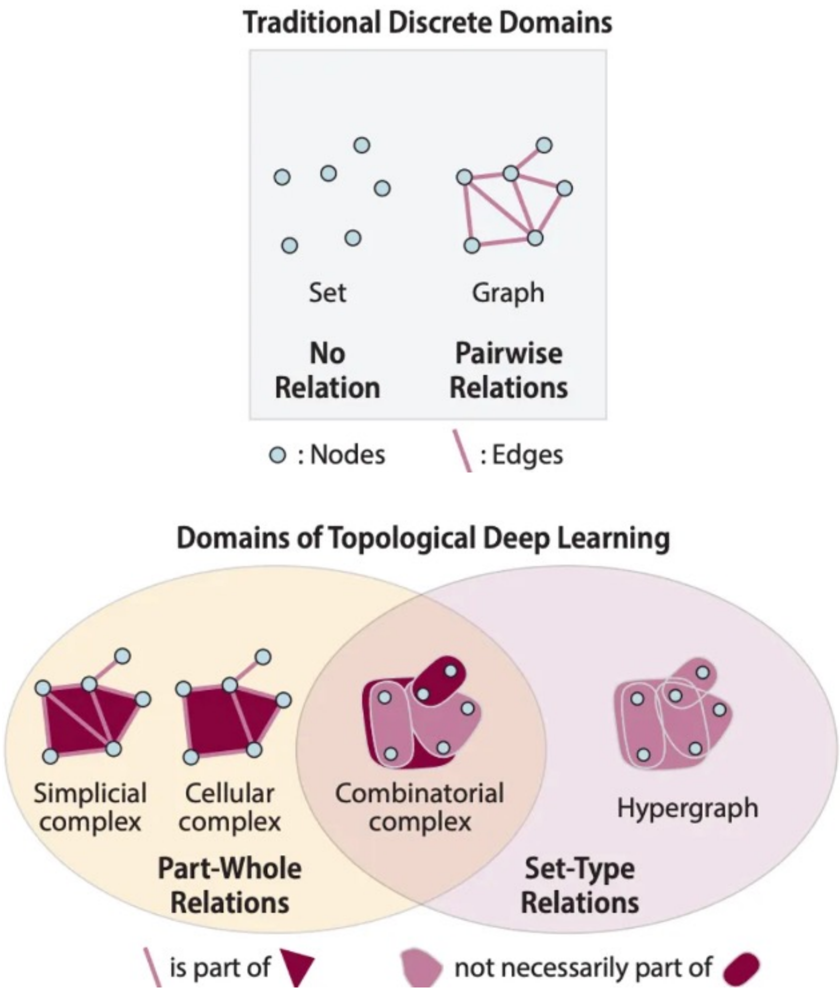
Local Topology Issue



Class Neighborhood Distribution Distinguishability

High **Low** **High**

Topology Issue of Complex Graphs



Outline

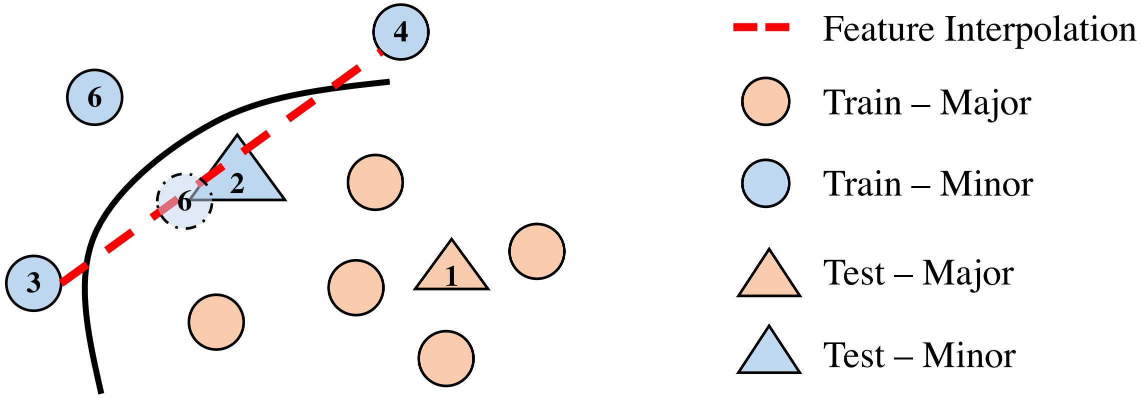
- Introduction and Background
- Topology Issues
- **Imbalance Issues**
- Short Break
- Bias and Fairness Issues
- Limited Labeled Data Issues
- Abnormal Graph Data Issues
- Summary

Imbalance Issues

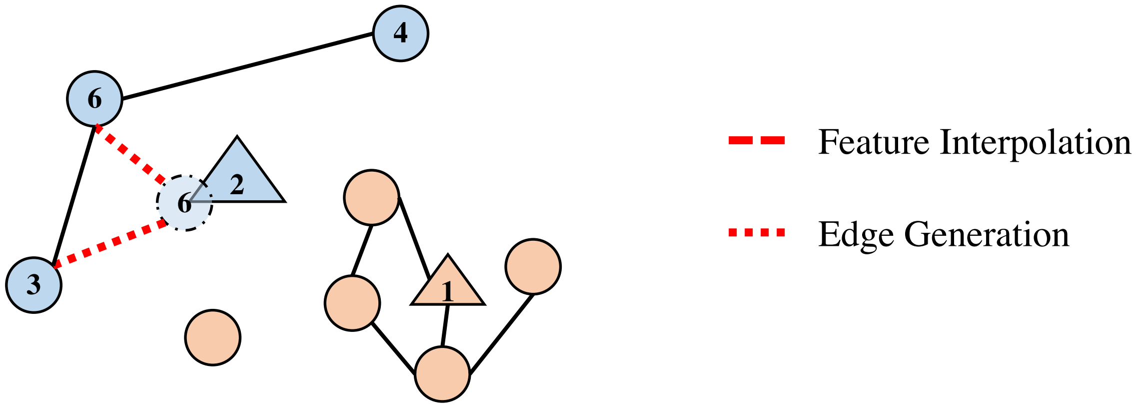
- Node-level Imbalance
- Graph-level Imbalance
- Edge-level Imbalance
- Future Directions and Q&A

Imbalance Issues – Node-level imbalance

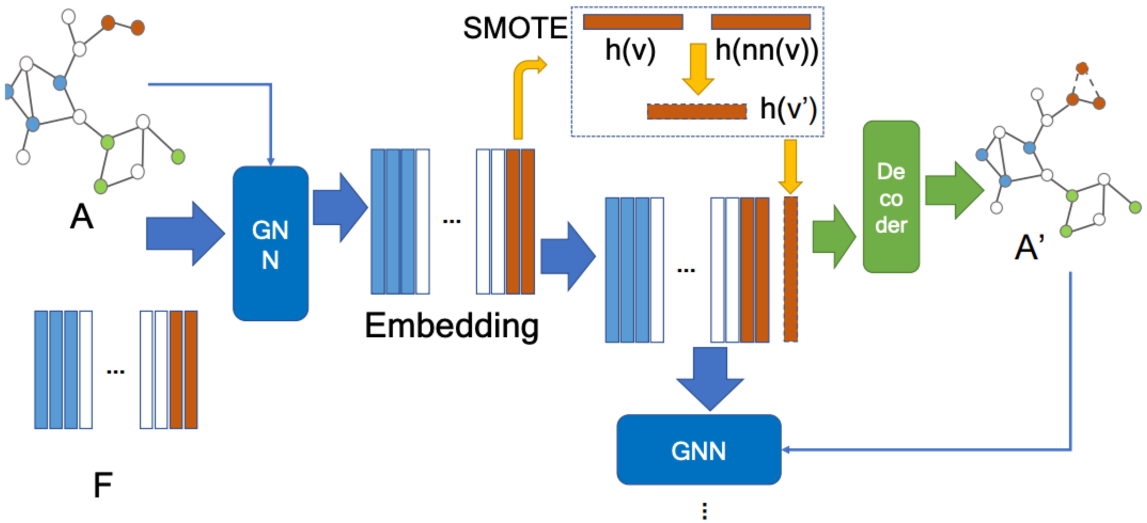
SMOTE



Graph-structured data has both feature and edge



GraphSMOTE



$$nn(v) = \text{argmin}_u ||\mathbf{h}_u^1 - \mathbf{h}_v^1||, \text{s.t.}\, \mathbf{Y}_u = \mathbf{Y}_v$$

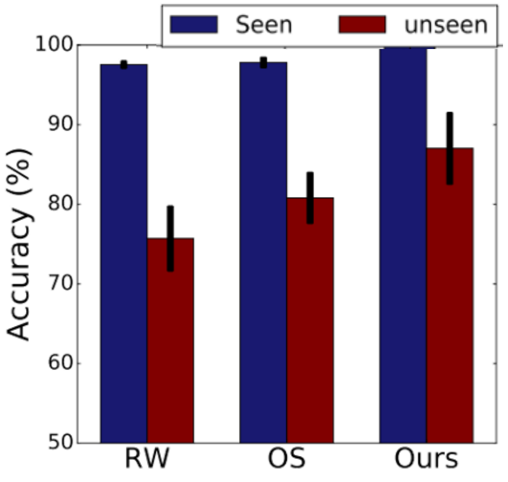
$$\mathbf{h}_{v'}^1 = (1-\delta)\mathbf{h}_v^1 + \delta \mathbf{h}_{nn(v)}^1$$

$$\mathbf{A}_{v'u} = \begin{cases} 1, \text{if } \mathbf{E}_{v'u} \geq \eta \\ 0, \text{otherwise} \end{cases} \qquad \mathcal{L}_{edge} = ||\mathbf{E} - \mathbf{A}||_F^2$$

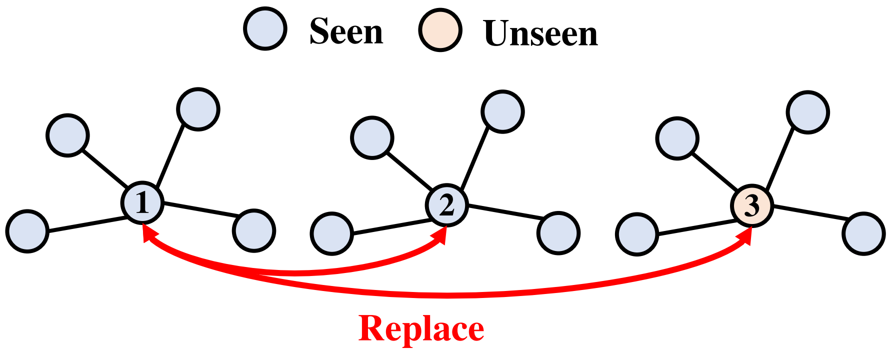
$$\mathbf{E}_{vu} = softmax(\sigma(\mathbf{h}_v^1 \mathbf{S} \mathbf{h}_u^1))$$

Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." JAIR, 2002
Zhao, Tianxiang., et al. "Graphsmote: Imbalanced node classification on graphs with graph neural networks." WSDM, 2021

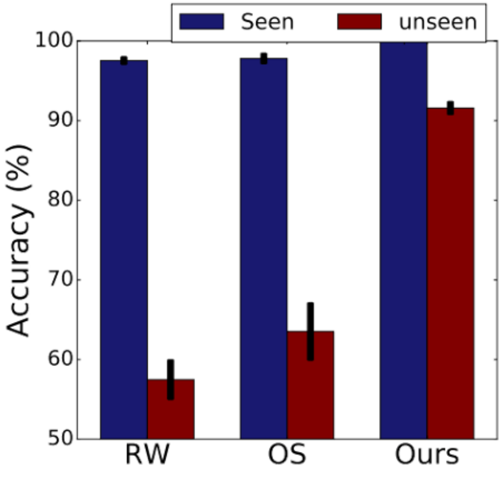
Imbalance Issues – Node-level imbalance



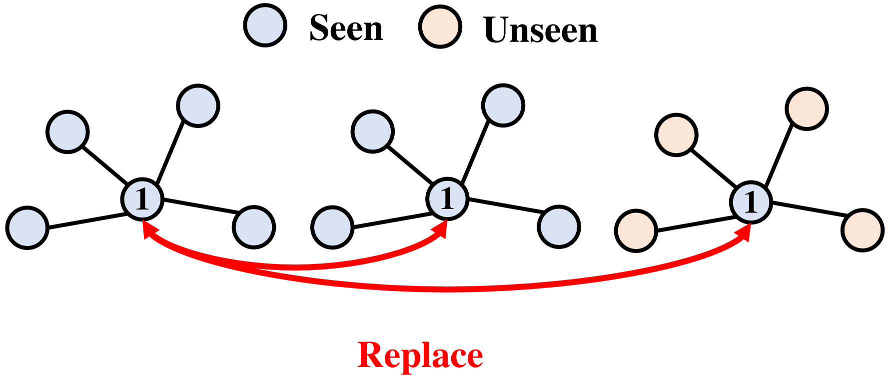
Node-replacing



Neighbor stays the same

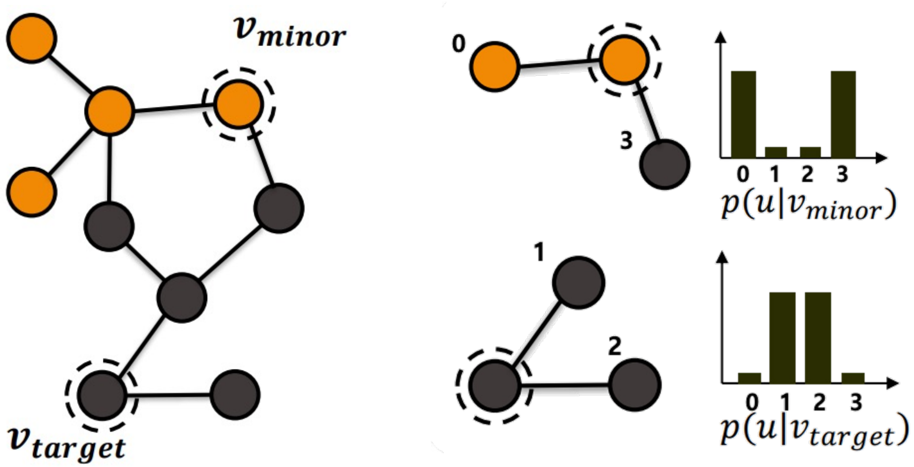


Neighbor-replacing



Center nodes stays the same

Neighborhood Memorization



$$p(u|v_{mixed}) = \hat{\phi}p(u|v_{minor}) + (1-\hat{\phi})p(u|v_{target})$$

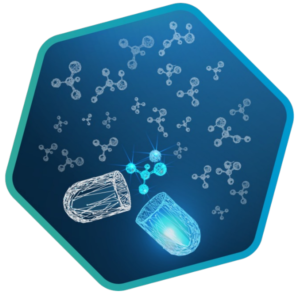
$$0.5 < \hat{\phi} = \frac{1}{1+e^{-\phi}} < 1 \quad \phi = KL(\sigma(\mathbf{o}_{minor})||\sigma(\mathbf{o}_{target}))$$

$$\mathbf{o}_{minor} = |\mathcal{N}_v|^{-1} \sum_{u \in \mathcal{N}_v} \mathbf{o}_{minor}$$

Park, Joonhyung et al. "GraphENS: Neighbor-aware ego network synthesis for class-imbalanced node classification." ICLR 2021
Ma, Yihong, et al. "Class-imbalanced learning on graphs: A survey." arXiv 2023

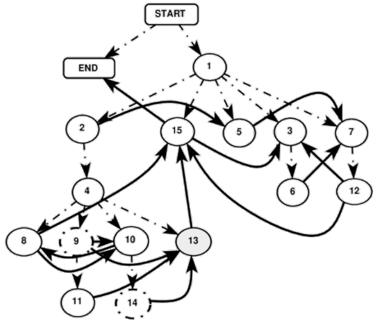
Imbalance Issues – Graph-level imbalance

Drug Discovery



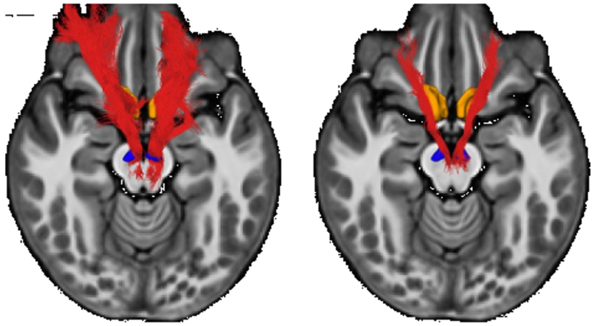
HTS Hit Ratio
0.05% to 0.5%

Malware Detection



0.01% Google, 2% Android,

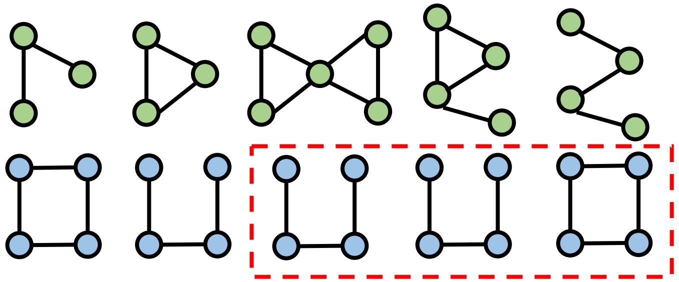
ASD Brain Classification



Normal : Autism
36 : 1

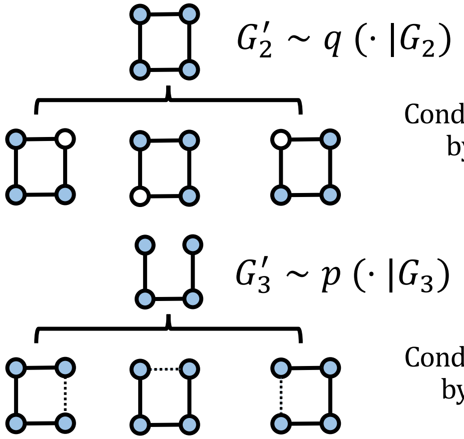
Autism Statistics. 2023

Quantity Augmentation



Structure Augmentation

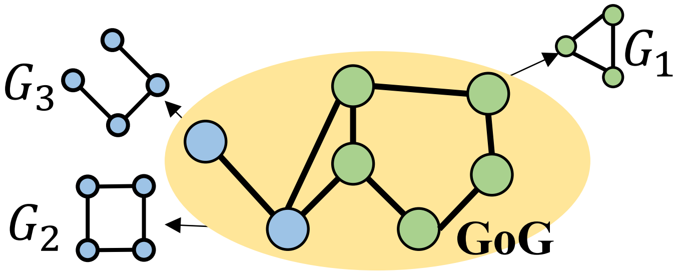
SPP - Structurally Similar Molecules tend to have similar properties

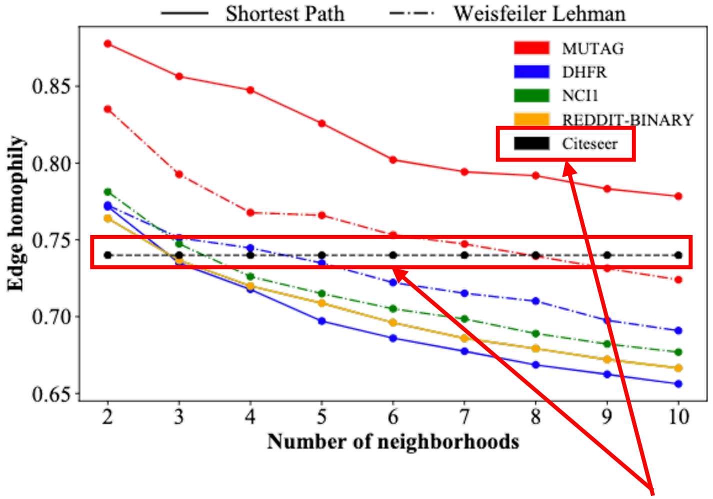


Conditional distribution by node masking

Conditional distribution by edge removing

Graph-of-Graphs (GoG)

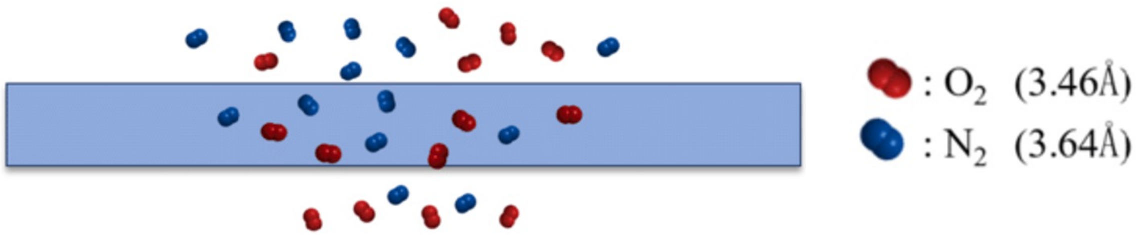




Constructed GoG demonstrates high homophily!

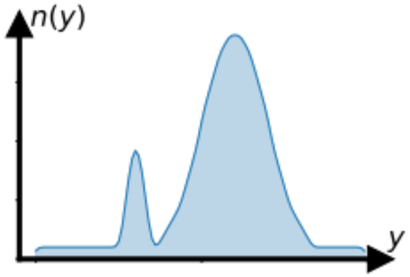
Wang, Yu, et al. "Imbalanced graph classification via graph-of-graph neural networks." CIKM 2022
Liu, Yunchao Lance, et al. "Interpretable chirality-aware GNN for structure activity relationship modeling in drug discovery." AAAI 2023

Imbalance Issues – Graph-level imbalance

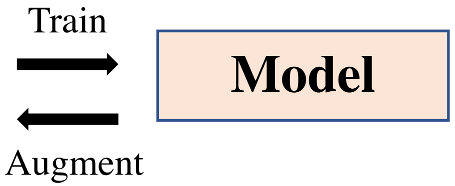


70 years, ~600 polymers, oxygen permeability ,
Polymer Gas Separation Membrane Database

Imbalance Graph Regression!



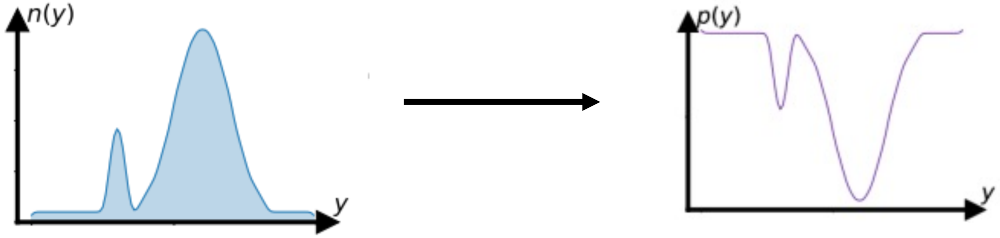
Self-training



(1) Use Model to predict on unlabeled graphs and select those high-quality-one

$$\sigma_i = \frac{1}{\mathrm{Var}\left(\{f(g(G_{(i,j)}))\}_{j=1,2,\ldots,B}\right)}.$$

(2) Sample more for label interval with less training samples



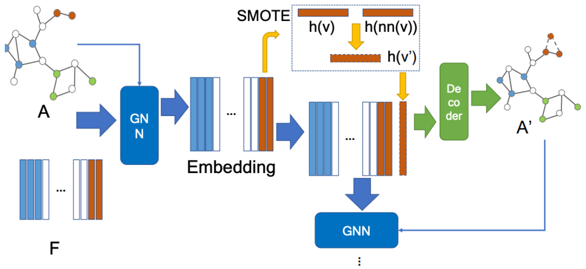
(3) Anchor-based Mix-up

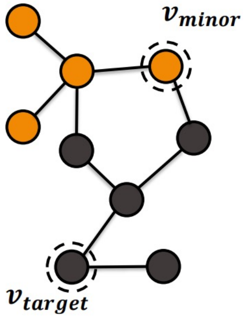
$a_i, \mathbf{z}_i$: anchor-label and embedding

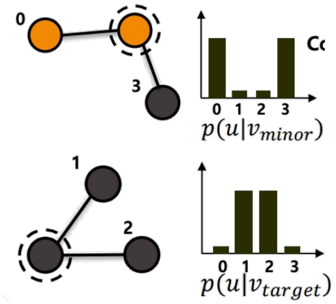
$$\begin{cases} \tilde{\mathbf{h}}_{(i,j)} & = \lambda \cdot \mathbf{z}_i + (1-\lambda) \cdot \mathbf{h}_j, \\ \tilde{y}_{(i,j)} & = \lambda \cdot a_i + (1-\lambda) \cdot y_j, \end{cases}$$

Q&A and Future Work – Imbalance Issues

Node-level Imbalance

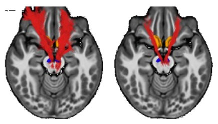






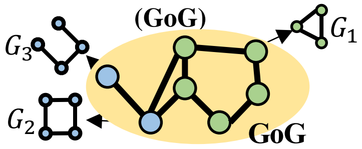
Graph-level Imbalance

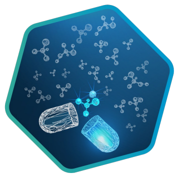
ASD Brain

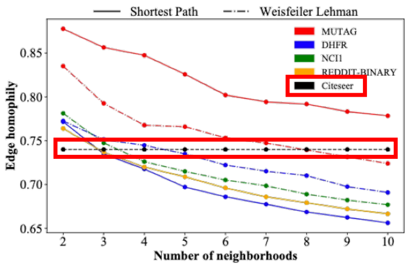


Normal Autism
36 : 1

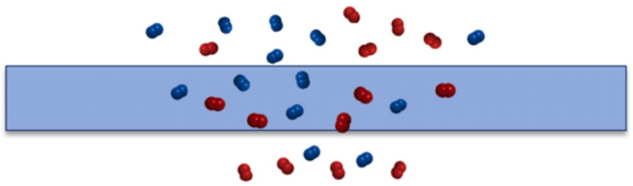
Graph-of-Graphs (GoG)

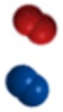






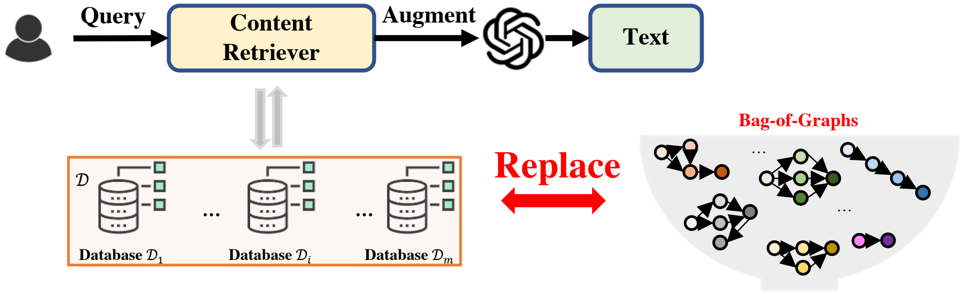
HTS Hit Ratio
0.05% to 0.5%



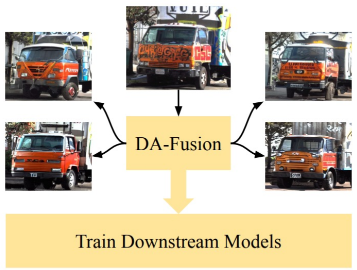


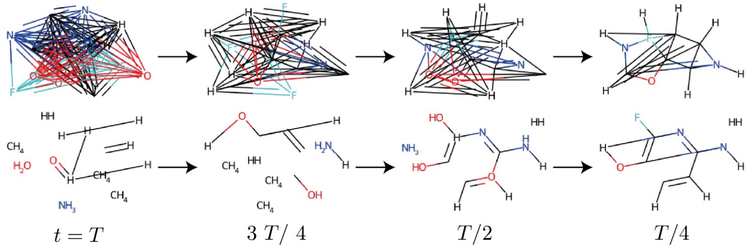
: O_2 (3.46Å)
: N_2 (3.64Å)

Retrieval Additional Supervision



Generate Additional Supervision





Outline

Short Break (4 min)

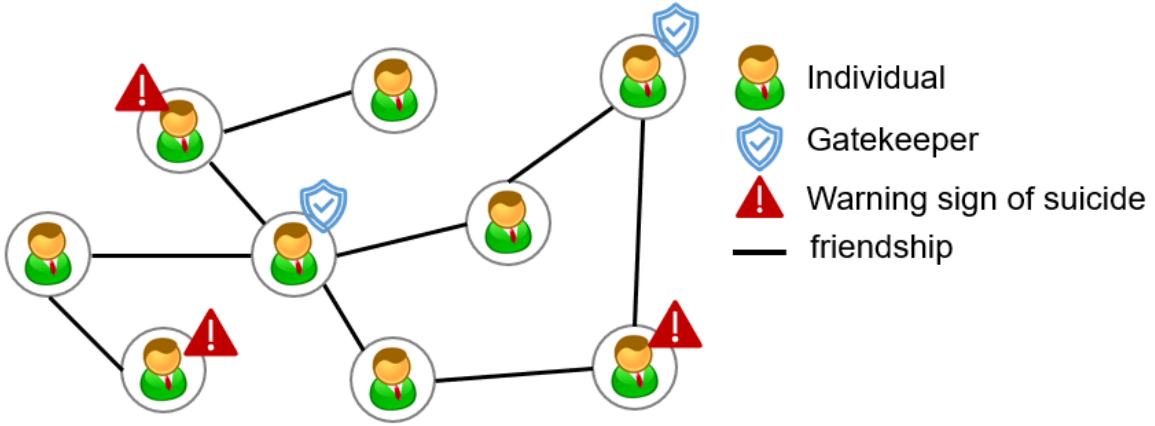
Outline

- Introduction and Background
- Topology Issues
- Imbalance Issues
- Short Break
- **Bias and Fairness Issues**
- Limited Labeled Data Issues
- Abnormal Graph Data Issues
- Summary

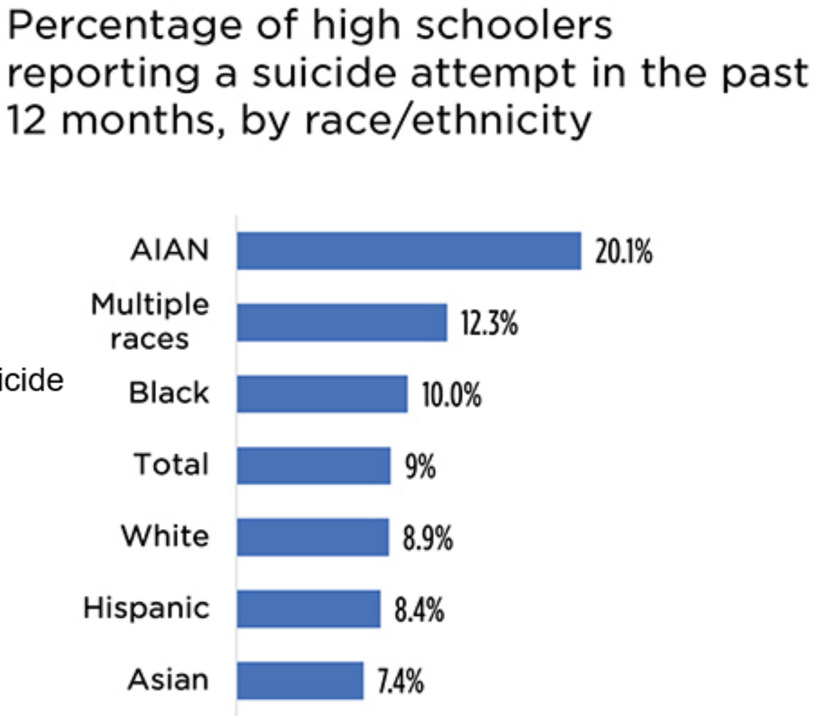
Bias and Fairness Issues - Suicide Prevention

- **Why suicide prevention?**
  - Suicide is one of the leading causes of death in United States

Gatekeeper training programs



Toy example of a gatekeeper training program

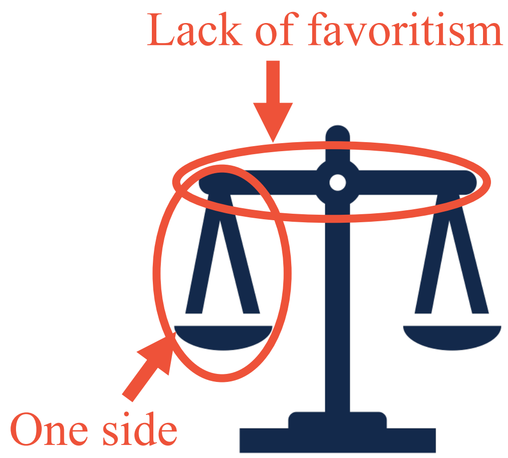


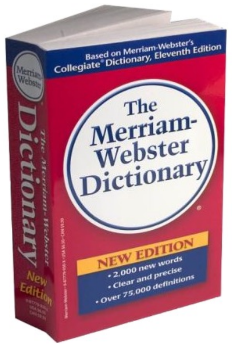
Suicide attempts by race/ethnicity

  - Existing prevention strategies disproportionately affect different groups
- **Key question**
  - How to correct the bias and ensure fairness on graphs?

Bias and Fairness Issues - Fairness Definition

- **Principle**
  - Lack of favoritism from one side or another
- **Rich fairness definitions**
  - Group fairness
    - Statistical parity
    - Equal opportunity
    - Equalized odds
    - Accuracy parity
    - …
  - Individual fairness
  - Counterfactual fairness
  - Degree fairness (on graphs)





Fairness definition	**Two sides**
Group fairness	Two demographic groups
Individual fairness	Two data points
Counterfactual fairness	A data point and its counterfactual version
Degree fairness	Two group of nodes with same degree

Bias and Fairness Issues

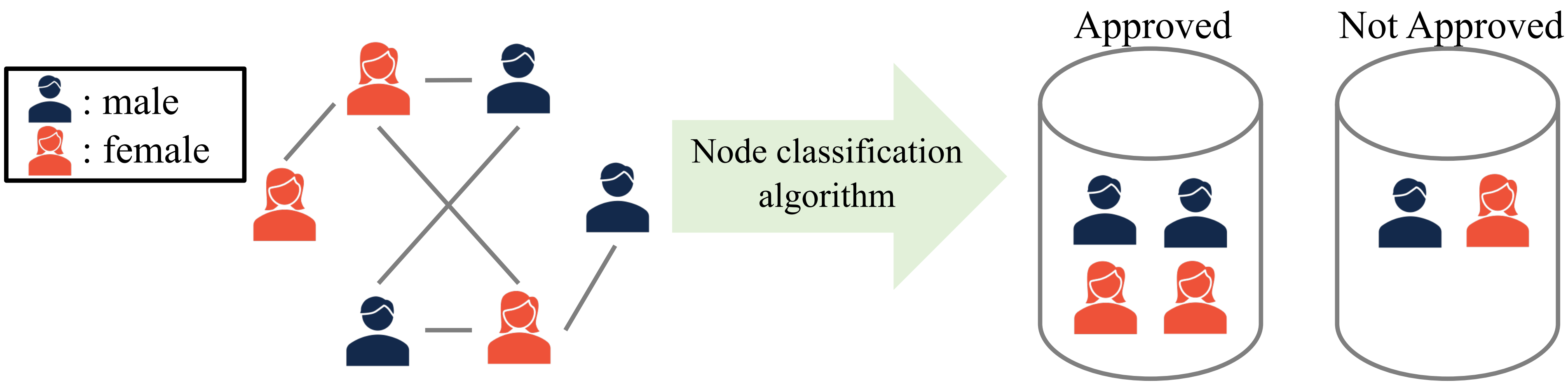
- **Group Fairness on Graphs**
- Individual Fairness on Graphs
- Degree Fairness on Graphs
- Future Directions and Q&A

Group Fairness: Statistical Parity

- **Statistical parity = equal acceptance rate**

$$\Pr_{+}(\hat{y} = c) = \Pr_{-}(\hat{y} = c)$$

  - $\hat{y}$: model prediction
  - $\Pr_{+}$: probability for the protected group
  - $\Pr_{-}$: probability for the unprotected group
  - Also known as demographic parity, disparate impact
- **Example: clinical trial participation**



$\Pr(\hat{y} = \text{approved}) = 2/3$
$\Pr(\hat{y} = \text{approved}) = 2/3$

Fair result
Same approval rate for male and female

Group Fairness: Equal Opportunity

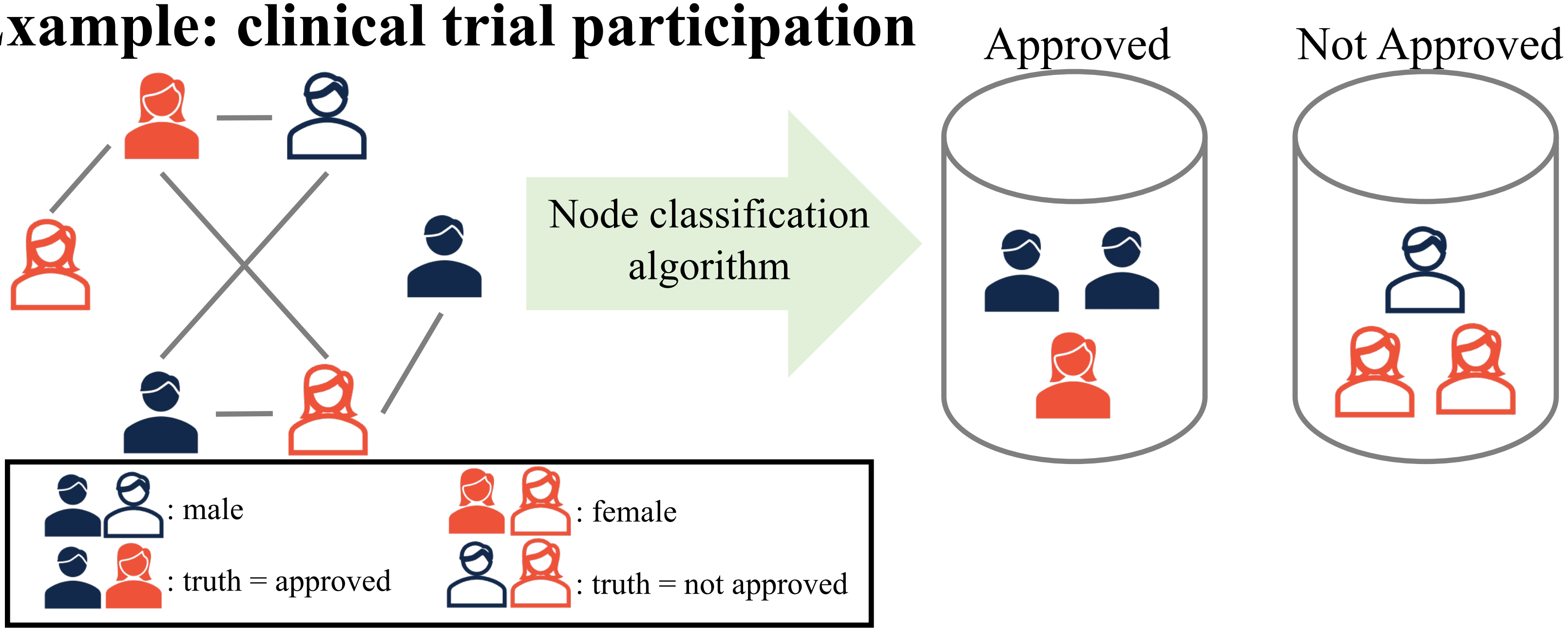
- **Equal opportunity = equal true positive rate**

$$\boxed{\Pr_{+}(\hat{y} = c | y = c) = \Pr_{-}(\hat{y} = c | y = c)}$$

If hold for all classes, it is called **equalized odds**

  - y: true label
  - $\hat{y}$: model prediction
  - $\Pr_{+}$: probability for the protected group
  - $\Pr_{-}$: probability for the unprotected group

- **Example: clinical trial participation**



$\Pr(\hat{y} = \text{approved} |) = 1$
$\Pr(\hat{y} = \text{approved} |) = 1$

Fair result
Same true positive rate for male and female

Hardt, M., et al. Equality of opportunity in supervised learning. NeurIPS 2016.

Adversarial Learning for Fair Representation Learning

- **Statistical parity**
  - Independence between the learned embedding $\mathbf{z}$ and a sensitive attribute a

$$\mathbf{z}_u \perp a_u, \forall \text{ node } u$$

  where a_u is the sensitive value of node u

- **Formulation**
  - Mutual information minimization

$$I(\mathbf{z}_u, a_u) = 0, \forall \text{ node } u$$

  - Analogous to statistical parity in classification task
  - Fail to predict a_u using $\mathbf{z}_u$ ← no information about a_u in $\mathbf{z}_u$

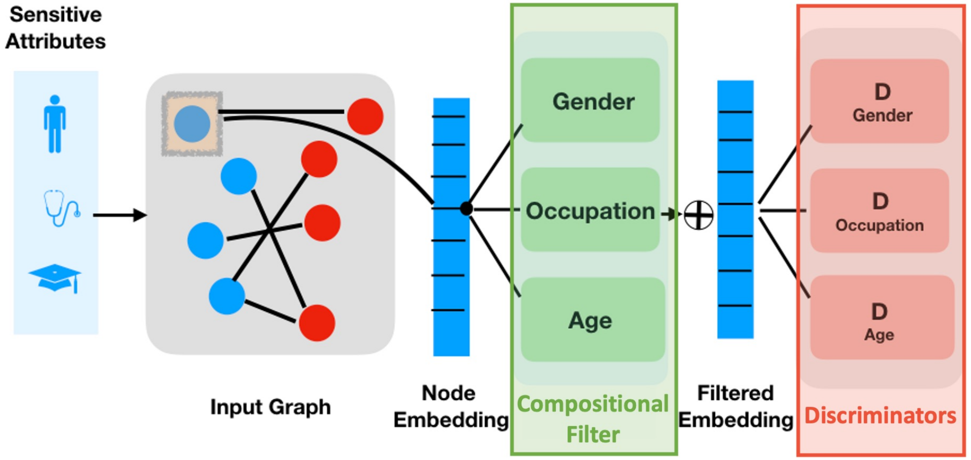
Corresponding to 'adversarial'

- **Solution**
  - Adversarial learning
  - Encoder: encode node into low-dimensional embedding space for downstream tasks
  - Discriminator: fail to predict a_u using $\mathbf{z}_u$

Bose, A., & Hamilton, W. (2019). Compositional fairness constraints for graph embeddings. ICML 2019.

Limitation #1: Full Access to Sensitive Attribute Information

- **Adversarial learning**
  - Minimize a task-specific loss function to learn 'good' representations
  - Maximize the error of predicting sensitive feature to learn 'fair' representations
- **Limitations**
  - Require the sensitive attribute of all training nodes to train a good discriminator
  - Ignore the fact that sensitive information is hard to obtain due to privacy
- **Question**
  - What if we only have limited sensitive attribute information?



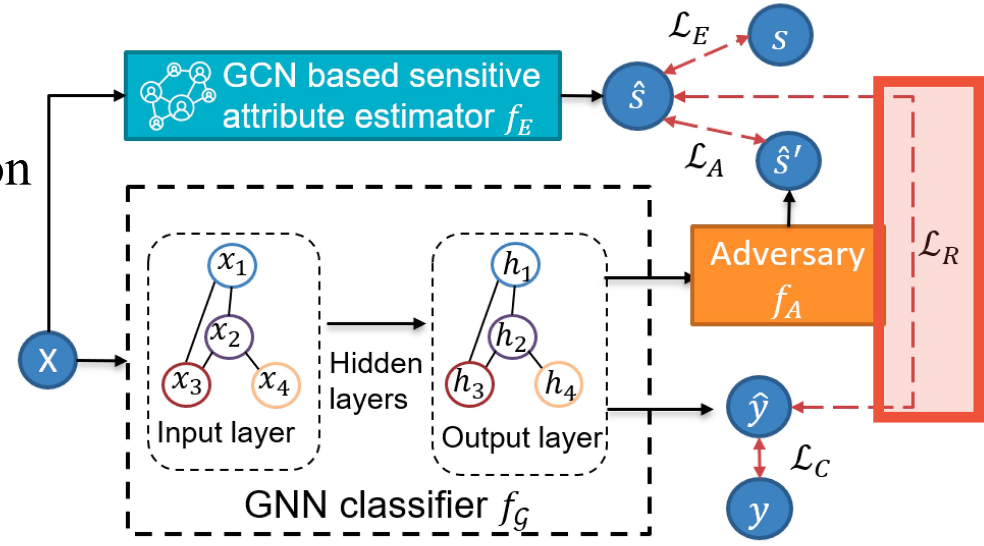
Dai, E. et al. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. WSDM 2021.

FairGNN: Additional Supervision Signal

- **Observation**
  - Adversarial learning is unstable to train ← even worse with limited sensitive attribute
  - Failure to converge may also cause discrimination
- **Key idea**
  - Additional prerequisite of independence for additional supervision
  - Independence → zero covariance
- **Solution**
  - Pseudo sensitive attribute from a sensitive attribute estimator
    - Not embedding from encoder
    - Offer pseudo-label for covariance minimization
  - Absolute covariance minimizer to minimize absolute covariance between model prediction $\hat{y}$ and pseudo sensitive attribute $\hat{s}$

$$\mathcal{L}_R = |\mathrm{cov}(\hat{s}, \hat{y})| = |\mathbb{E}[(\hat{s} - \mathbb{E}[\hat{s}])(\hat{y} - \mathbb{E}[\hat{y}])]|$$

  - Absolute covariance to avoid minimizing negative covariance

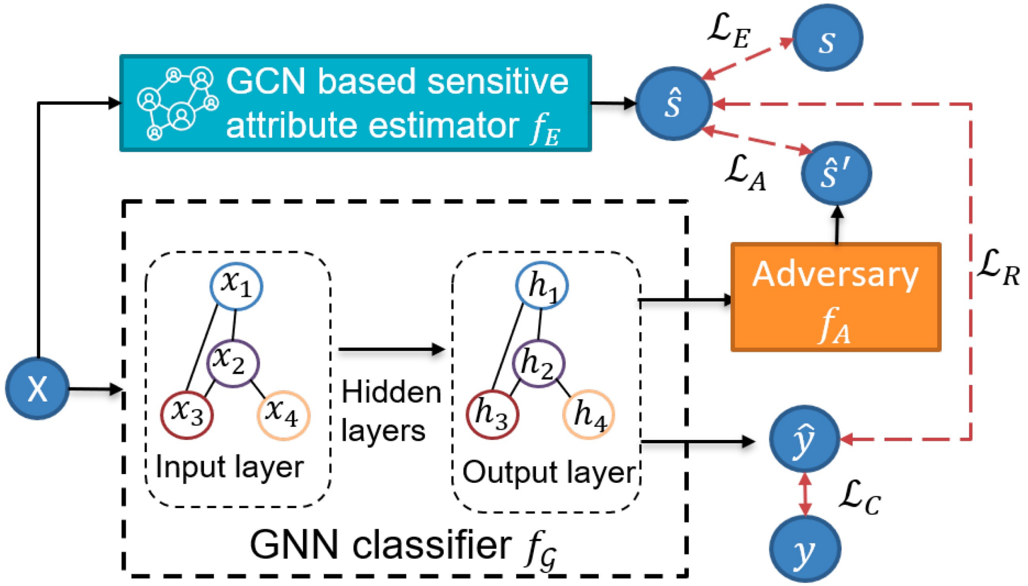


FairGNN: Overall Framework

- **Overall loss function**

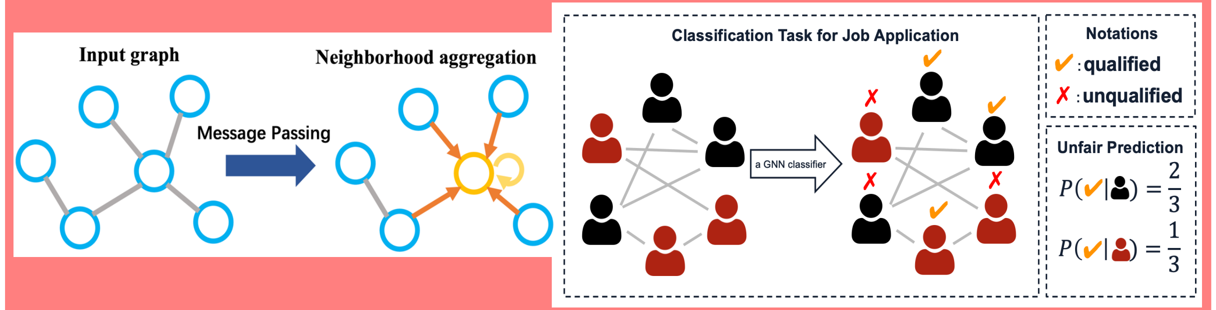
$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_E - \alpha\mathcal{L}_A + \beta\mathcal{L}_R$$

- **Intuition**
  - $\mathcal{L}_C$: classification loss (e.g., cross entropy) for learning representative node representation
  - $\mathcal{L}_E$: sensitive attribute estimation loss for generating accurate pseudo sensitive attribute information
  - $\mathcal{L}_A$: adversarial loss for debiasing the learned node representation
  - $\mathcal{L}_R$: covariance minimizer to stabilize the adversary training

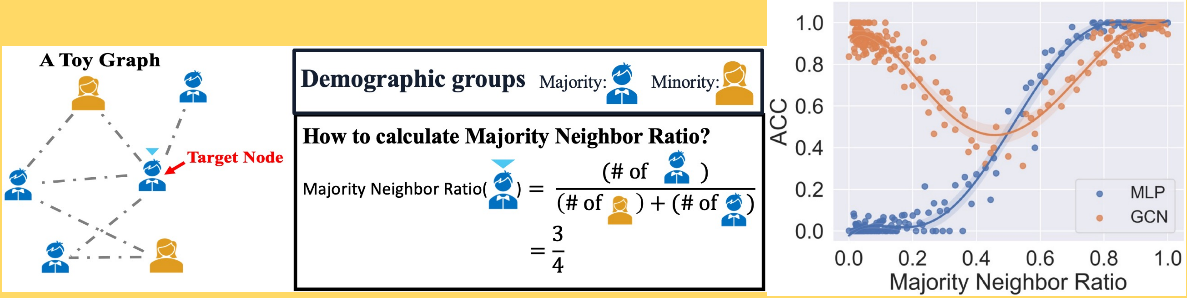


BeMap: Fair Topology View Generation

- **Motivation**
  - Message passing could be unfair

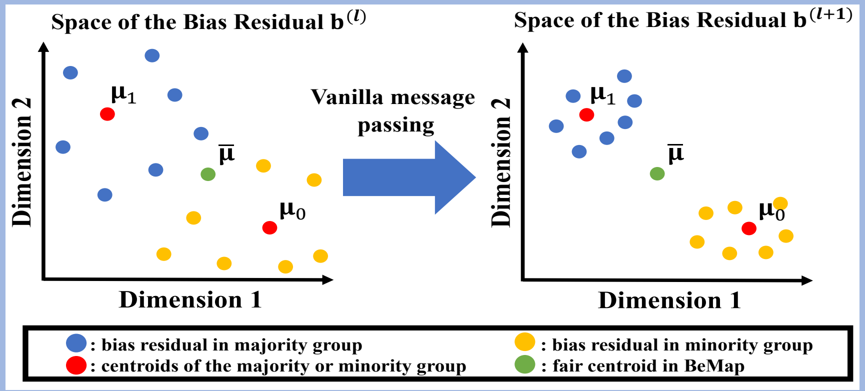


- **Empirical evidence**
  - Predict node sensitive attribute using embeddings learned from GCN and MLP (no MP)



- **Theoretical analysis**

node embedding = fair embedding + bias residual



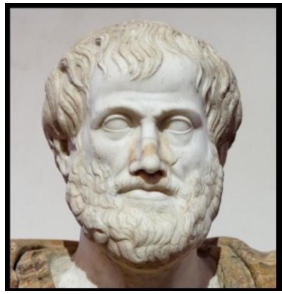
- **Method: BeMap**
  - (In every training epoch) neighbor sampling for balanced neighborhood and MP on it
  - Up to 80% bias reduction
  - Comparable or even better classification accuracy
  - More details in the paper

Bias and Fairness Issues

- Group Fairness on Graphs
- **Individual Fairness on Graphs**
- Degree Fairness on Graphs
- Future Directions and Q&A

Individual Fairness

- **Definition**
  - Similar individuals should have similar outcomes
  - Rooted in Aristotle's conception of justice as consistency

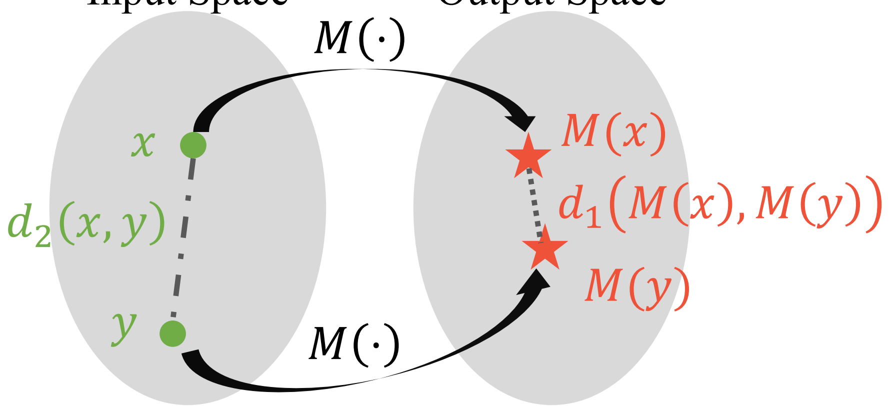


"Equality consists in the same treatment of similar persons, and no government can stand which is not founded upon justice."

- **Formulation: Lipschitz inequality (most common)**

$$d_1\big(M(x), M(y)\big) \leq L d_2(x, y)$$

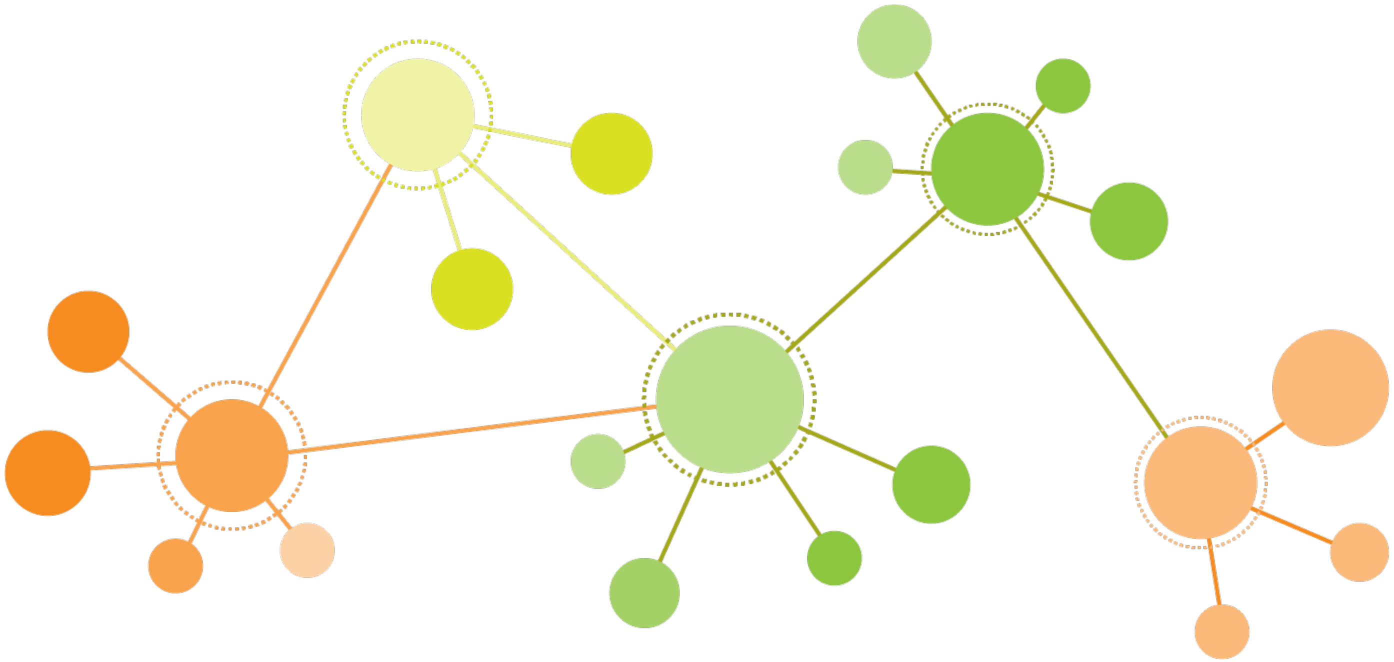
  - M: a mapping from input to output
  - d_1: distance metric for output
  - d_2: distance metric for input
  - L: a constant scalar



Kang, J., He, J., Maciejewski, R., & Tong, H. Inform: Individual fairness on graph mining. KDD 2020.
Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. Fairness through awareness. ITCS 2012.

InFoRM: Individual Fairness on GRaph Mining

- **Research questions**
  - **RQ1. Measure:** how to quantitatively measure individual bias?
  - **RQ2. Algorithms:** how to ensure individual fairness?
  - **RQ3. Cost:** what is the cost of individual fairness?



Kang, J., He, J., Maciejewski, R., & Tong, H. Inform: Individual fairness on graph mining. KDD 2020.

InFoRM Measure: Quantifying Individual Bias

- **Principle**
  - Similar nodes → similar mining results
- **Mathematical formulation**

Similarity between node i and node j

$$\|\mathbf{Y}[i,:] - \mathbf{Y}[j,:]\|_F^2 \leq \frac{\epsilon}{\mathbf{S}[i,j]} \quad \forall i,j = 1,\ldots,n$$

(1) For any node pair (i,j)
$\|\mathbf{Y}[i,:] - \mathbf{Y}[j,:]\|_F^2 \mathbf{S}[i,j] \leq \epsilon$

(2) Sum it up for all node pairs

  - If $\mathbf{S}[i,j]$ is high, $\frac{\epsilon}{\mathbf{S}[i,j]}$ is small → push $\mathbf{Y}[i,:]$ and $\mathbf{Y}[j,:]$ to be more similar
  - Inequality should hold for every pairs of nodes i and j → too restrictive
- **Relaxed criteria**

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \|\mathbf{Y}[i,:] - \mathbf{Y}[j,:]\|_F^2 \mathbf{S}[i,j] \leq m\epsilon$$

$$\| \qquad //$$

$$2\mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y}) \leq \delta$$

Overall individual bias of the graph

  - m: number of edges in the graph
  - $\delta = m\epsilon$

Kang, J., He, J., Maciejewski, R., & Tong, H. Inform: Individual fairness on graph mining. KDD 2020.
Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. ITCS 2012.

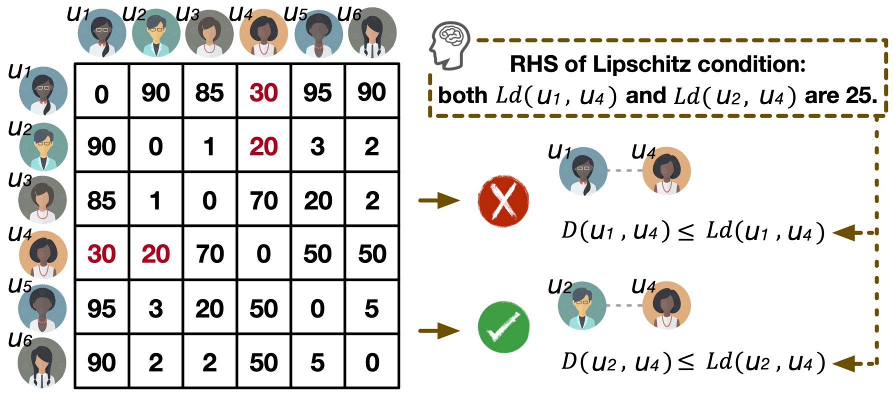
Alternative Measure: Ranking-Based Individual Fairness

- **Key challenge in InFoRM measure**
  - Lipschitz condition (used in InFoRM)

$$d_1\big(M(x), M(y)\big) \leq L d_2(x, y)$$

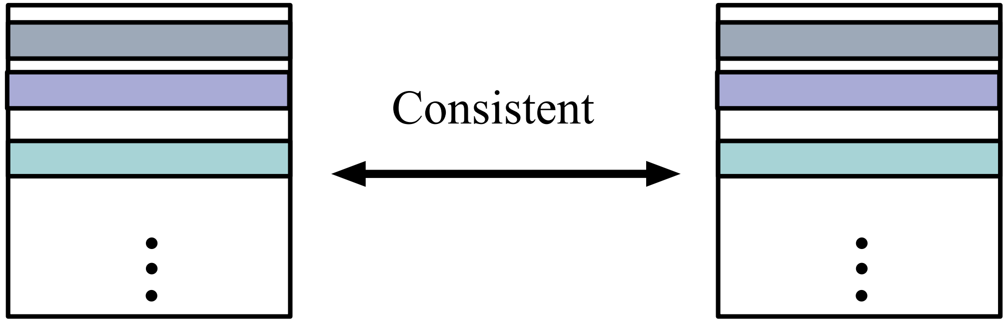
  - Distance comparison fails to calibrate between different individuals
- **Definition**
  - Given
    - (1) the node similarity matrix $\mathbf{S}_G$ of the input graph G
    - (2) the similarity matrix $\mathbf{S}_{\hat{\mathbf{Y}}}$ of the GNN output $\hat{\mathbf{Y}}$
  - $\hat{\mathbf{Y}}$ is individually fair if, for each node i, it satisfies that

$$\text{ranking list derived by } \mathbf{S}_G[i,:] = \text{ranking list derived by } \mathbf{S}_{\hat{\mathbf{Y}}}[i,:]$$



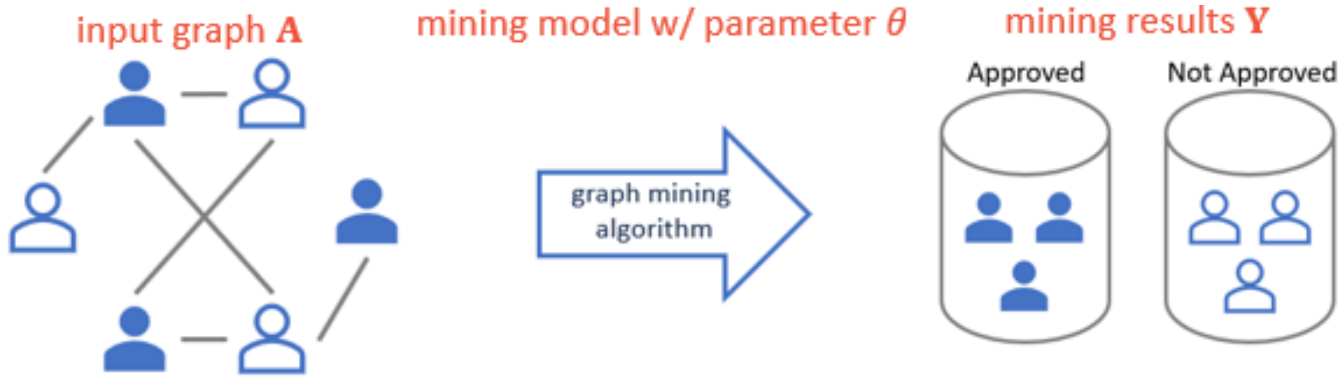
(a) Outcome distance matrix from distance metric D

(b) Lipschitz condition judgement based on human knowledge



InFoRM Measure: Mitigating Individual Bias

- **Graph mining workflow**



- Debiasing methods
  - Debiasing the input graph: $\min_{\mathbf{Y}} \; J = \underbrace{\left\|\tilde{\mathbf{A}} - \mathbf{A}\right\|_F^2}_{\text{topology consistency}} + \alpha \mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y})$

    s.t. $\partial_{\mathbf{Y}} l\left(\tilde{\mathbf{A}}, \mathbf{Y}, \theta\right) = 0$
  - Debiasing the mining model: $\min_{\mathbf{Y}} \; J = \underbrace{l(\mathbf{A}, \mathbf{Y}, \theta)}_{\text{task-specific loss function}} + \alpha \mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y})$
  - Debiasing the mining results: $\min_{\mathbf{Y}} \; J = \underbrace{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_F^2}_{\text{mining results consistency}} + \alpha \mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y})$

Individual bias
(InFoRM measure)

Kang, J., He, J., Maciejewski, R., & Tong, H. Inform: Individual fairness on graph mining. KDD 2020.

InFoRM Cost: Characterizing Individual Bias

- **Main focus**
  - Debiasing the mining results (model-agnostic)
- **Given**
  - A graph with n nodes and adjacency matrix $\mathbf{A}$
  - A node-node similarity matrix $\mathbf{S}$
  - Vanilla mining results $\bar{\mathbf{Y}}$
  - Debiased mining results $\mathbf{Y}^* = (\mathbf{I} + \alpha\mathbf{S})^{-1}\bar{\mathbf{Y}}$
- If $\|\mathbf{S} - \mathbf{A}\|_F = \Delta$, we have

$$\|\bar{\mathbf{Y}} - \mathbf{Y}^*\|_F \leq 2\alpha\sqrt{n}\left(\Delta + \sqrt{rank(\mathbf{A})}\sigma_{\max}(\mathbf{A})\right)\|\bar{\mathbf{Y}}\|_F$$

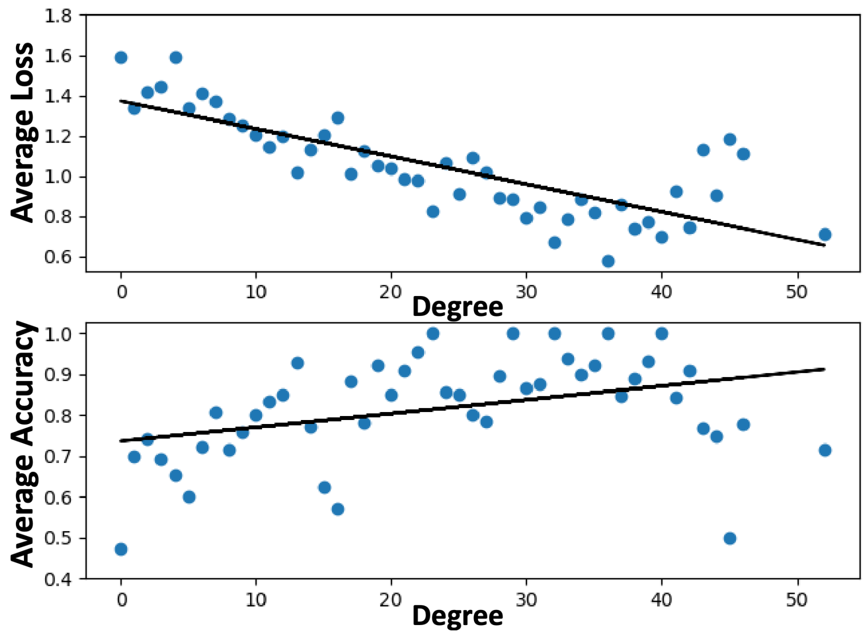
- **Key factors**
  - The number of nodes n (i.e., size of the input graph)
  - The difference Δ between $\mathbf{A}$ and $\mathbf{S}$
  - The rank of $\mathbf{A}$ → could be small due to (approximate) low-rank structures in real-world graphs
  - The largest singular value of $\mathbf{A}$ → could be small if $\mathbf{A}$ is normalized

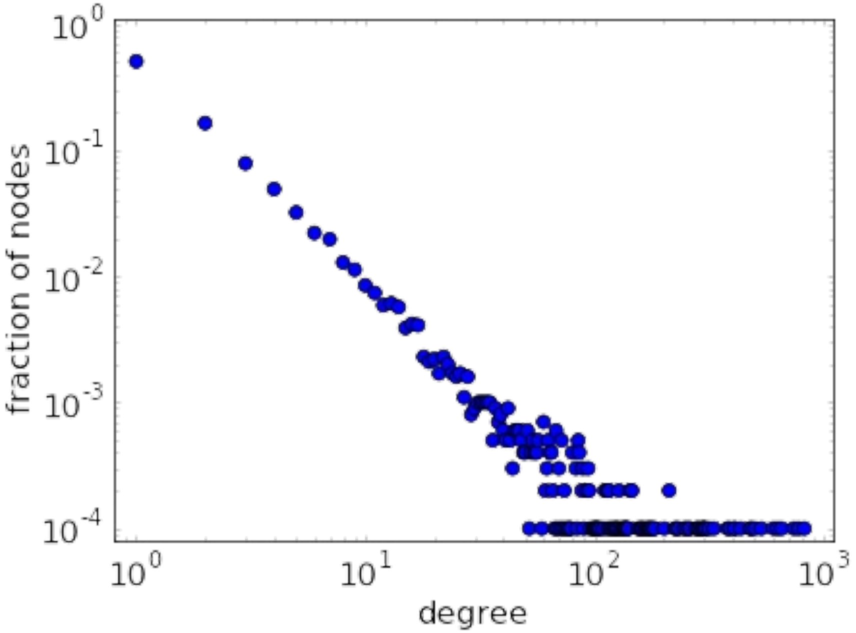
Kang, J., He, J., Maciejewski, R., & Tong, H. Inform: Individual fairness on graph mining. KDD 2020.

- Group Fairness on Graphs
- Individual Fairness on Graphs
- **Degree Fairness on Graphs**
- Future Directions and Q&A

Degree Fairness: Definition and Motivation

- **Definition**
  - Nodes of different degrees should have balanced utility on a graph mining task
- **Example: online advertising**
  - (A small portion of) celebrities often enjoy high-quality model performance
  - (A large portion of) grassroot users often suffer from bad model performance





Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H. Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. WWW 2022.
Subramonian, A., Kang, J., & Sun, Y. Theoretical and empirical insights into the origins of degree bias in graph neural networks. arXiv 2024.

Degree Unfairness: Pitfall of Graph Neural Networks

- **Given**
  - (1) $\mathcal{G} = (\mathbf{A}, \mathbf{X})$
  - (2) Any test node i in $\mathcal{G}$ with label c
  - (3) A graph learning model M which output (before softmax) $\mathbf{Z}$
  - (4) Any wrong prediction $c' \neq c$
- **Our results**
  - Misclassification rate

$$\Pr\left(\Pr(\hat{y} = c | i, M) > \Pr(\hat{y} = c' | M, i)\right) \leq \frac{1}{1 + R_{i,c'}}$$

where $R_{i,c'} = \frac{\left(\mathbb{E}\left[\mathbf{Z}[i,c'] - \mathbf{Z}[i,c]\right]\right)^2}{\mathrm{Var}\left[\mathbf{Z}[i,c'] - \mathbf{Z}[i,c]\right]}$ (reciprocal of measure of dispersion from economics)

  - $R_{i,c'}$ is positively correlated with the degree of node i
- **Conclusion**
  - high-degree nodes often have lower misclassification rate!

Causes #1: High-Degree Nodes with High Influence in Node Embeddings

- **Given**
  - $\mathcal{V}_{\text{labeled}}$: a set of labeled nodes $\mathcal{V}_{\text{labeled}}$
  - $\mathbf{W}^{(L)}$: the weight of L-th layer in an L-layer GCN
  - d_i: degree of node i
  - $\mathbf{x}_i$: input node feature of node i
  - $\mathbf{h}_i^{(L)}$: output embeddings of node i learned by the L-layer GCN
- **Influence of node *i* on GCN training**

$$S(i) = \sum_{k \in \mathcal{V}_{\text{labeled}}} \left\| \mathbb{E}\left[\partial \mathbf{h}_i^{(L)} / \partial \mathbf{x}_k \right] \right\| \propto \sqrt{d_i} \left\| \mathbf{W}^{(L)} \right\| \sum_{k \in \mathcal{V}_{\text{labeled}}} \sqrt{d_k}$$

- **Remark**
  - For two nodes i and j, if $d_i > d_j$, then $S(i) > S(j)$

    → Node with higher degree will have higher influence on GCN training

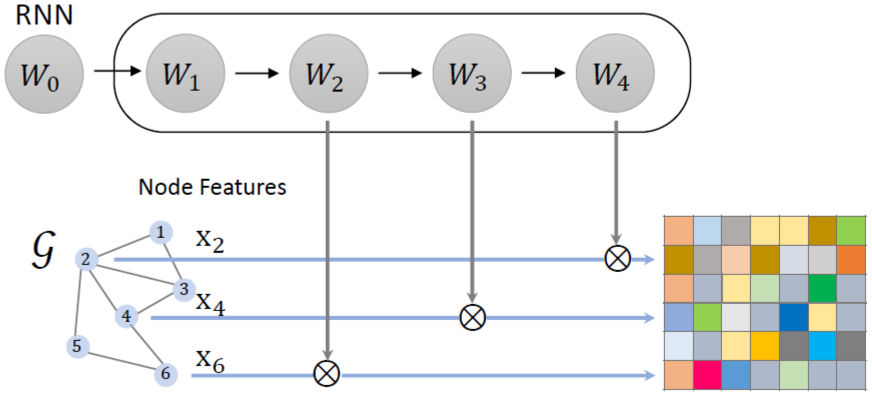
Tang, X., et. al. Investigating and mitigating degree-related biases in graph convolutional Networks. CIKM 2020

Solution #1: Degree-Specific Graph Convolution

- **Key idea**
  - Degree-specific weights to encode degree information
- **Given**
  - d_i: the degree of node i
  - $\mathbf{W}_{d_j}^{(l)}$: the degree-specific weight w.r.t. degree of node j
- **Degree-specific graph convolution**

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} \left(\mathbf{W}^{(l)} + \mathbf{W}_{d_j}^{(l)}\right) \mathbf{h}_j^{(l)}\right)$$

  - DEMO-Net → $\mathbf{W}_{d_j}^{(l)}$ is generated randomly
  - SL-DSGCN → $\mathbf{W}_{d_j}^{(l)}$ is generated using a recurrent neural network



Tang, X., et. al. Investigating and mitigating degree-related biases in graph convolutional Networks. CIKM 2020
Wu, J., He, J., & Xu, J. Net: Degree-specific graph neural networks for node and graph classification. KDD 2019.

Causes #2: High-Degree Nodes with High Influence in Gradient

- **Gradient of loss w.r.t. weight**

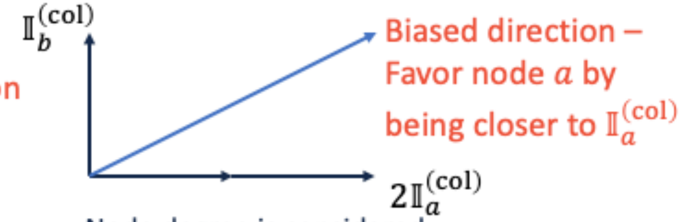
Row sum in $\hat{\mathbf{A}}$ — Column sum in $\hat{\mathbf{A}}$

$$\frac{\partial J}{\partial \mathbf{W}^{(l)}} = \sum_{i=1}^{n} d_{\hat{\mathbf{A}}}(i)\mathbb{I}_i^{(\mathrm{col})} = \sum_{j=1}^{n} d_{\hat{\mathbf{A}}}(j)\mathbb{I}_j^{(\mathrm{row})}$$

  - $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\tilde{\mathbf{D}}^{-\frac{1}{2}}$ → symmetric normalization kernel
  - $\mathbb{I}_i^{(\mathrm{col})}$ and $\mathbb{I}_j^{(\mathrm{row})}$ → the directions for gradient descent
  - $d_{\hat{\mathbf{A}}}(i)$ and $d_{\hat{\mathbf{A}}}(j)$ → the importance of the direction
  - High degree → more focus on that direction

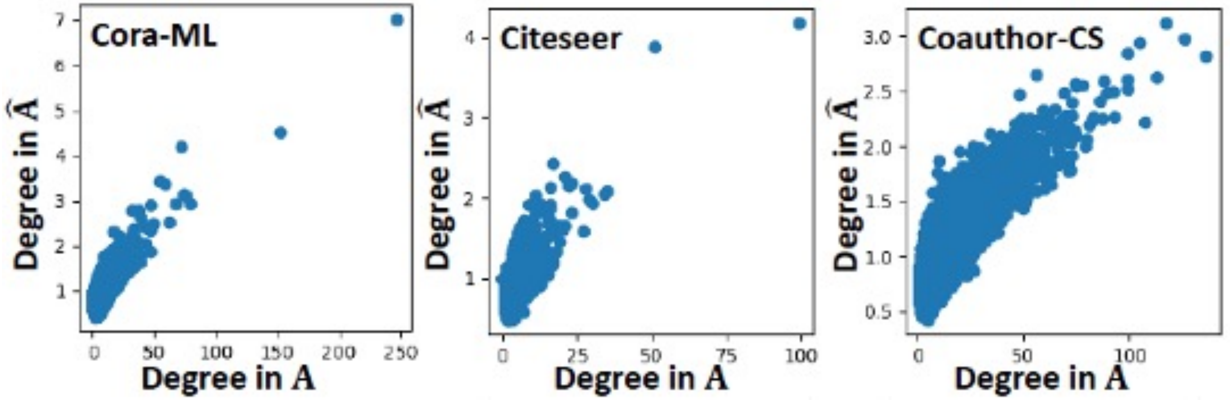
Node a: $d_{\hat{\mathbf{A}}}(a) = 2$
Node b: $d_{\hat{\mathbf{A}}}(b) = 1$

Node degree takes no effect



Node degree is considered

- **Symmetric normalization**
  - Normalize the largest eigenvalue but not degree
  - High degree in A → high degree in $\hat{\mathbf{A}}$



Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H. Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. WWW 2022

Solution #2: Graph Normalization

- **Key idea**
  - Mitigate impacts of node degree by normalizing it to constant (i.e., 1)
  - Normalize the graph to a doubly stochastic graph
- **Sinkhorn-Knopp (SK) algorithm**
  - Iteratively normalize row and columns
  - **(Our result)** SK always finds the unique doubly stochastic form of symmetric normalization kernel
- **Fair gradient computation**

$$\left(\frac{\partial J}{\partial \mathbf{W}^{(l)}}\right)_{\text{fair}} = \left(\mathbf{H}^{(l-1)}\right)^T \hat{\mathbf{A}}_{\text{DS}}^T \frac{\partial J}{\partial \mathbf{E}^{(l)}}$$

  - $\hat{\mathbf{A}}_{\text{DS}}$ → doubly-stochastic normalization of $\hat{\mathbf{A}}$
- **RawlsGCN family**
  - RawlsGCN-Graph: during data pre-processing, compute $\hat{\mathbf{A}}_{\text{DS}}$ and treat it as the input of GCN
  - RawlsGCN-Grad: during optimization (in-processing), treat $\hat{\mathbf{A}}_{\text{DS}}$ as a normalizer to equalize the importance of node influence

Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H. Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. WWW 2022.

Bias and Fairness Issues

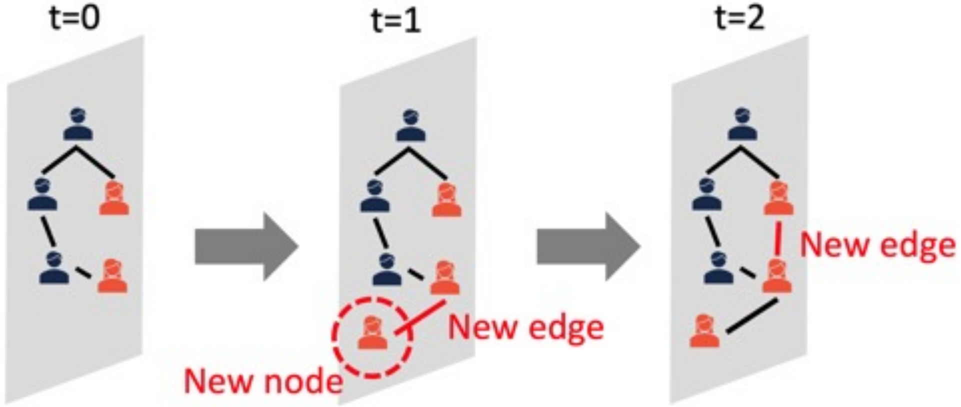
- Group Fairness on Graphs
- Individual Fairness on Graphs
- Degree Fairness on Graphs
- **Future Directions and Q&A**

Future Direction #1: Fairness beyond Plain and Static Graphs

- **Observation**
  - Real-world graphs are often dynamic and/or multi-sourced
- **Research questions**
  - How to ensure fairness for multiple type of nodes/edges or multi-graphs?
  - How to efficiently update the fair mining results at each timestamp?
  - How to characterize the impact of graph dynamics and multiple sources over the bias measure?

Multi-sourced social networks

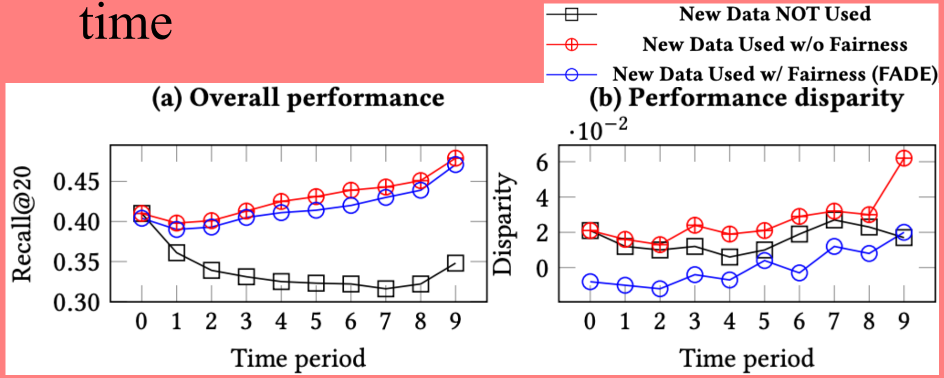
Dynamic graphs over 3 timestamps



Preliminary Work: Dynamic Group Fairness in Recommender Systems

• Observation

- performance disparity is getting larger over time



• Theory

- Fine-tuning is better than re-training for fairness over time

Re-training

- (1) $L^{rt}_{t_{test}}$ = real loss of re-training at test time; (2) $L^{*}_{t_{test}}$ = optimal loss at time test; (3) m_0 = #. edges at time 0; (4) m_t = #. edge changes at time t; (5) $0 < \gamma < 1$

$$L^{rt}_{t_{test}} \le L^{*}_{t_{test}} + 2\frac{m_0 d_{0,t_{test}} + \sum_{t=1}^{t_{test}-1} m_1 d_{1,t_{test}}}{m_0 + (t_{test}-1)m_1} + 4\sqrt{\frac{1}{m_0 + (t_{test}-1)m_1}\log\frac{2}{\delta}}$$

Fine-tuning

- Similar settings as re-training but $L^{ft}_{t_{test}}$ = real loss of fine-tuning at test time

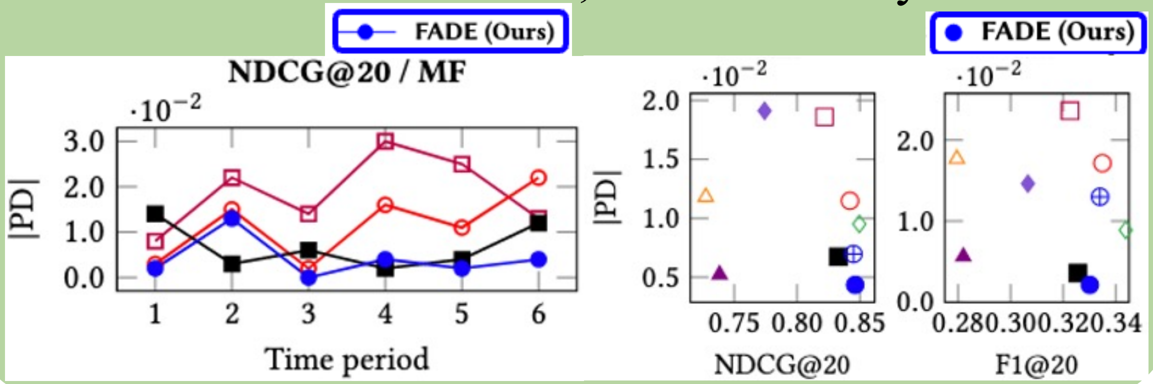
$$L^{ft}_{t_{test}} \le L^{*}_{t_{test}} + 2\frac{(1-\gamma)\left(2\sum_{t=0}^{t_{test}-1} \gamma^{t_{test}-t-1} d_{t,t_{test}} + 4\sqrt{\left(\frac{\gamma^{2t_{test}-2}}{m_0} + \frac{1-\gamma^{2t_{test}-2}}{(1-\gamma^2)m_1}\right)\log\frac{2}{\delta}}\right)}{1-\gamma^{t_{test}}}$$

• Method: FADE

- Model-agnostic
- Fine-tuning with newly observed data
- Periodically re-training to keep historical information
- Linear complexity w.r.t. # new data

• Results

- Fairness over time, small accuracy decrease



Yoo, H., Zeng, Z., Kang, J., Liu, Z., Zhou, D., Wang, F., ... & Tong, H. Ensuring user-side fairness in dynamic recommender systems. WWW 2024

Future Direction #2: Fairness on Graphs → Fairness with Graphs

- **Fairness on graphs**
  - Graph as data
  - Nodes = entities
  - Social networks → nodes = users
  - Citation networks → nodes = papers
  - Web graph → nodes = webpages
- **Fairness with graphs**
  - Graph as context
  - Nodes = models/datasets/modalities
- **Example: supply chain**

1. Demand + supply for medical resources
2. Models to allocate medical resources

  - How can we leverage demand + supply + model collectively for fair supply chain?

Future Direction #3: Benchmark and Evaluation Metrics

- **Observation**
  - No consensus on the experimental settings for fair graph learning
  - Which data to compare? What sensitive attribute to consider?
  - Which evaluation metrics for each type of fairness?
- **Consequences**
  - Different settings for different research works
  - Hardly fair comparison among fair graph learning methods
  - Hardly deployable methods in real-world scenarios
- **Call for community effort**
  - Evaluation benchmark for consistent experimental settings and fair comparison
  - Collection of large-scale, realistic, but challenging dataset for evaluation

Outline

- Introduction and Background
- Topology Issues
- Imbalance Issues
- Short Break
- Bias and Fairness Issues
- **Limited Labeled Data Issues**
- Abnormal Graph Data Issues
- Summary

Limited Labeled Data Issues

- Graph Data Augmentations
- Self-supervised Learning on Graphs

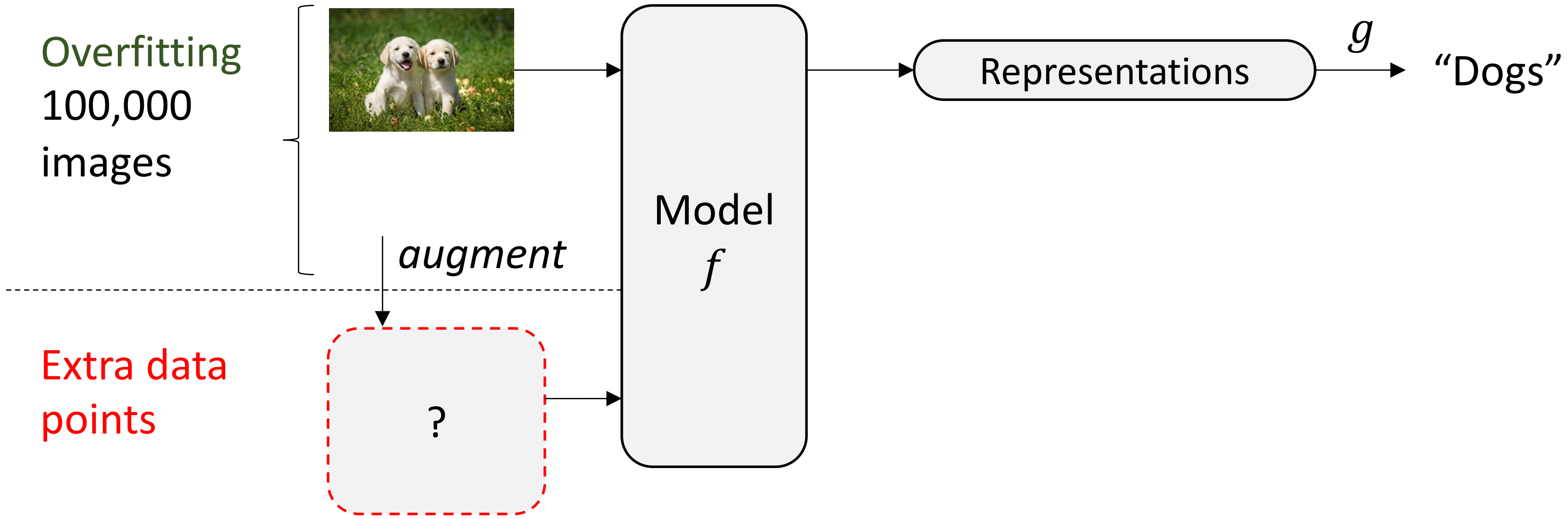
Data Augmentation

Wikipedia: Techniques used to increase the amount of data by adding *slightly modified* copies of already existing data or *newly created* synthetic data from existing data.

- Why data augmentation?
  - It helps reduce overfitting when training a machine learning model.
  - The acquisition of labeled graph data can be expensive.

Data Augmentation

Wikipedia: Techniques used to increase the amount of data by adding *slightly modified* copies of already existing data or *newly created* synthetic data from existing data.



Data Augmentation

Wikipedia: Techniques used to increase the amount of data by adding *slightly modified* copies of already existing data or *newly created* synthetic data from existing data.

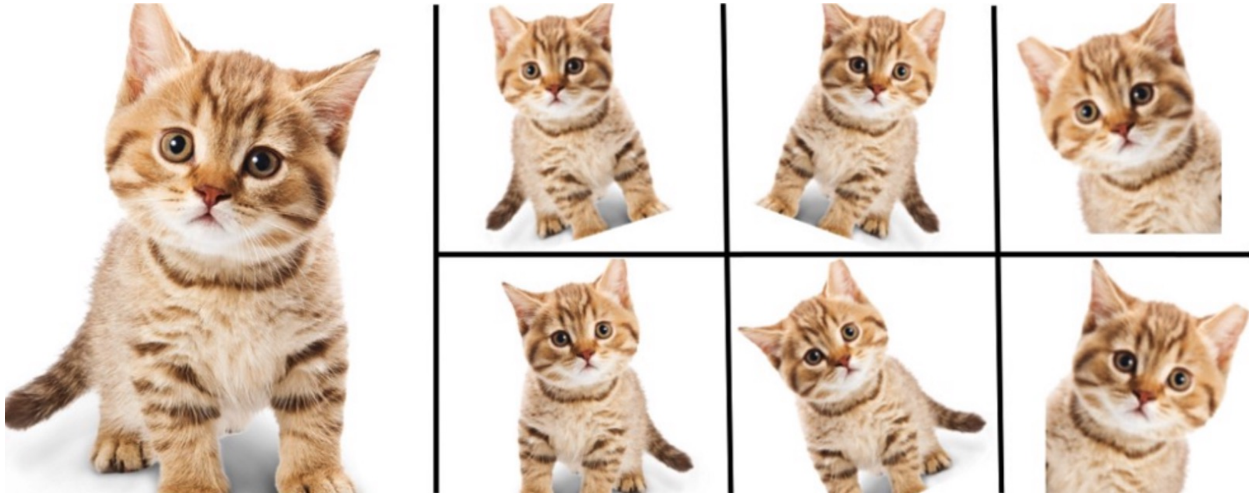
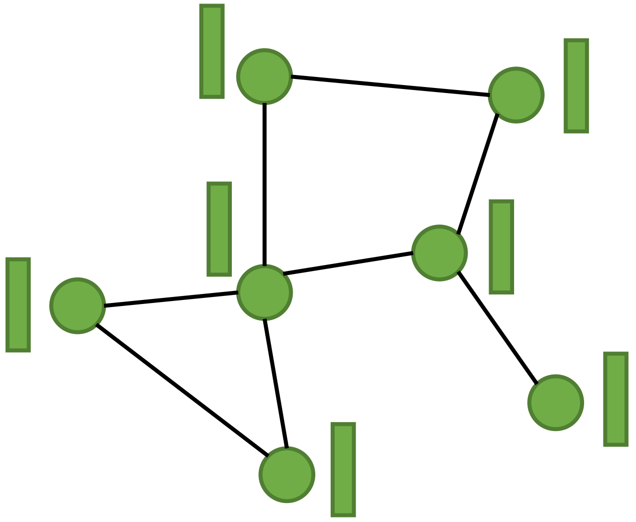


Image sources:
https://www.kdnuggets.com/2018/05/data-augmentation-deep-learning-limited-data.html
https://amitness.com/2020/05/data-augmentation-for-nlp/

Graph Data Augmentation

- Structure Augmentation
  - Drop/add nodes/edges, etc.
- Feature Augmentation
  - Mask off features, etc.
- Label Augmentation
  - Label propagation, etc.



Graph Data Augmentation

- Rule-based augmentations
  - Designed based on heuristic rules
  - Usually efficient and scalable
  - Simple and easy to implement
    - Commonly used in self-supervised learning

- Learned augmentations
  - Involve learning during augmentation
  - Augmented data better fits GML models
    - Better performances in supervised learning

Rule-based Graph Data Augmentation Approaches

	Methodology	Representative Works	Task Level			Augmented Data		
			Node	Graph	Edge	Structure	Feature	Label
Rule-based GDA	Stochastic Dropping/Masking	DropEdge [87]	✓			✓		
		DropNode [27]		✓			✓	
		NodeDropping [127]		✓		✓		
		Feature Masking [100]	✓				✓	
		Feature Shuffling [106]	✓				✓	
		DropMessage [23]	✓		✓		✓	
		Subgraph Masking [127]		✓		✓	✓	
	Subgraph Cropping/Substituting	GraphCrop [111]		✓		✓		
		M-Evolve [145]		✓		✓		
		MoCL [97]		✓		✓	✓	
	Virtual Node	Graphormer [125]		✓		✓		
		GNN-CM$^+$/CM [45]			✓	✓		
	Mixup	Graph Mixup [115]	✓	✓				✓
		ifMixup [37]		✓		✓	✓	✓
		Graph Transparent [85]		✓		✓	✓	✓
		G-Mixup [39]		✓		✓	✓	✓
	SMOTE	GraphSMOTE [140]	✓				✓	
		GATSMOTE [75]	✓			✓		
		GNN-CL [70]	✓			✓	✓	
	Diffusion	GDA [60]	✓			✓		
	Counterfactual Augmentation	CFLP [141]			✓	✓		✓
	Attribute Augmentation	LA-GNN [74]	✓				✓	
		SR+DR [93]	✓				✓	
	Pseudo-labeling	Label Propagation [147]	✓					✓
		PTA [21]	✓					✓

Zhao, et al. Graph Data Augmentation for Graph Machine Learning: A Survey. 2022.

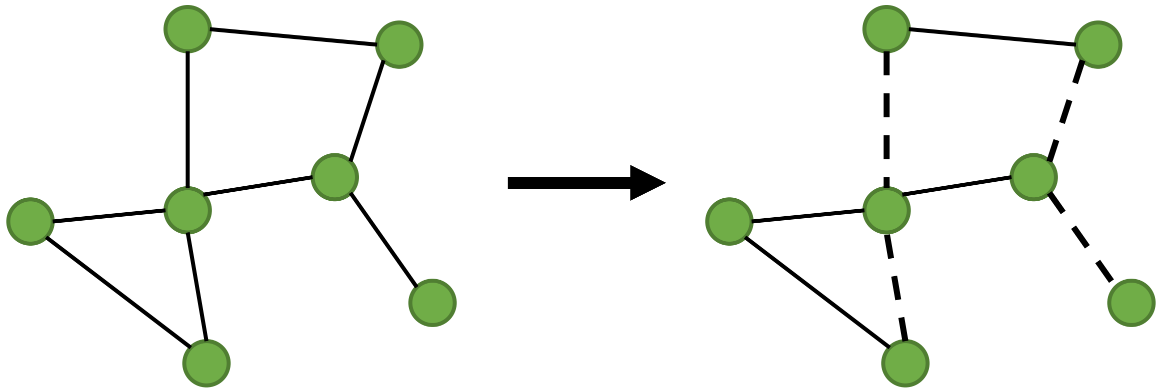
DropEdge

- Dropout on edges: randomly remove some edges at the beginning of every training epoch.

$$\tilde{\mathbf{A}} = \mathbf{M} \odot \mathbf{A}$$

$$\mathbf{M} \in \{0,1\}^{N \times N} \text{ s.t. } M_{i,j} = Bernoulli(\varepsilon)$$

- Prevents overfitting and over-smoothing.



Yu, et al. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. ICLR 2020.

Other Stochastic Masking/Dropping Methods

- Node Dropping
  - Randomly removing part of the nodes.
- Feature Masking
  - Randomly mask off node features.
  - Random row-shuffling on node feature matrix $\mathbf{X}$.
- Subgraph Masking
  - Randomly mask off a connected subgraph.

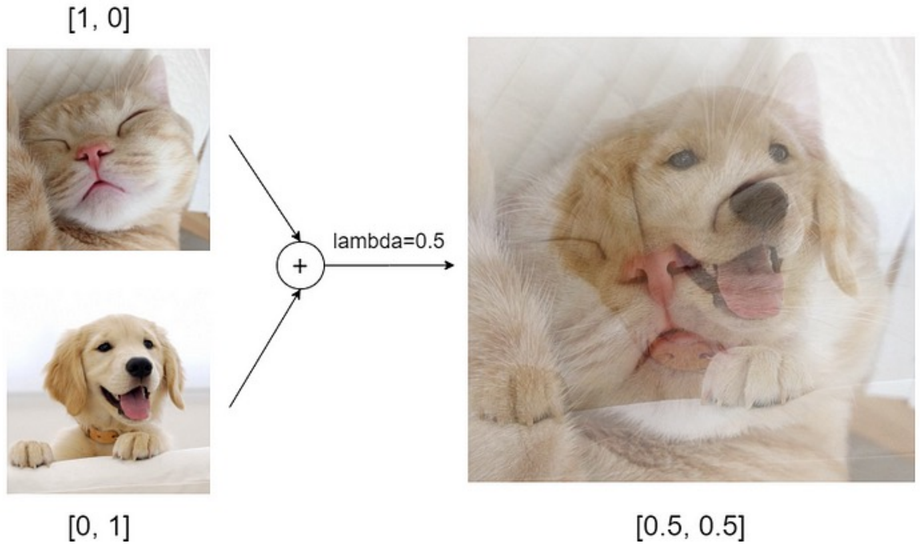
Feng, et al. Graph Random Neural Networks for Semi-supervised Learning on Graphs. NeurIPS 2020.
You, et al. Graph Contrastive Learning with Augmentations. NeurIPS 2020.
Thakoor, et al. Large-scale Representation Learning on Graphs via Bootstrapping. ICLR 2022.
Velickovic, et al. Deep Graph Infomax. ICLR 2019.

Mixup

- Mixup: generates a weighted combination of random pairs from the training data.

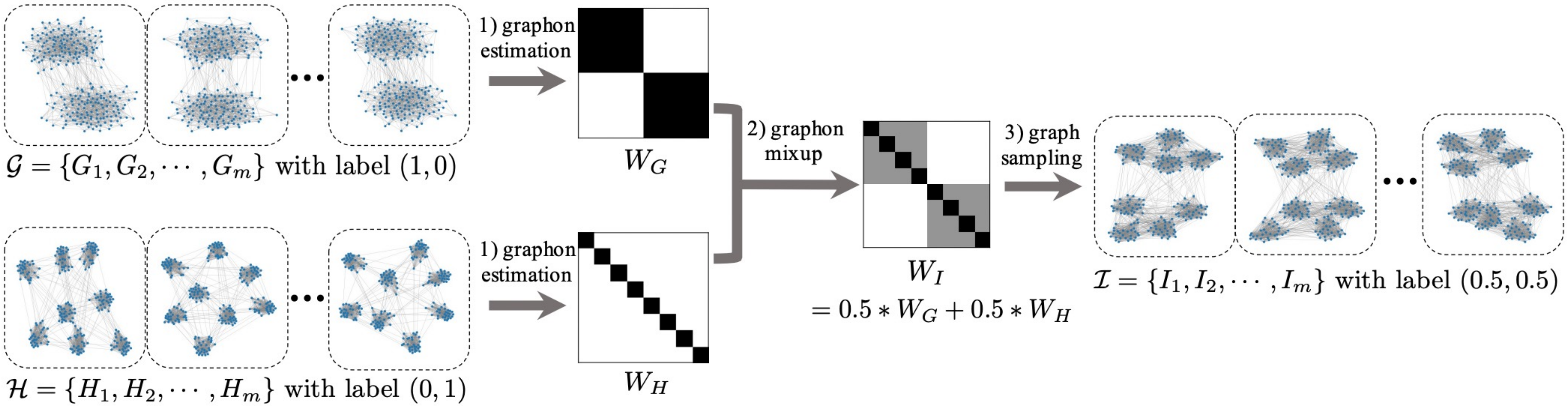
$$\tilde{\boldsymbol{x}} = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j,$$
$$\tilde{\boldsymbol{y}} = \lambda \boldsymbol{y}_i + (1-\lambda)\boldsymbol{y}_j.$$

- Manifold Mixup: interpolating hidden states.



Zhang, et al. Mixup: Beyond Empirical Risk Minimization. ICLR 2018.
Verma, et al. Manifold Mixup: Better Representations by Interpolating Hidden States. ICML 2019.
Image source: https://medium.com/@wolframalphav1.0/easy-way-to-improve-image-classifier-performance-part-1-mixup-augmentation-with-codes-33288db92de5

G-Mixup



1. Graphon estimation: $\mathcal{G} \to W_{\mathcal{G}}, \mathcal{H} \to W_{\mathcal{H}}$
2. Graphon Mixup: $W_{\mathcal{I}} = \lambda W_{\mathcal{G}} + (1 - \lambda) W_{\mathcal{H}}$
3. Graph Generation: $\{I_1, I_2, \cdots, I_m\} \overset{\text{i.i.d}}{\sim} \mathbb{G}(K, W_{\mathcal{I}})$
4. Label Mixup: $\mathbf{y}_{\mathcal{I}} = \lambda \mathbf{y}_{\mathcal{G}} + (1 - \lambda) \mathbf{y}_{\mathcal{H}}$

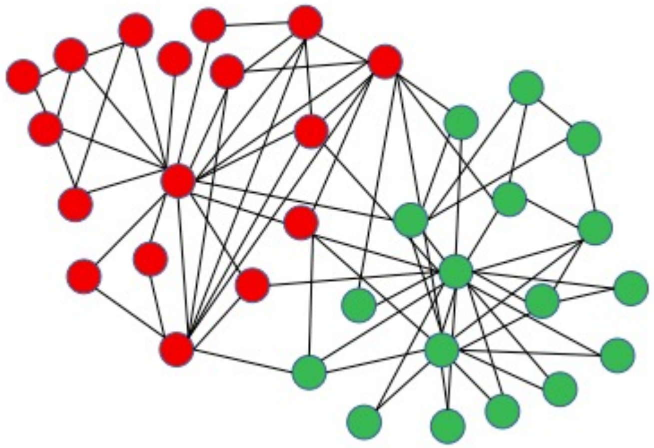
Learned Graph Data Augmentation Approaches

Learned GDA	Graph Structure Learning	GAug [140]	✓			✓	
		GLCN [47]	✓			✓	
		LDS [28]	✓			✓	
		ProGNN [50]	✓			✓	
		Eland [141]	✓			✓	
	Graph Adversarial Training	RobustTraining [125]	✓			✓	
		AdvT [18]	✓		✓	✓	
		FLAG [63]	✓	✓	✓		✓
		GraphVAT [25]	✓				✓
	Graph Rationalization	GREA [71]		✓		✓	✓
		AdvCA [97]		✓		✓	✓
	Automated Augmentation	AutoGDA [144]	✓			✓	✓
		GraphAug [79]		✓		✓	✓
		JOAO [130]		✓		✓	✓
		MolCLE [116]		✓		✓	✓

Zhao, et al. Graph Data Augmentation for Graph Machine Learning: A Survey. 2022.

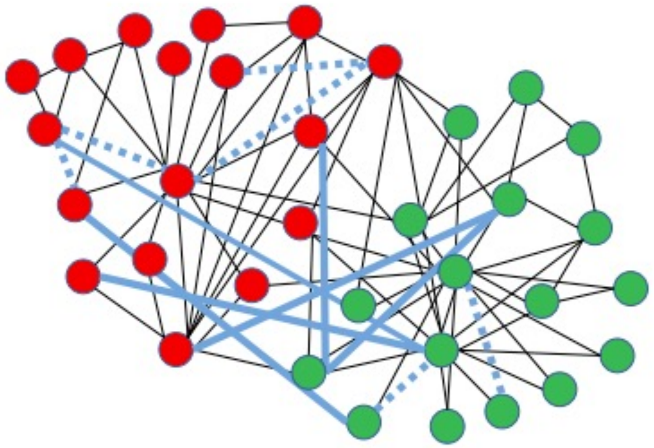
Limitations of Rule-based Approaches

Do not leverage task information and could hurt the downstream performance



(a) Original graph.

F1 Score: 92.4



(b) Random mod.

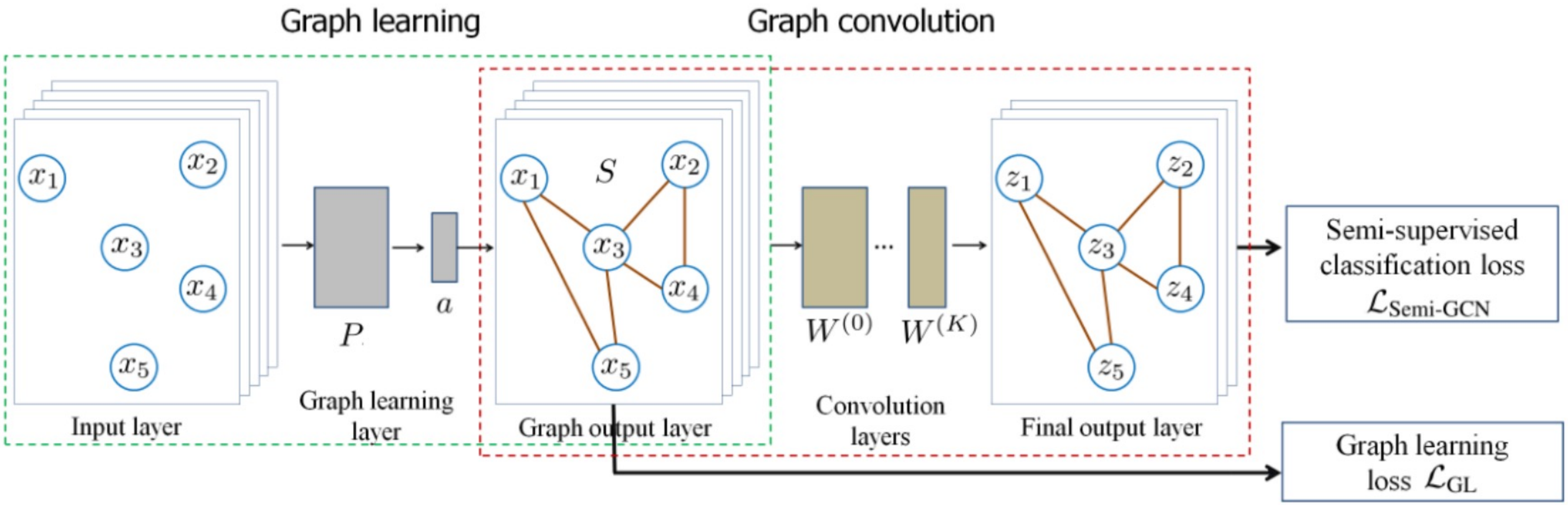
F1 Score: 91.0

Learned Graph Data Augmentation Approaches

- Graph Structure Learning
  - Augment data with good graph structures
- Adversarial Training
  - Augment data with adversarial examples
- Rationalization
  - Augment data by changing graph environment
- Automated Augmentation
  - Automatically combine different augmentations

Graph Structure Learning

Graph Learning + Graph Convolution



Jiang et al. Semi-supervised Learning with Graph Learning-Convolutional Networks. CVPR 2019.

GAug: Neural Edge Predictor

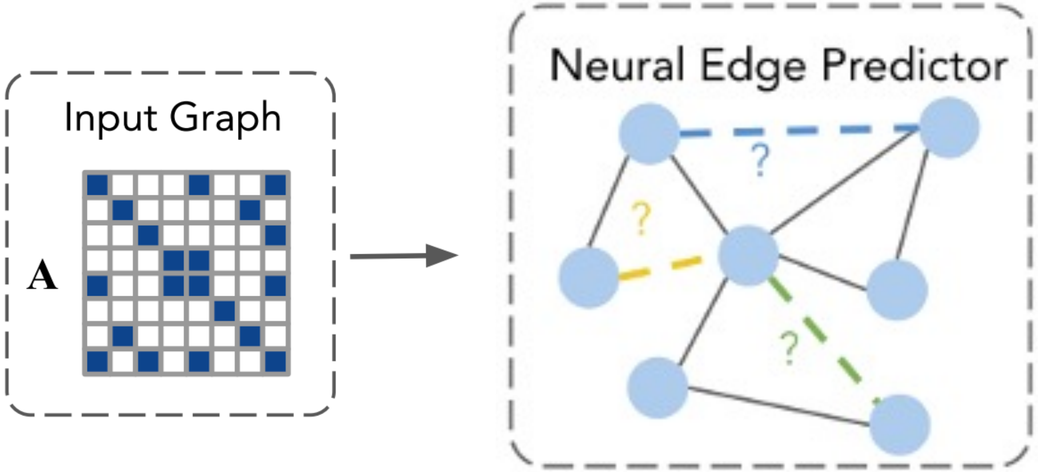
What are better graph structures?

- "Noisy" edges should be removed

  Inter-class edges

- "Missing" edges should be added

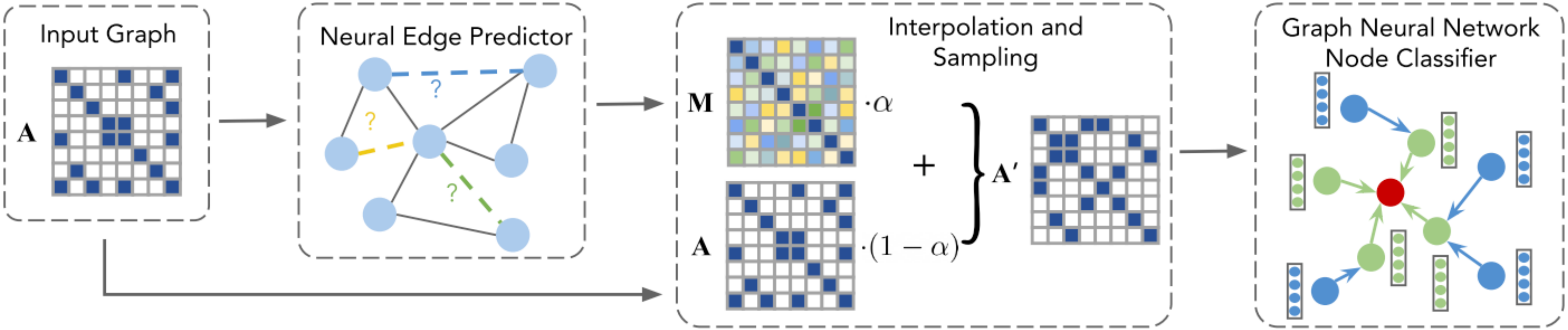
  Intra-class edges



$$\underbrace{\mathbf{M} = \sigma\left(\mathbf{Z}\mathbf{Z}^T\right)}_{M \text{ models node similarities}}, \text{ where } \mathbf{Z} = f_{GCL}^{(1)}\left(\mathbf{A}, f_{GCL}^{(0)}\left(\mathbf{A}, \mathbf{X}\right)\right)$$

M models node similarities

GAug: Interpolation and Sampling



$$\mathbf{P}_{ij} = \alpha \mathbf{M}_{ij} + (1-\alpha)\mathbf{A}_{ij}$$

Bernoulli sampling

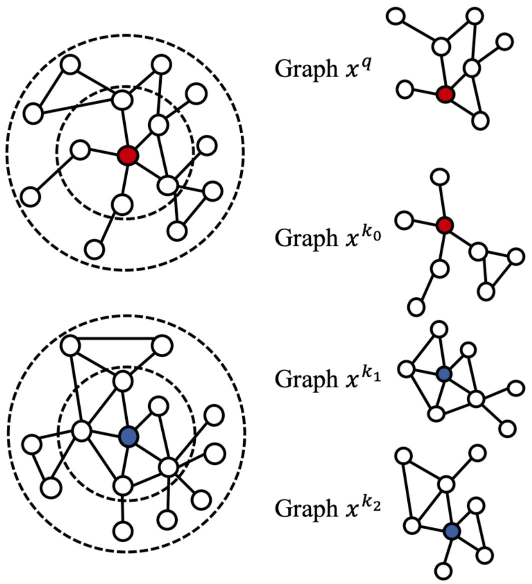
$$\mathbf{A}'_{ij}$$

Graph Self-supervised Learning

- Graph Self-Supervised Learning aims to learn generalizable node/edge/graph representations without using any human-annotated labels
  - Graph Generative Modeling
    - ➢ Learn generalizable representations by reconstructing the node features or/and graph structure

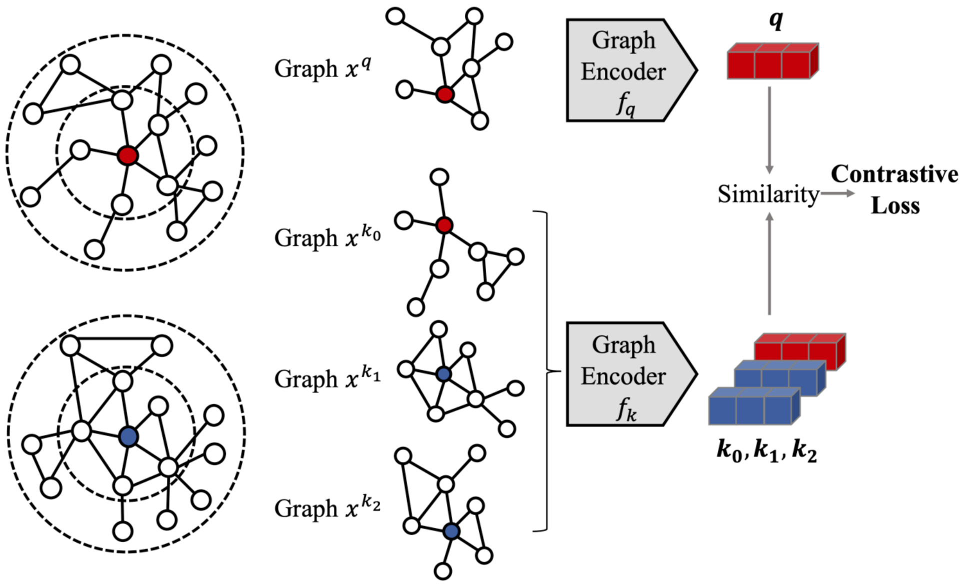
Graph Self-supervised Learning

- Graph Self-Supervised Learning aims to learn generalizable node/edge/graph representations without using any human-annotated labels
  - Graph Generative Modeling
    - ➢Learn generalizable representations by reconstructing the node features or/and graph structure
  - Graph Contrastive Learning (GCL)
    - ➢Create different views from the unlabeled input graph via data augmentation



Graph Self-supervised Learning

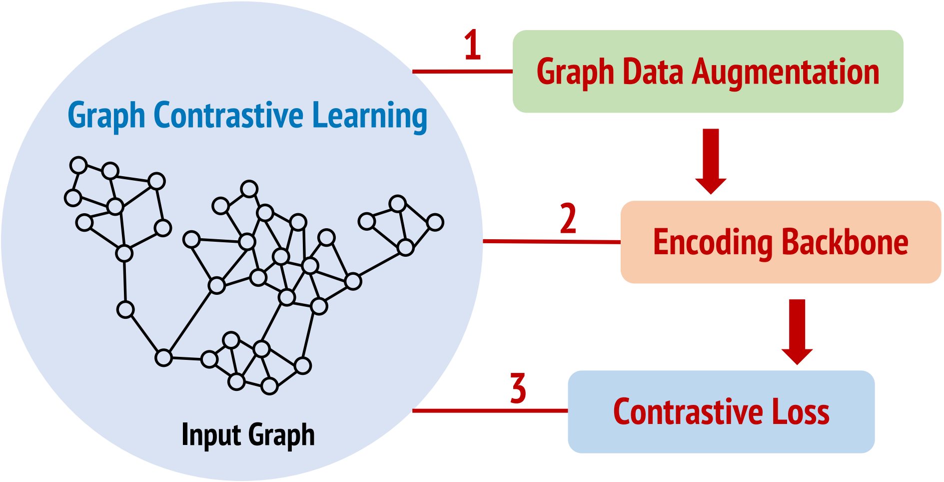
- Graph Self-Supervised Learning aims to learn generalizable node/edge/graph representations without using any human-annotated labels
  - Graph Generative Modeling
    - ➢Learn generalizable representations by reconstructing the node features or/and graph structure
  - Graph Contrastive Learning (GCL)
    - ➢Create different views from the unlabeled input graph via data augmentation
    - ➢Maximize the agreement between representations of different augmented views of the same instance



Qiu et al. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. KDD 2020

Typical Unsupervised Graph Contrastive Learning

- Graph Data Augmentation
  - ➢Create different views of each instance (e.g., node, subgraph)
  - ➢Arbitrary graph data augmentation (e.g., edge dropping, feature masking)
- Encoding Backbone
  - ➢Encode different augmented views
  - ➢Shallow GNNs (e.g., 2-layer GCN)
- Contrastive Loss
  - ➢Maximize the agreement between representations learned from different augmented views
  - ➢Instance-level contrastive learning



Outline

- Introduction and Background
- Topology Issues
- Imbalance Issues
- Short Break
- Bias Issue
- Limited Labeled Data Issues
- **Abnormal Graph Data Issues**
- Summary

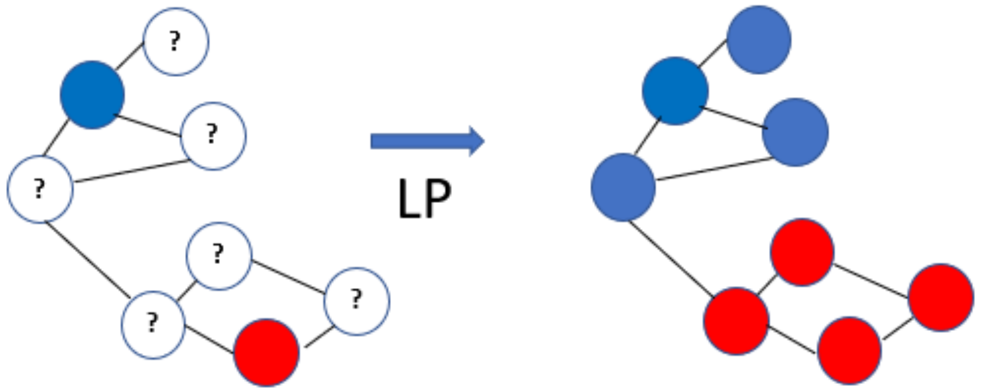
Abnormal Graph Data Issues

- Missing Data
- Adversarial Attacked Data

Missing Data

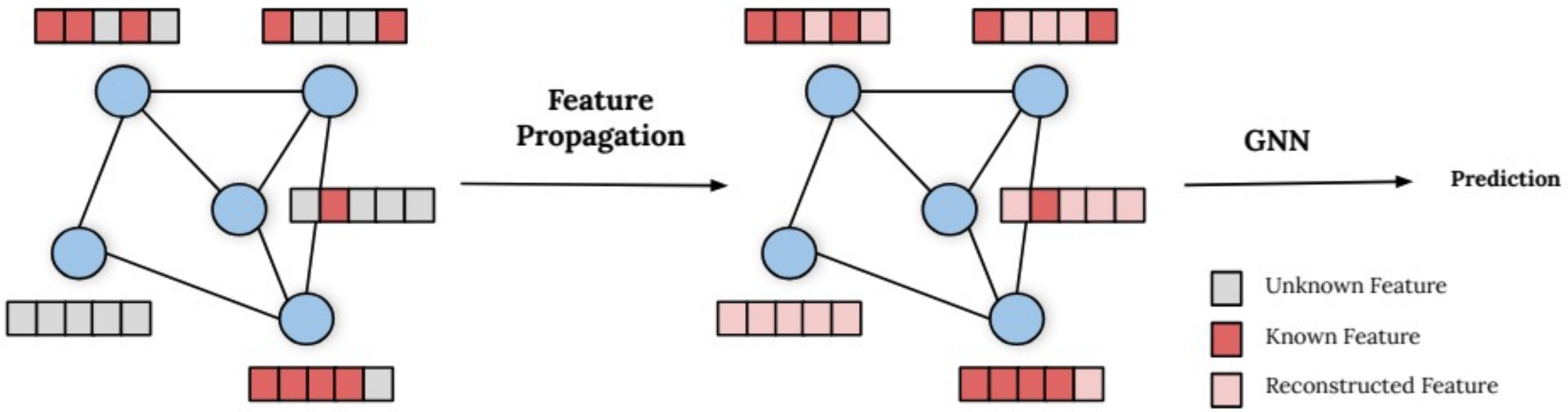
There are various solutions to deal with missing labels:

- Label propagation (LP)
- Self-supervised learning
- Unsupervised learning
- …



What if we have missing features?

- Feature propagation



Missing Data

What if we have missing features?

- Feature propagation

Algorithm 1 Feature Propagation

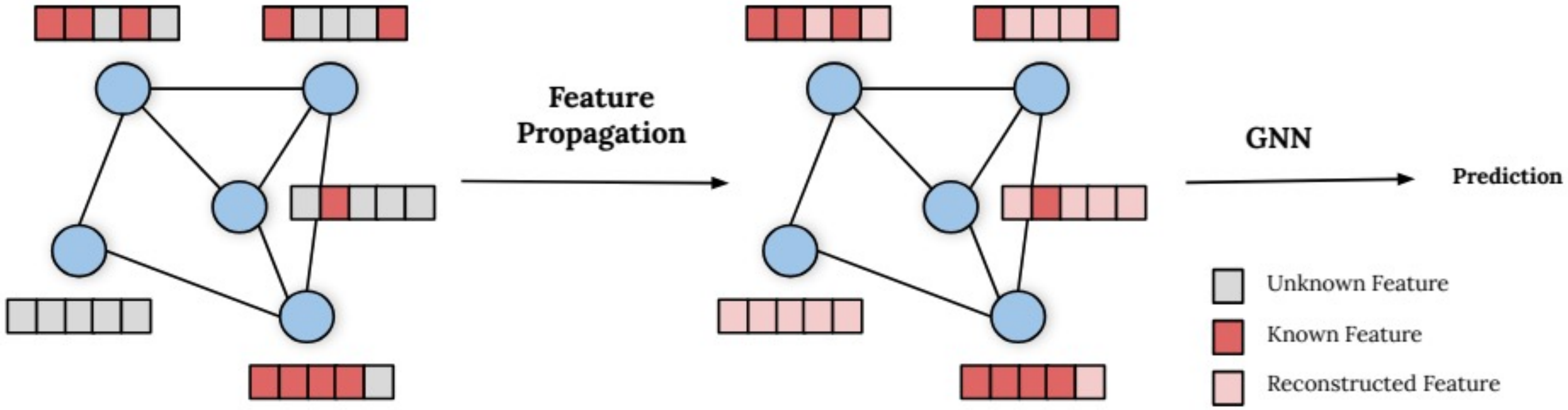
Input: feature vector $\mathbf{x}$, diffusion matrix $\tilde{\mathbf{A}}$

$\mathbf{y} \leftarrow \mathbf{x}$

while $\mathbf{x}$ has not converged **do**

$\quad \boxed{\begin{array}{ll} \mathbf{x} \leftarrow \tilde{\mathbf{A}}\mathbf{x} & \triangleright \text{ Propagate features} \\ \mathbf{x}_k \leftarrow \mathbf{y}_k & \triangleright \text{ Reset known features} \end{array}}$

end while



Missing Data

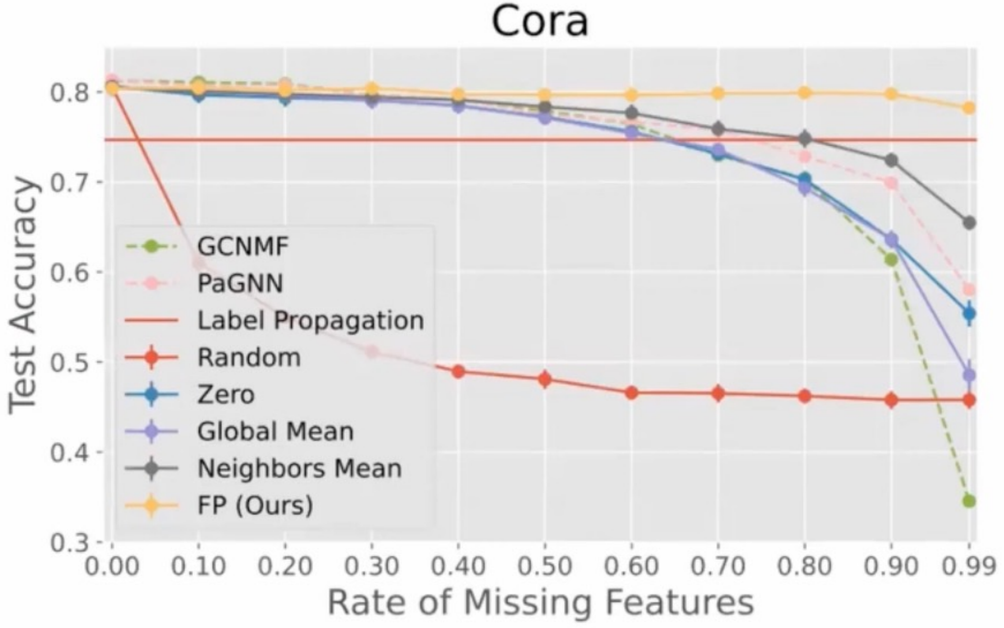
Comparison of Feature Propagation to Label Propagation

Feature Propagation:

- Propagates features (continuous)
- Prediction is made by a GNN on top of the propagated features
- Uses features, and a low % of them being present is enough for good performance

Label Propagation:

- Propagates class labels (discrete)
- Prediction is obtained directly from propagating class labels
- Feature-agnostic



Experiment Results

Across different levels of missing features,
Feature Propagation achieves the best performance

Missing Data

Beyond missing features on graphs, can we solve the general missing data problem?

Data Matrix with Missing Values

	F_1	F_2	F_3	F_4
O_1	0.3	0.5	NA	0.1
O_2	NA	NA	0.6	0.2
O_3	0.3	NA	NA	0.5

Labels

Y
y_1
y_2
?

Two ways of approaching missing data problems:

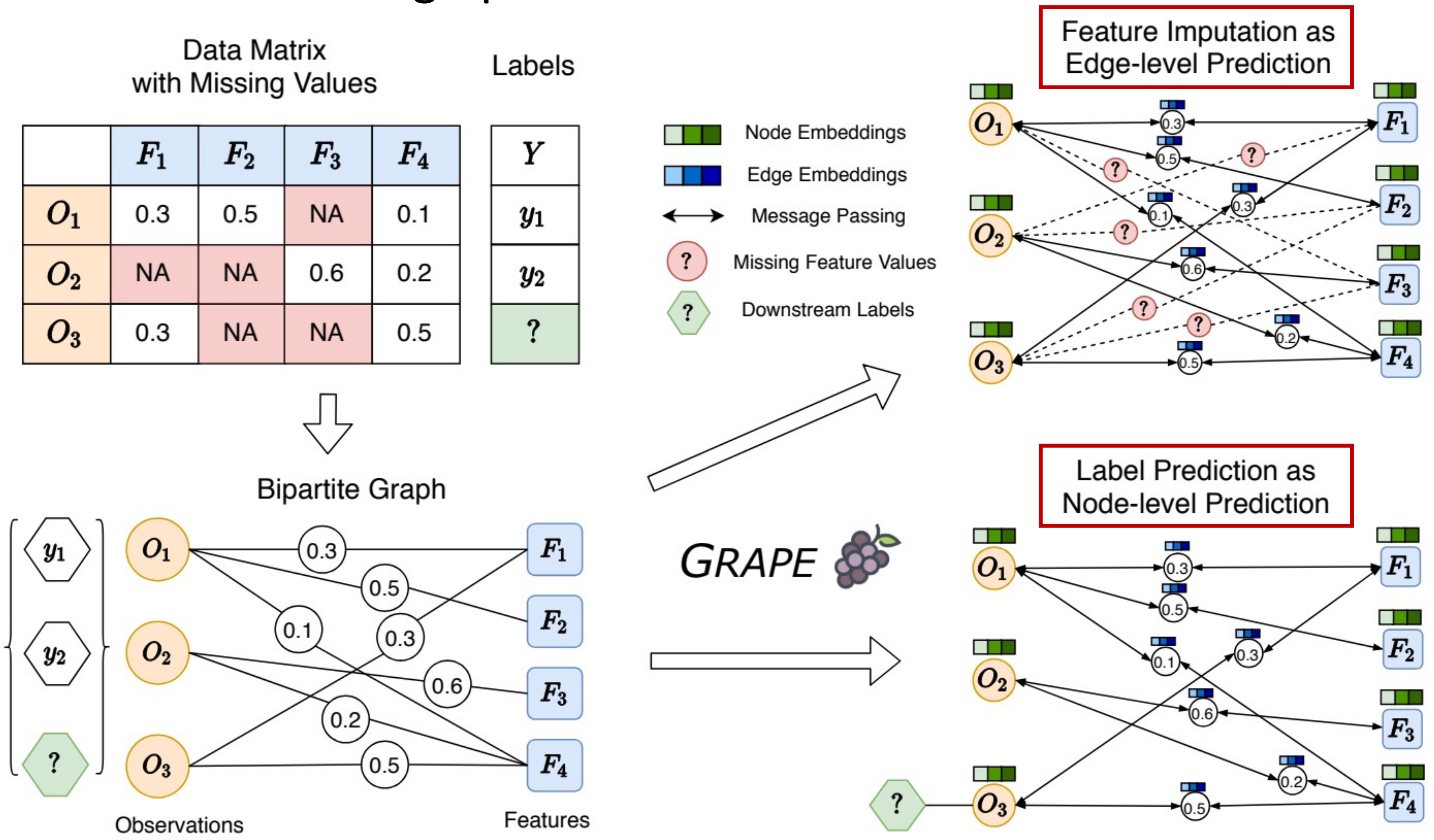
- **Feature imputation**: missing feature values are estimated based on observed values
- **Label prediction**: downstream labels are learned directly from incomplete data

Issues:

- Existing methods fail to make full use of feature values from other observations
- Existing methods tend to make biased assumptions about the missing values by initializing them with special default values

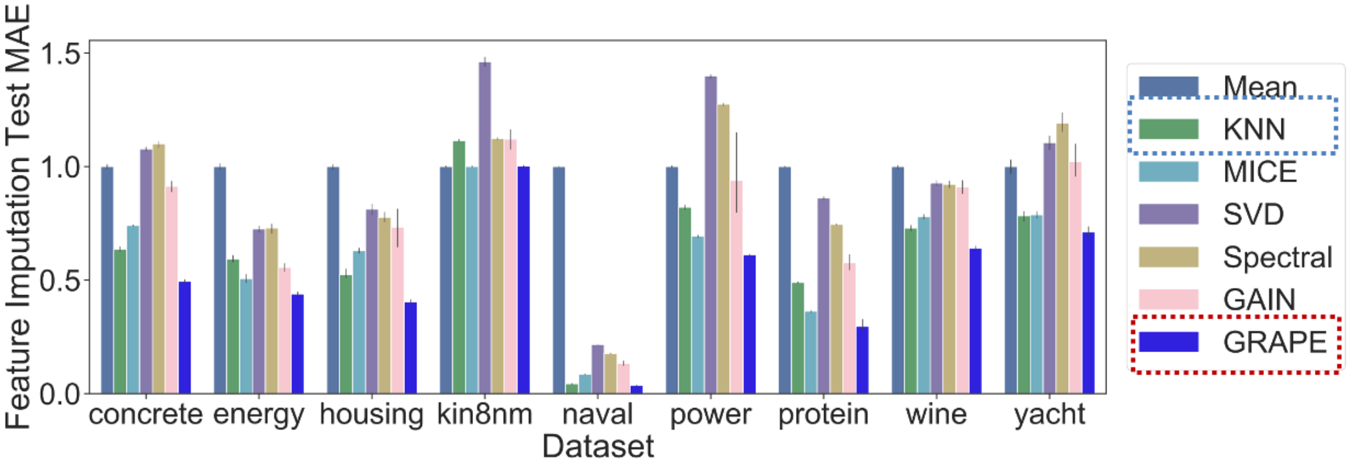
Missing Data

GRAPE: reformulate the tasks as graph tasks

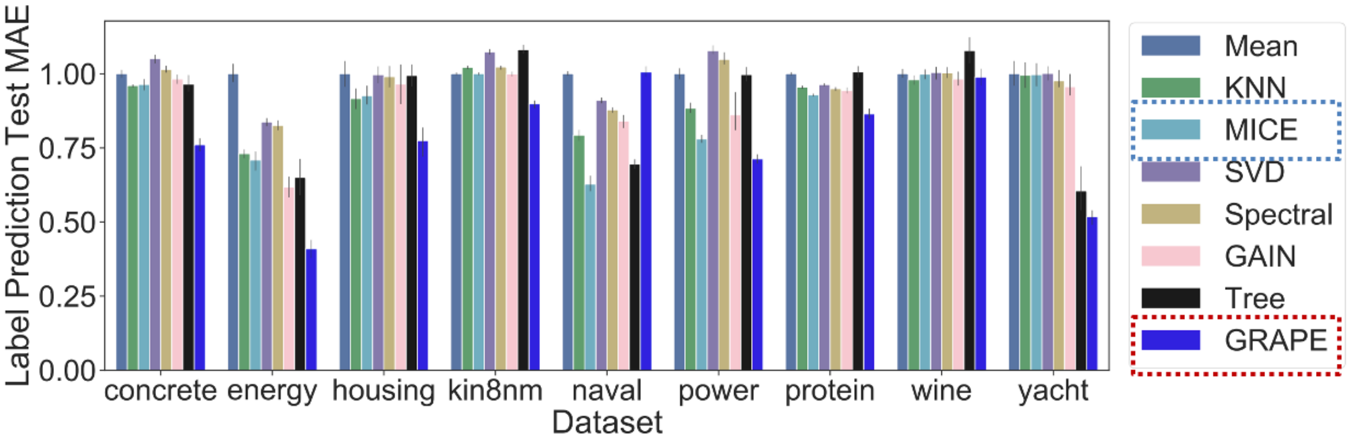


Missing Data

GRAPE yields 20% lower mean absolute error for feature imputation, and 10% lower MAE for label prediction



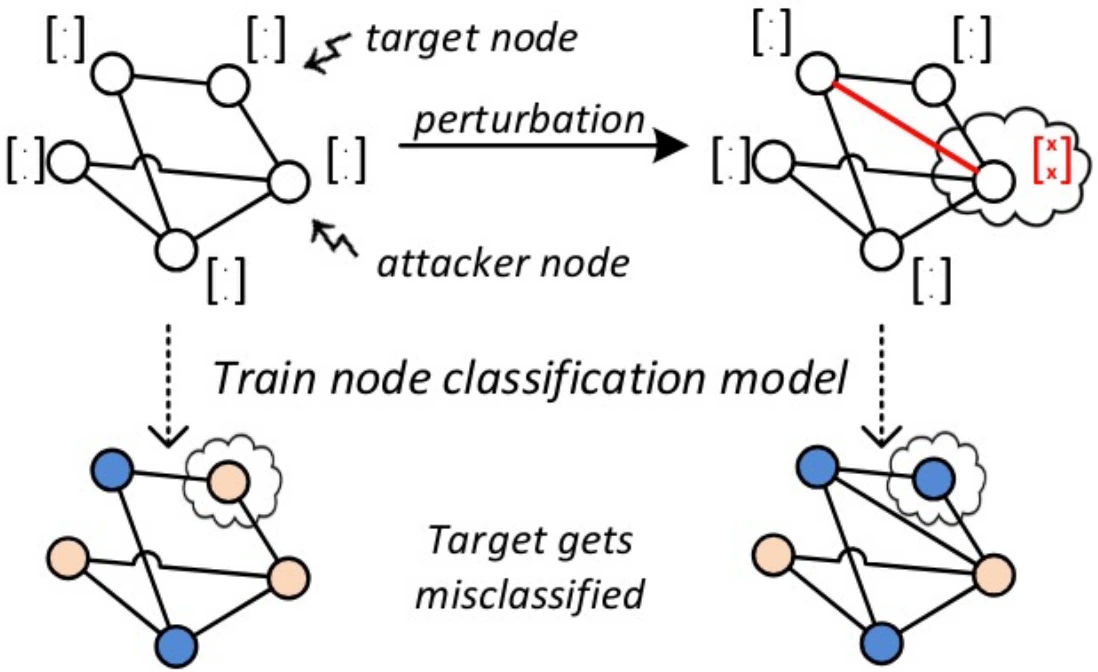
Feature imputation: 20% lower MAE than best baseline (KNN)

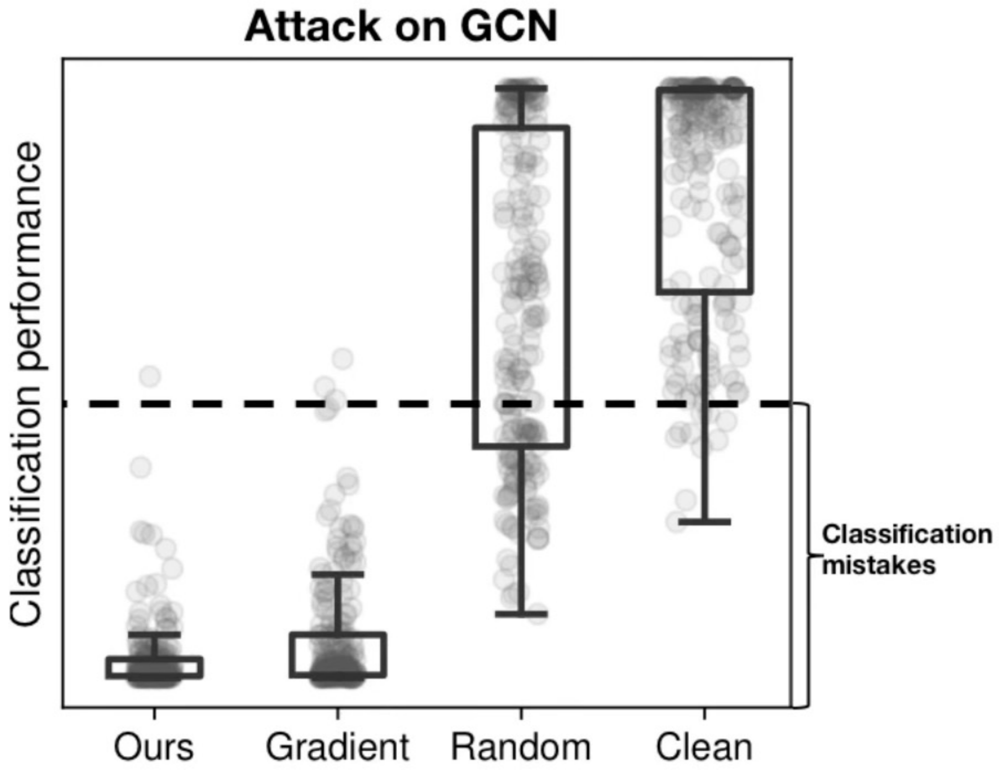


Label prediction: 10% lower MAE than best baseline (MICE)

Adversarial Attacked Data

Observation: Small perturbations of the graph structure and node features lead to misclassification of the target





Adversarial Attacked Data

Can we leverage small data perturbations to improve performance?

Yes, adversarial training

Adversarial training is the process of crafting adversarial data points, and then injecting them into training data

$$\min_{\boldsymbol{\theta}} \quad E_{(x,y)\sim\mathcal{D}}\left[\max_{||\boldsymbol{\delta}||_p\leq\epsilon} L\left(f_{\boldsymbol{\theta}}(x+\boldsymbol{\delta}),y\right)\right]$$

Find the optimal perturbation sample to achieve maximum loss

Find the optimal model parameters to resist the attack of perturbation sample

D: distribution

$||.||_p$: l_p-norm distance metric

ϵ: perturbation budget

Kong, Kezhi, et al. "Robust optimization as data augmentation for large-scale graphs." CVPR 2022

Adversarial Attacked Data

Can we leverage small data perturbations to improve performance?

Yes, adversarial training

Node Classification

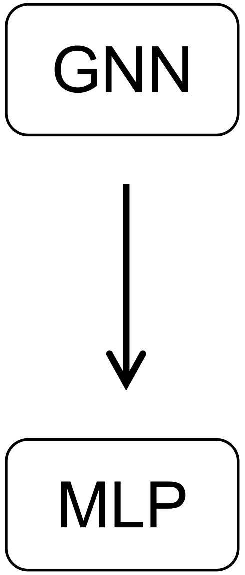
Backbone	ogbn-products Test Acc	ogbn-proteins Test ROC-AUC	ogbn-arxiv Test Acc
GCN	-	**72.51**±0.35	71.74±0.29
+FLAG	-	71.71±0.50	**72.04**±0.20
GraphSAGE	78.70±0.36	**77.68** ±0.20	71.49±0.27
+FLAG	**79.36**±0.57	76.57±0.75	**72.19**±0.21
GAT	79.45±0.59	-	73.65±0.11
+FLAG	**81.76**±0.45	-	**73.71**±0.13
DeeperGCN	80.98±0.20	85.80±0.17	71.92±0.16
+FLAG	**81.93**±0.31	**85.96**±0.27	**72.14**±0.19

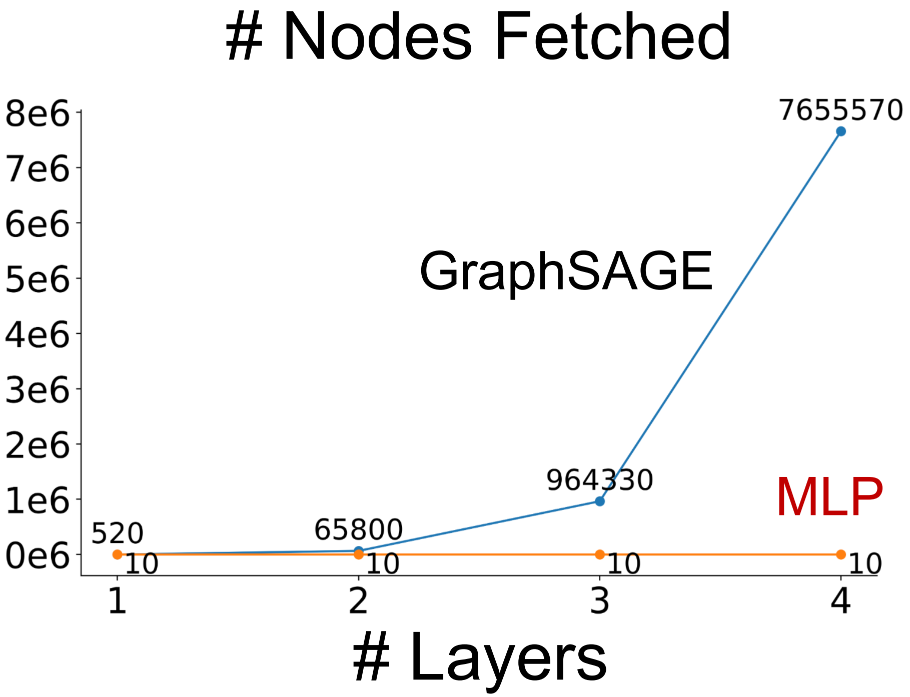
Kong, Kezhi, et al. "Robust optimization as data augmentation for large-scale graphs." CVPR 2022

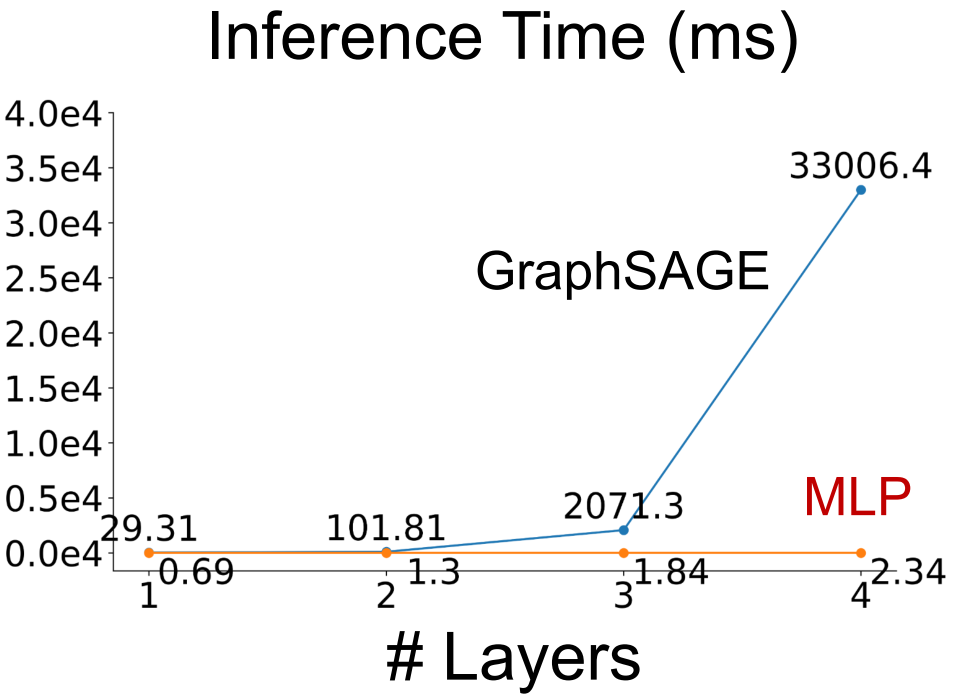
Adversarial Attacked Data

Can we leverage small data perturbations to improve robustness?
Yes, adversarial training

A use case: training an MLP on graphs
Reason: to avoid the computation-intensive message passing mechanism





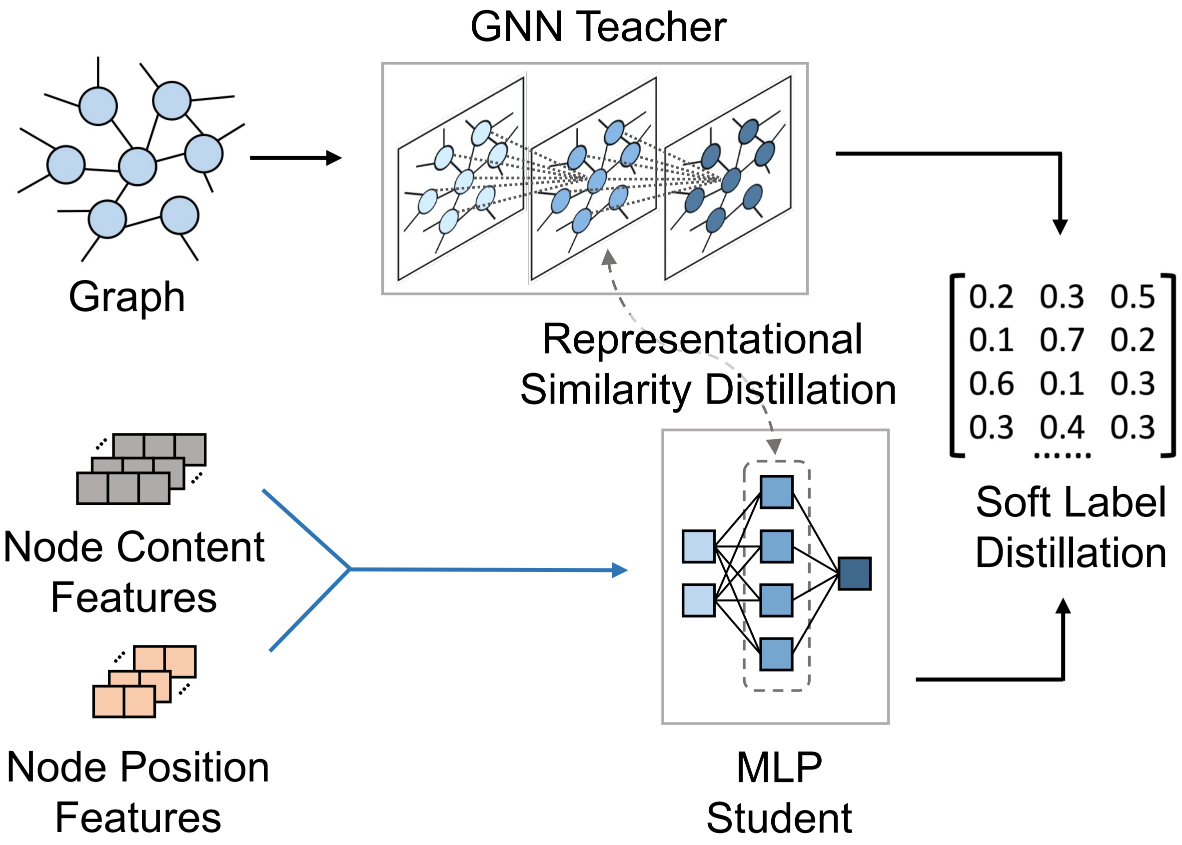


Adversarial Attacked Data

Can we leverage small data perturbations to improve robustness?

Yes, adversarial training

A use case: training an MLP on graphs

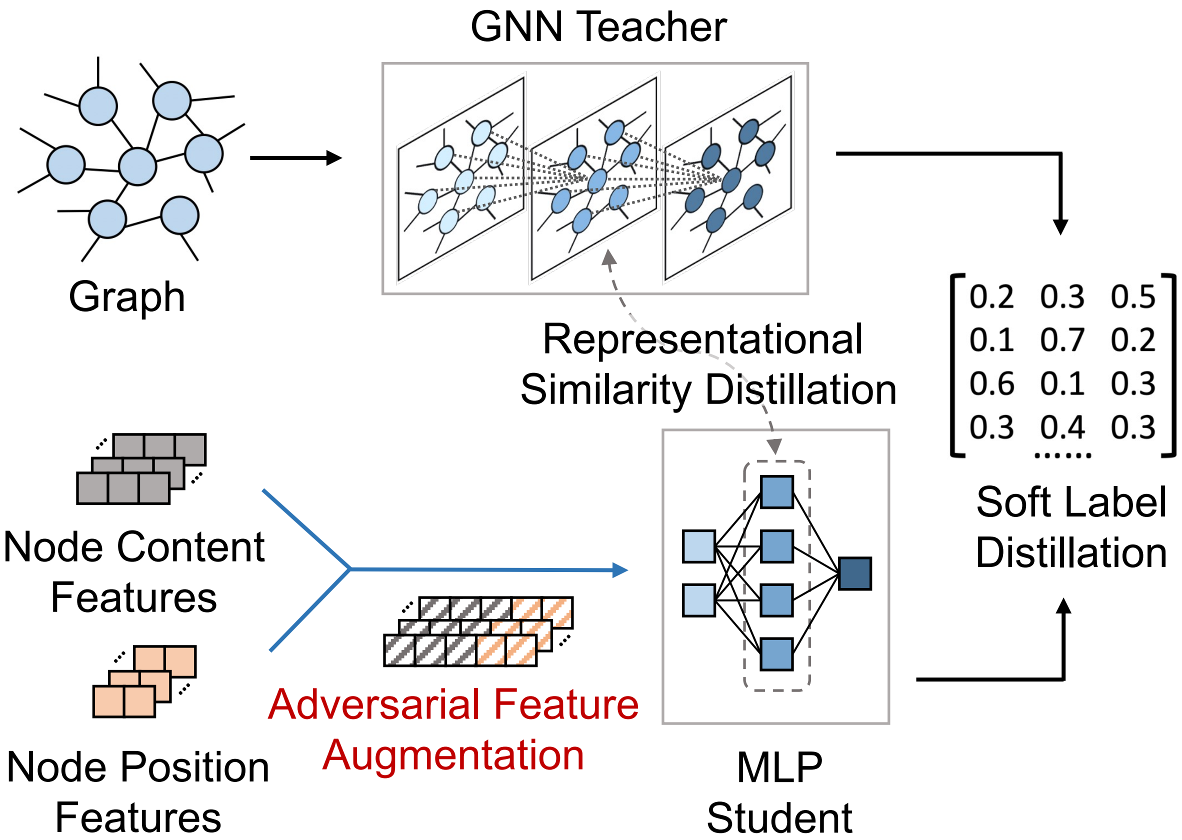


The problem of training an MLP on graphs: sensitive to features

Adversarial Attacked Data

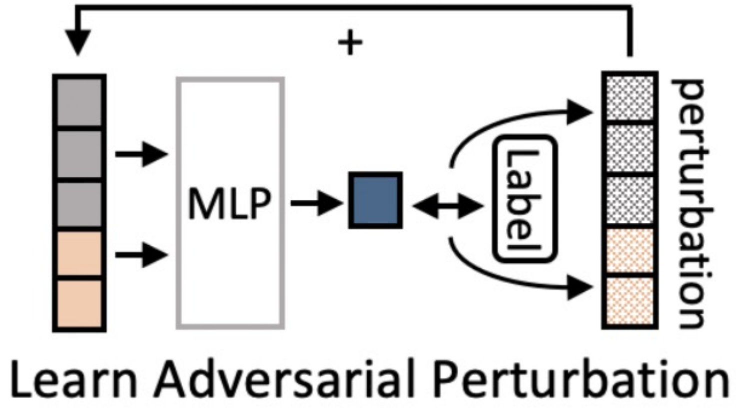
Can we leverage small data perturbations to improve robustness?
Yes, adversarial training

A use case: training an MLP on graphs



The problem of training an MLP on graphs: sensitive to features

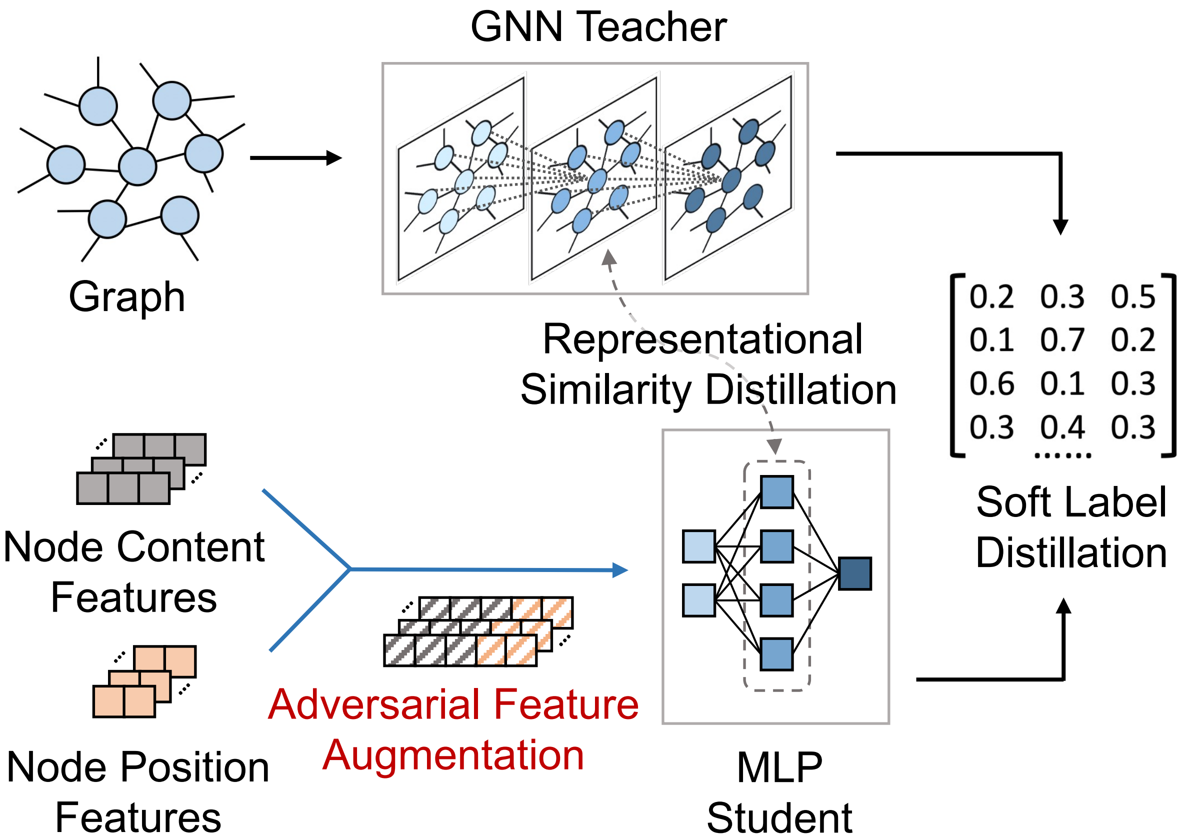
Overcome this problem with adversarial training



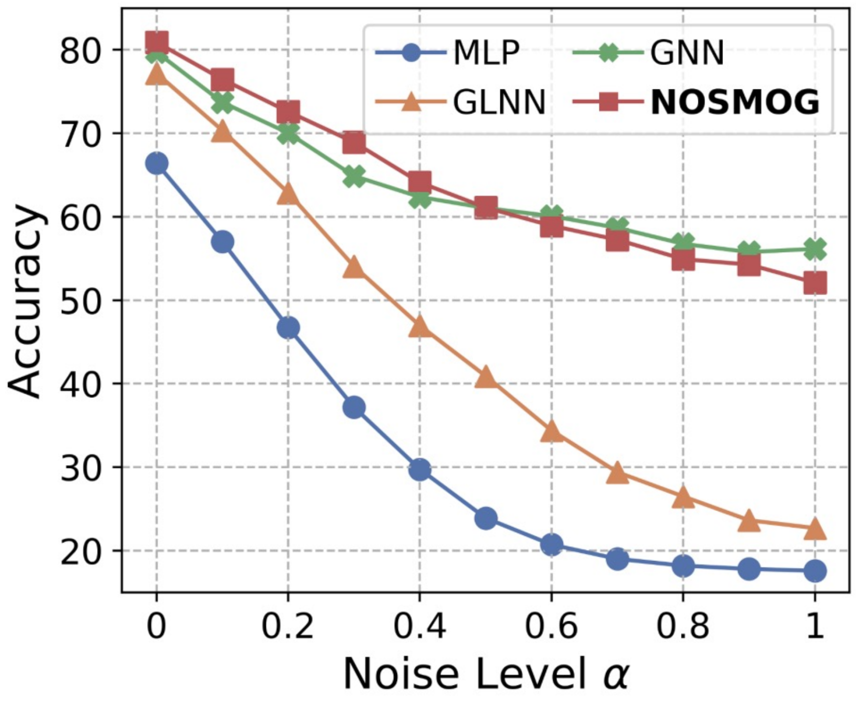
Adversarial Attacked Data

Can we leverage small data perturbations to improve robustness?
Yes, adversarial training

A use case: training an MLP on graphs



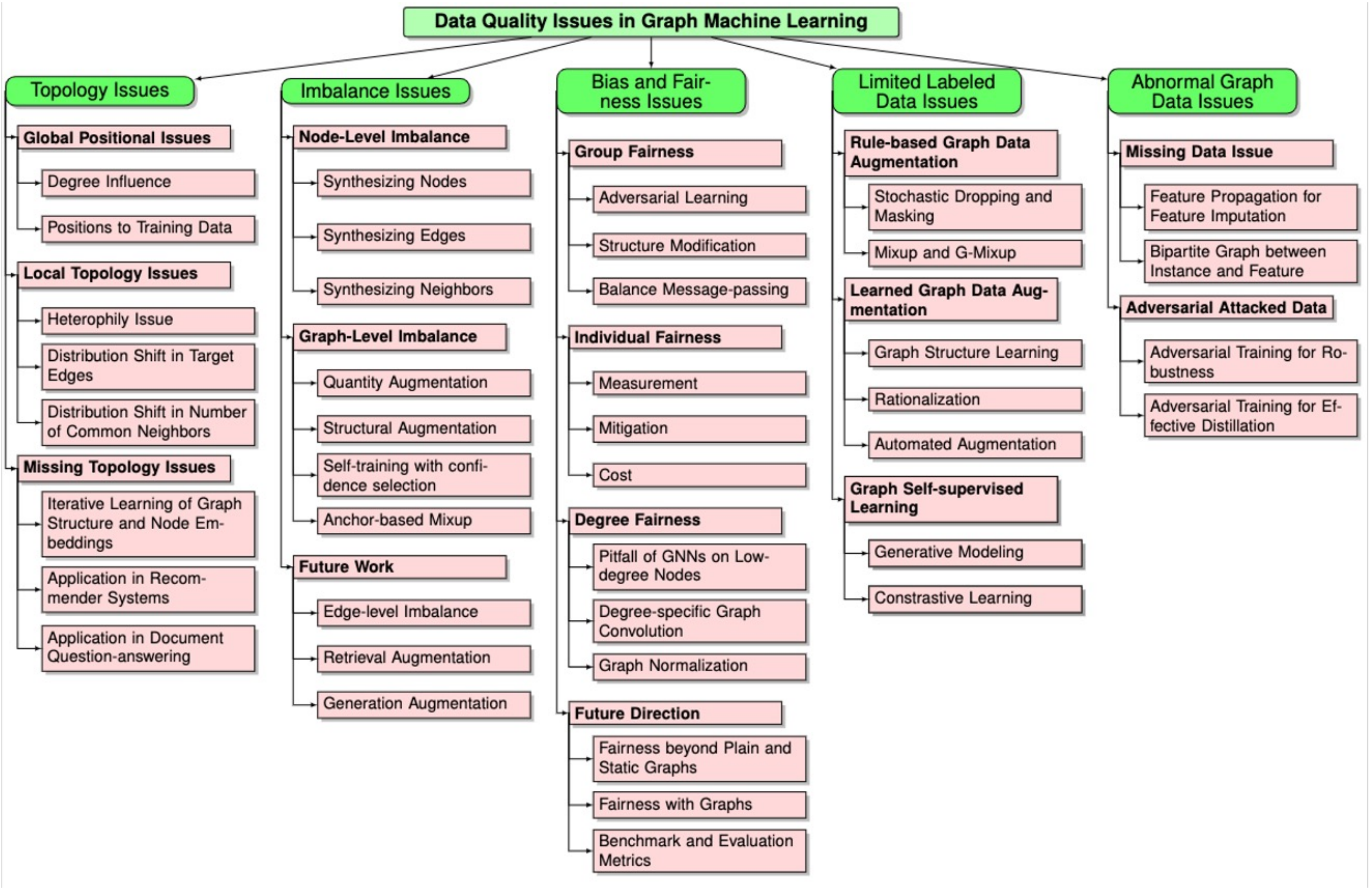
NOSMOG is as robust as GNNs



Summary

- Introduction and Background
- Topology Issues
- Imbalance Issues
- Short Break
- Bias Issue
- Limited Labeled Data Issues
- Abnormal Graph Data Issues
- **Summary**

Summary



Data Quality Issues in Graph Machine Learning
Topology Issues
Global Positional Issues
Degree Influence
Positions to Training Data
Local Topology Issues
Heterophily Issue
Distribution Shift in Target Edges
Distribution Shift in Number of Common Neighbors
Missing Topology Issues
Iterative Learning of Graph Structure and Node Embeddings
Application in Recommender Systems
Application in Document Question-answering
Imbalance Issues
Node-Level Imbalance
Synthesizing Nodes
Synthesizing Edges
Synthesizing Neighbors
Graph-Level Imbalance
Quantity Augmentation
Structural Augmentation
Self-training with confidence selection
Anchor-based Mixup
Future Work
Edge-level Imbalance
Retrieval Augmentation
Generation Augmentation
Bias and Fairness Issues
Group Fairness
Adversarial Learning
Structure Modification
Balance Message-passing
Individual Fairness
Measurement
Mitigation
Cost
Degree Fairness
Pitfall of GNNs on Low-degree Nodes
Degree-specific Graph Convolution
Graph Normalization
Future Direction
Fairness beyond Plain and Static Graphs
Fairness with Graphs
Benchmark and Evaluation Metrics
Limited Labeled Data Issues
Rule-based Graph Data Augmentation
Stochastic Dropping and Masking
Mixup and G-Mixup
Learned Graph Data Augmentation
Graph Structure Learning
Rationalization
Automated Augmentation
Graph Self-supervised Learning
Generative Modeling
Constrastive Learning
Abnormal Graph Data Issues
Missing Data Issue
Feature Propagation for Feature Imputation
Bipartite Graph between Instance and Feature
Adversarial Attacked Data
Adversarial Training for Robustness
Adversarial Training for Effective Distillation

Future Directions

Topology Issue

Imbalance Data Issue

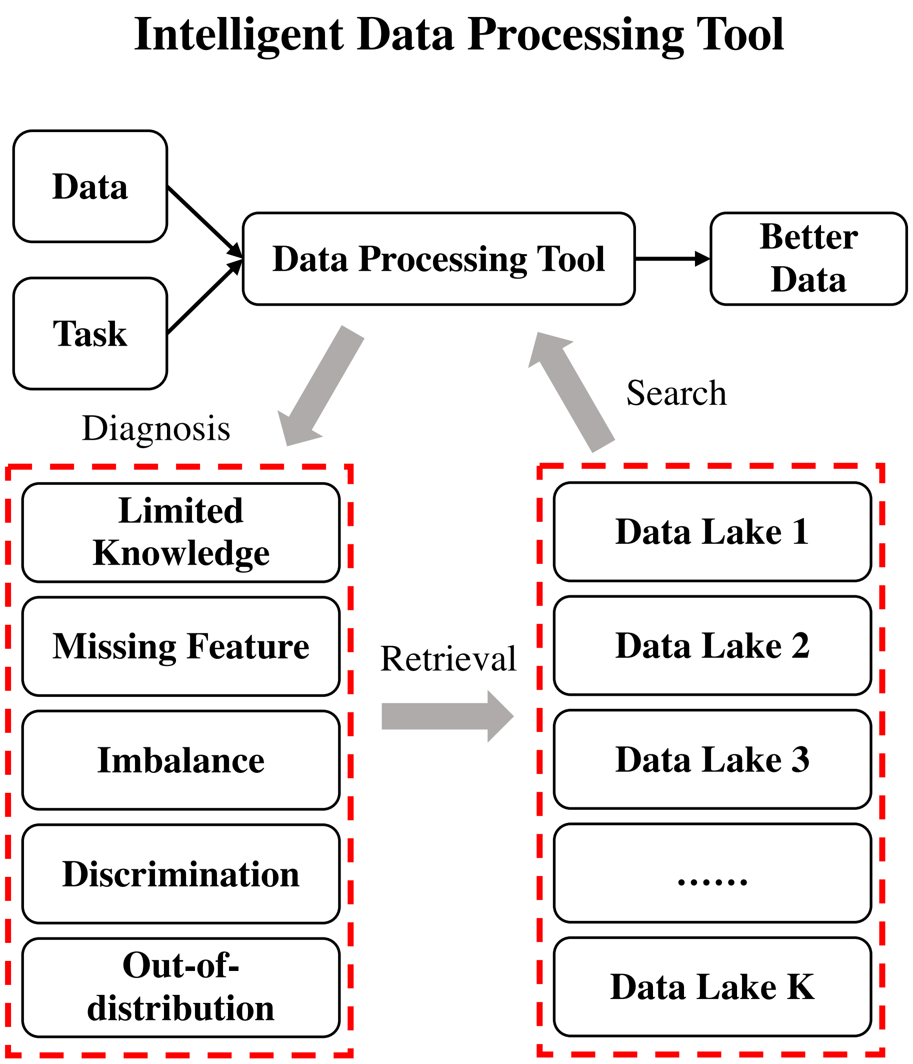
Bias and Fairness Issue

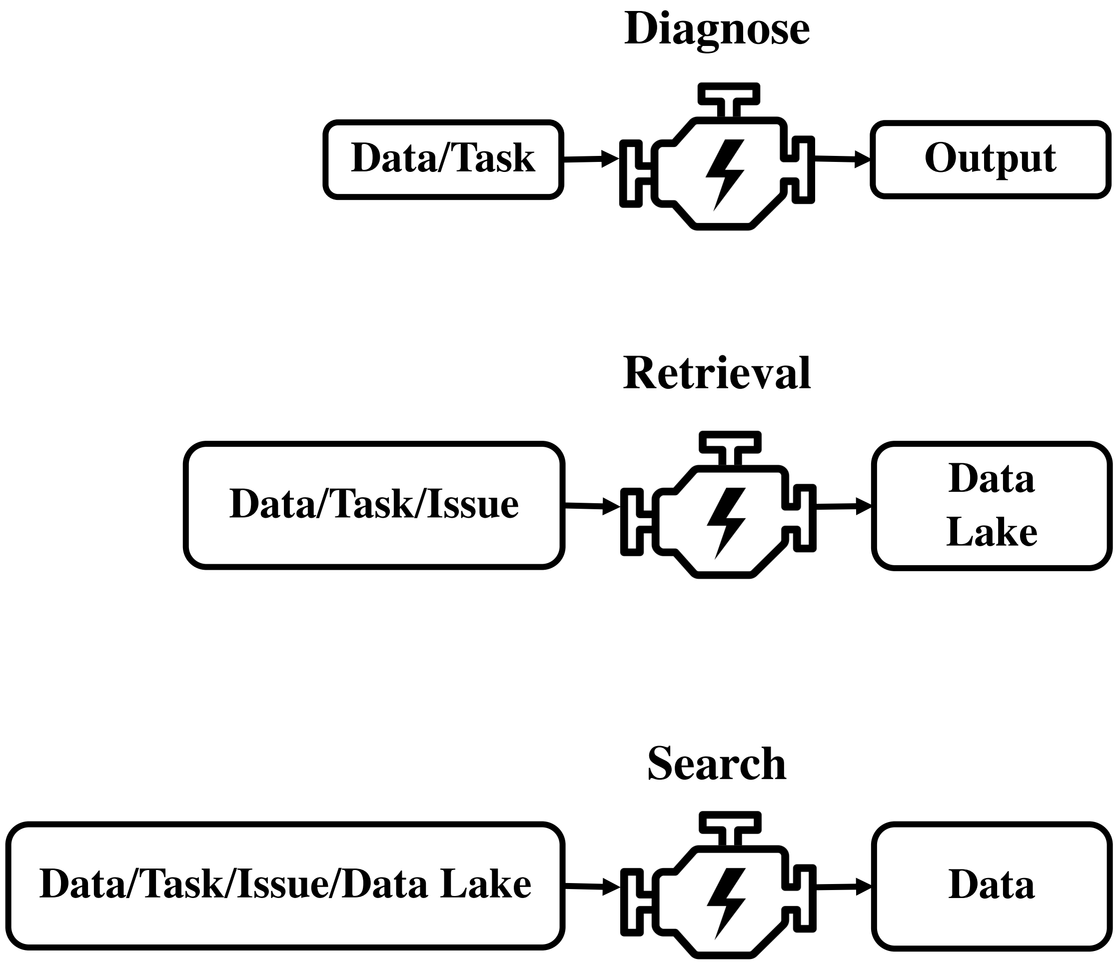
Limited Labeled Data Issue

Abnormal Graph Data Issue

Existing Data Processing is very time-consuming and labor extensive!

Summary


Intelligent Data Processing Tool
Data
Task
Data Processing Tool
Better Data
Diagnosis
Search
Retrieval
Limited Knowledge
Missing Feature
Imbalance
Discrimination
Out-of-distribution
Data Lake 1
Data Lake 2
Data Lake 3
……
Data Lake K


Diagnose
Data/Task
Output
Retrieval
Data/Task/Issue
Data Lake
Search
Data/Task/Issue/Data Lake
Data