## Lab Seminar

# Fairness-aware Graph Learning (2)

HTET ARKAR

Undergraduate

School of Computer Science and Engineering

Chung-Ang University

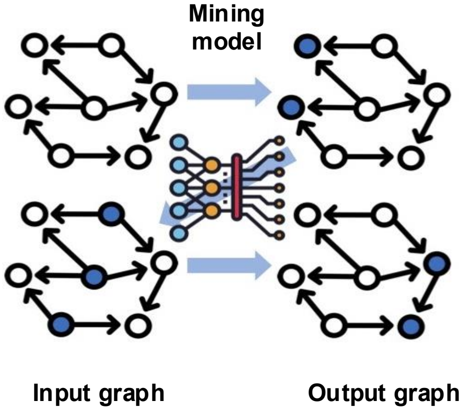
# Outline

- **Introduction**
- **Fairness-aware Graph Learning**
- **Problems Definitions**
- **Methodology**
  - ❑ FMP (AAAI-24)
  - ❑ DAB-GNN (AAAI-25)
- **Conclusion**

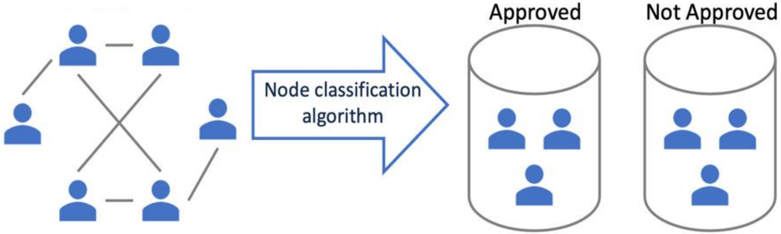
# Introduction

## ❖ Graph Learning

- ❑ A process in which the mining techniques are used to analyze data represented as graphs to discover **patterns, relationships, and trends**
- ❑ Graph-structured data is pervasive in diverse real-world applications
- ❑ To gain a deeper understanding of such data, graph learning methods (e.g. GNNs) are adopted



Loan Approval System

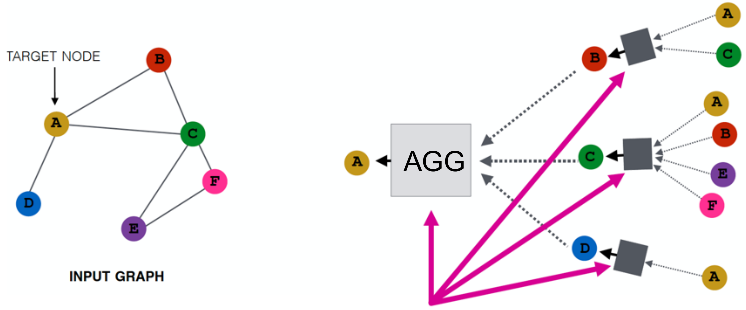


# Graph Learning Methods

## ❖ Graph Neural Networks

- ❑ **Aggregate information** from neighboring nodes
- ❑ **Update node embeddings** by stacking $L$ layers
- ❑ Final node embeddings can be used for downstream tasks
  - ➢ Node classification and link prediction
  - ➢ Enhance model accuracy

$$
\begin{aligned}
\mathbf{h}_u^{k+1} &= Update^k\left(\mathbf{h}_u^k, Aggregate(\mathbf{h}_v^k \mid \forall v \in N(u))\right) \\
&= Update^k\left(\mathbf{h}_u^k, \mathbf{m}_{N(u)}^k\right)
\end{aligned}
$$



Neural networks

# Fairness-aware Graph Learning

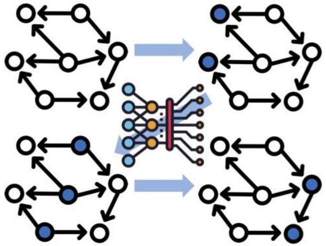
## ❖ Fairness and Bias

Stanford University
Human-Centered
Artificial Intelligence

> “Creating algorithms that avoid bias or discrimination, and considering the diverse needs and circumstances of all stakeholders, thereby aligning with broader societal standards of equity.”

## ❖ Why GNNs give unfair result?

- ❑ Societal Bias in the data
- ❑ Graph Typology
- ❑ Message-passing Mechanism of GNNs

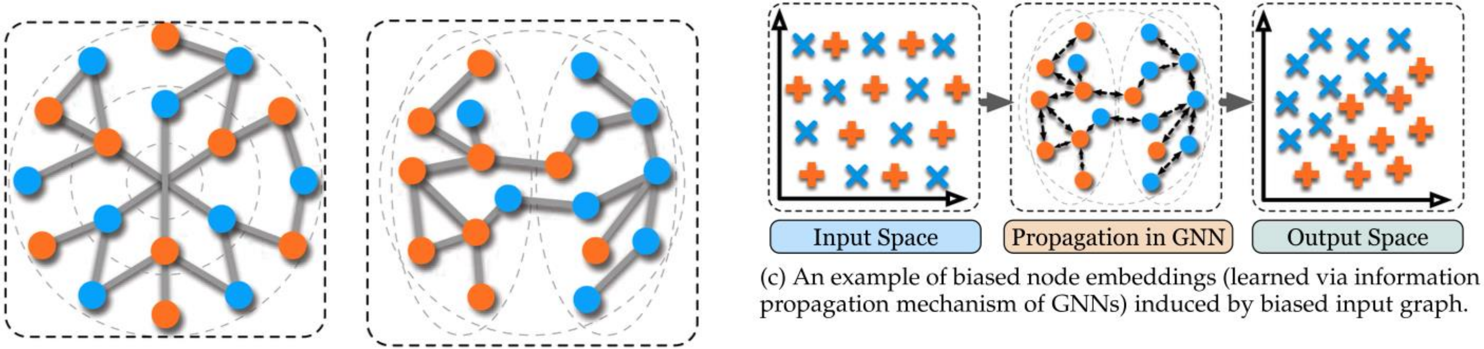


Pipeline of Graph Mining

# Problem Definition

## ❖ Bias in Graph-structured Data and GNNs

- ❑ (1) The **raw features** of nodes could be statistically **correlated to the sensitive attribute**
  - ➢ Lead to sensitive information leakage in encoded representations
- ❑ (2) **Homophily effects**: nodes with the same sensitive attribute tend to link with each other
  - ➢ Make the node representations in the same sensitive group more similar during message passing



(a) Unbiased graph topology

(b) Biased graph topology

(c) An example of biased node embeddings (learned via information propagation mechanism of GNNs) induced by biased input graph.

Fig. 1. Examples of (a) unbiased graph topology, (b) biased graph topology, and (c) how information propagation mechanism induces bias in GNNs. Nodes in two different demographic subgroups are in orange and blue.

# Bias in Graph-structured Data and GNNs

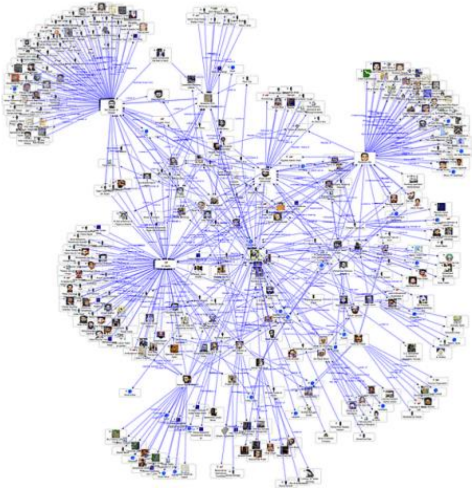
## ❖ Social Networks

- ❑ Young people tend to build friendship with people with similar age on the social network
- ❑ The message passing in GNNs will aggregate the neighbor features
- ❑ Thus, GNNs learn similar representations for nodes of similar sensitive information while different representations for modes of different sensitive features

Leading to severe bias in decision making

↓

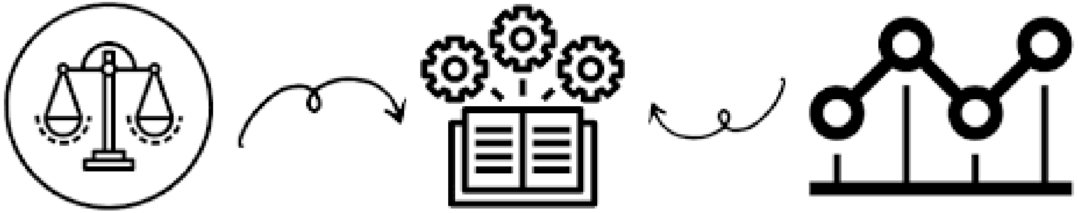
The predictions are highly correlated with the sensitive attributes of the nodes



# Fairness-aware Graph Learning

- **Crucial to ensure that GNNs do not exhibit discrimination towards users**
  - Develop Fair GNN to achieve various types of of fairness on different tasks

- **Challenges: How to tackle unfairness issues in graph mining algorithms**
  - How to formulate proper fairness notion
    - As the criteria to determine the existence of unfairness (i.e. Bias)
  - How to prevent the graph mining algorithms
    - From inheriting the bias exhibited in the input relational information



# Chasing Fairness in Graphs: A GNN Architecture Perspective

Zhimeng Jiang[1], Xiaotian Han[1], Chao Fan[2], Zirui Liu[3], Na Zou[4], Ali Mostafavi[1], Xia Hu[3]

[1]Texas A&M University

[2]Clemson University

[3]Rice University

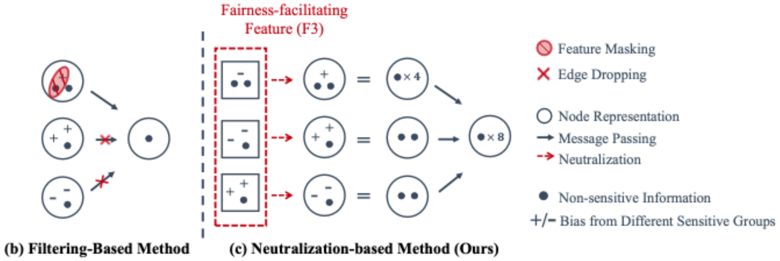
[4]University of Houston

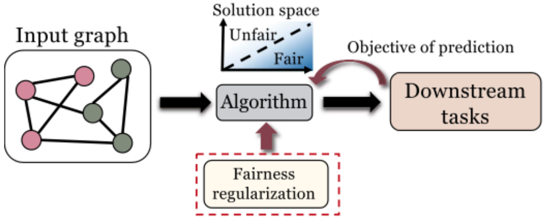
AAAI-24

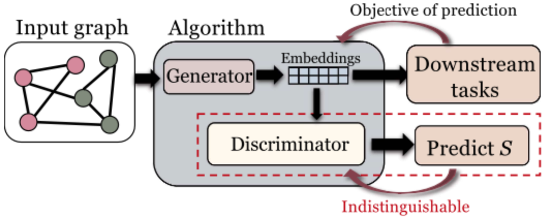
# Previous Works

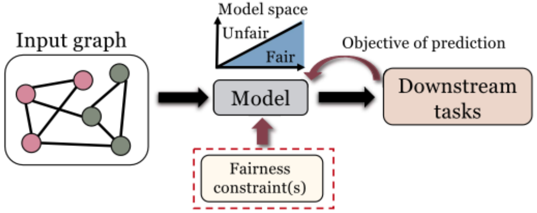
## ❖ Achieving Fair Prediction in Graphs

- ❑ Graph pre-processing
  - ➢ e.g, node feature masking, and topology rewiring
- ❑ Fair training strategies
  - ➢ E.g., Regularization, Adversarial debiasing, or Contrastive learning


Fairness-facilitating Feature (F3)
(b) Filtering-Based Method
(c) Neutralization-based Method (Ours)
×4
×8
Feature Masking
Edge Dropping
Node Representation
Message Passing
Neutralization
Non-sensitive Information
+/− Bias from Different Sensitive Groups







- ❑ **GNNs architecture perspective to improve fairness in graphs is less explored**
- ❑ **GNN aggregation amplifies bias compared to multilayer perception (MLP)**

# Proposed Method

## ❖ A new fairness-aware GNN architecture called Fair Message Passing (FMP)

- ❑ Aim to improve fairness directly within the model’s architecture
- ❑ **Not** just modifying the data or training process
- ❑ It follows a two-step approach:
  - ➢ **Aggregation**: Standard neighbor-based information gathering
  - ➢ **Bias Mitigation**: Adjust node representations to reduce disparities between demographic groups

## ❖ The Optimization Problem

- ❑ Pursue smoothness and fair node representation simultaneously

$$\min_{\mathbf{F}} \quad \underbrace{\frac{\lambda_s}{2} tr(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) + \frac{1}{2} ||\mathbf{F} - \mathbf{X}_{trans}||_F^2}_{h_s(\mathbf{F}) \text{ smoothness}} + \underbrace{\lambda_f ||\mathbf{\Delta}_s SF(\mathbf{F})||_1}_{h_f\left(\mathbf{\Delta}_s SF(\mathbf{F})\right) \text{ fairness}} .$$

Proposed Method (1)

# Fair Message Passing (FMP)

## ❖ Architecture

- ❑ Purpose: Aggregating useful information from neighbors while debiasing representation bias
- ❑ Be integrated into Fair GNNs at three stage
  - ➢ Transformation ➠ Node feature
  - ➢ Aggregation ➠ Graph typology
  - ➢ Debiasing step ➠ Sensitive attribute

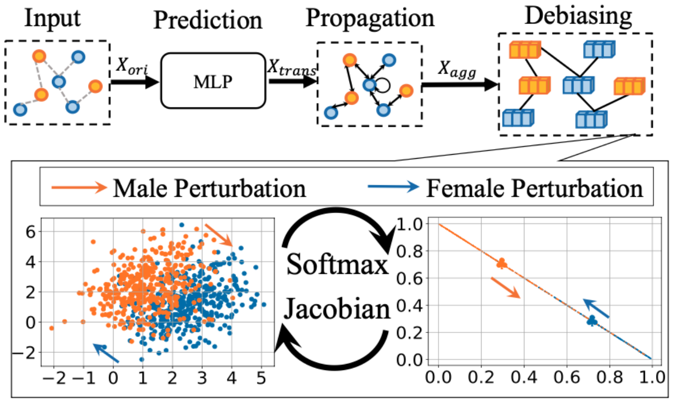
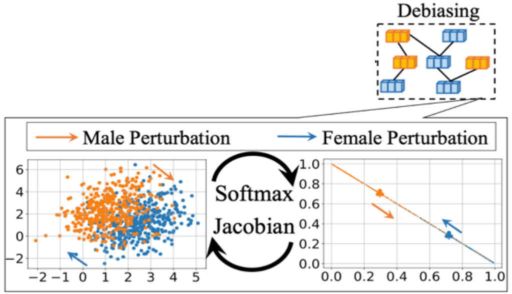


Figure 1: The model pipeline consists of three steps: MLP (feature transformation), propagation with skip connection, and debiasing in probability space.

# Discussion on FMP

## ❖ (1) Interpretation



- ❑ Gradient of fairness objective over node features
- ❑ Able to be interpreted as three steps
  - ➢ (1) Softmax transformation
    - ▪ First map the node representation into probability space via softmax transformation
  - ➢ (2) Perturbation in probability space
    - ▪ Calculate the gradient of fairness objective in probability space
    - ▪ (Perturbation actually poses low-rank debiasing in probability space)
    - ▪ In where the nodes with different sensitive attributes embrace opposite perturbations
  - ➢ (3) Debiasing in representation space
    - ▪ The perturbation in probability space will be transformed into representation space via Jacobian transformation

## ❖ (2) Efficiency (skip)

# Discussion on FMP

## ❖ (3) White-box Usage for Sensitive Attribute

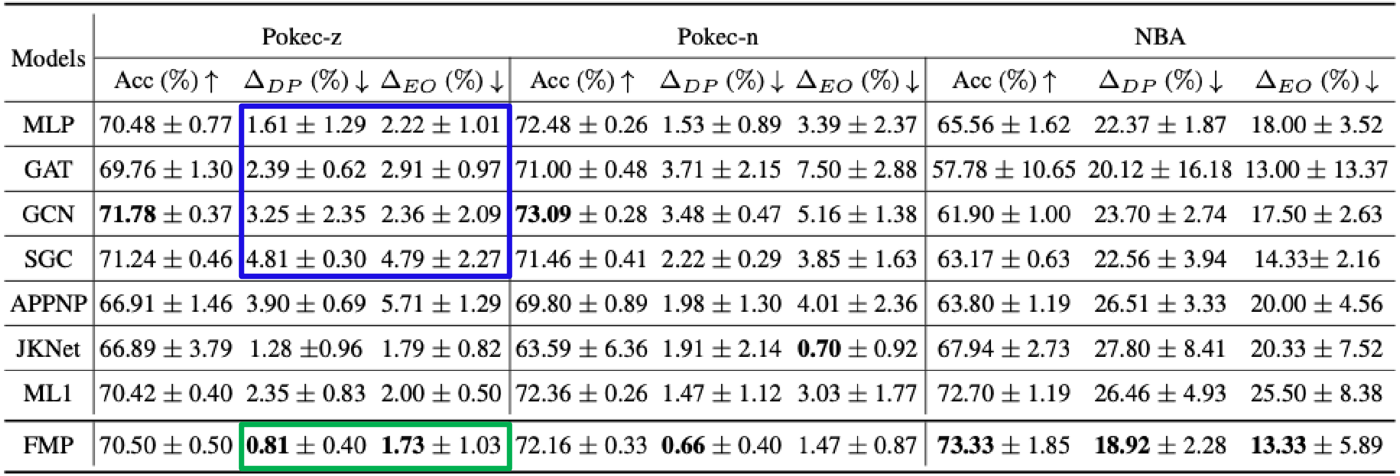
- ❑ A promising property to understand how sensitive attribute usage forces fairness
- ❑ Explicitly achieves graph smoothness and fairness objectives via alternative gradient descent
- ❑ Force the demographic group node representation centers together during forward propagation

**➔ Directly identify that the usage of sensitive attributes >> a white-box usage**

# Experiments

## ❖ Comparison with Existing GNNs

- ❑ Many existing GNNs underperform MLP model on all three datasets in terms of fairness metric
- ❑ FMP consistently achieves **the lowest prediction bias** in terms of DP and EO on all datasets



| Models | Pokec-z | | | Pokec-n | | | NBA | | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc (%) ↑ | $\Delta_{DP}$ (%) ↓ | $\Delta_{EO}$ (%) ↓ | Acc (%) ↑ | $\Delta_{DP}$ (%) ↓ | $\Delta_{EO}$ (%) ↓ | Acc (%) ↑ | $\Delta_{DP}$ (%) ↓ | $\Delta_{EO}$ (%) ↓ |
| MLP | 70.48 ± 0.77 | 1.61 ± 1.29 | 2.22 ± 1.01 | 72.48 ± 0.26 | 1.53 ± 0.89 | 3.39 ± 2.37 | 65.56 ± 1.62 | 22.37 ± 1.87 | 18.00 ± 3.52 |
| GAT | 69.76 ± 1.30 | 2.39 ± 0.62 | 2.91 ± 0.97 | 71.00 ± 0.48 | 3.71 ± 2.15 | 7.50 ± 2.88 | 57.78 ± 10.65 | 20.12 ± 16.18 | 13.00 ± 13.37 |
| GCN | **71.78** ± 0.37 | 3.25 ± 2.35 | 2.36 ± 2.09 | **73.09** ± 0.28 | 3.48 ± 0.47 | 5.16 ± 1.38 | 61.90 ± 1.00 | 23.70 ± 2.74 | 17.50 ± 2.63 |
| SGC | 71.24 ± 0.46 | 4.81 ± 0.30 | 4.79 ± 2.27 | 71.46 ± 0.41 | 2.22 ± 0.29 | 3.85 ± 1.63 | 63.17 ± 0.63 | 22.56 ± 3.94 | 14.33± 2.16 |
| APPNP | 66.91 ± 1.46 | 3.90 ± 0.69 | 5.71 ± 1.29 | 69.80 ± 0.89 | 1.98 ± 1.30 | 4.01 ± 2.36 | 63.80 ± 1.19 | 26.51 ± 3.33 | 20.00 ± 4.56 |
| JKNet | 66.89 ± 3.79 | 1.28 ±0.96 | 1.79 ± 0.82 | 63.59 ± 6.36 | 1.91 ± 2.14 | **0.70** ± 0.92 | 67.94 ± 2.73 | 27.80 ± 8.41 | 20.33 ± 7.52 |
| ML1 | 70.42 ± 0.40 | 2.35 ± 0.83 | 2.00 ± 0.50 | 72.36 ± 0.26 | 1.47 ± 1.12 | 3.03 ± 1.77 | 72.70 ± 1.19 | 26.46 ± 4.93 | 25.50 ± 8.38 |
| FMP | 70.50 ± 0.50 | **0.81** ± 0.40 | **1.73** ± 1.03 | 72.16 ± 0.33 | **0.66** ± 0.40 | 1.47 ± 0.87 | **73.33** ± 1.85 | **18.92** ± 2.28 | **13.33** ± 5.89 |

Table 1: Comparative Results with Baselines on Node Classification.

# Experiments

## ❖ Comparison with Adversarial Debiasing and Regularization

- ❑ FMP can achieve better DP-Acc trade-off
- ❑ Message passing in GNNs does matter -> Different GNNs embrace huge distinctions
  - ➢ Which implies that an appropriate message passing manner potentially leads to better trade-off performance
- ❑ Many GNNs underperforms MLP in low-label homophily coefficient dataset, such as NBA

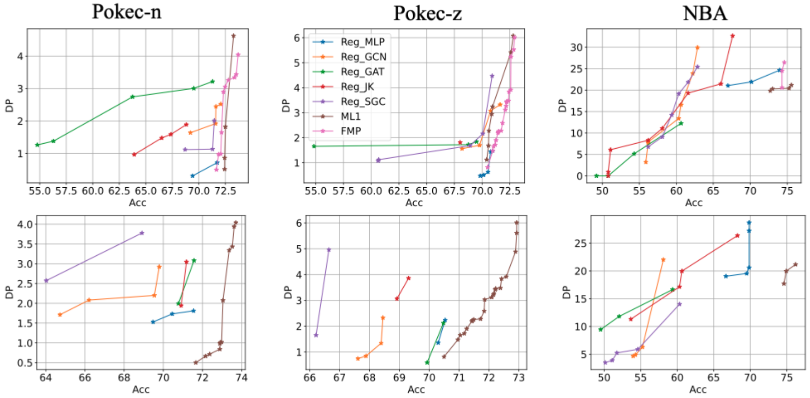


Figure 2: DP and Acc trade-off performance on three real-world datasets compared with adding regularization (Top) and adversarial debiasing (Bottom). The trade-off curve close to the right bottom corner means better trade-off performance. The units for x- and y-axis are percentages (%).

# FMP - Conclusion

- **Improve fairness in graphs from the model architecture perspective**

- **Design a fair message-passing scheme**
  - ❑ To achieve fair prediction for node classification
  - ❑ Using vanilla training loss without data pre-processing

- **Provide a comprehensive discussion of FMP**
  - ❑ Model architecture **interpretation**, efficiency, and the **white-box usage** of sensitive attributes aspects

- **Experimental results on real-world datasets**
  - ❑ Demonstrate the effectiveness of FMP compared with several baselines in node classification tasks

# Disentangling, Amplifying, and Debiasing: Learning Disentangled Representations for Fair Graph Neural Networks

Yeon-Chang Lee[1], Hojung Shin[2], Sang-Wook Kim[2]

[1]UNIST, [2]Hanyang University

AAAI-25

# Motivation

## ❖ Existing methods

- ❑ Often **overlook** a critical aspect in removing sensitive information from the final node embeddings

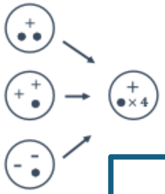
## ❖ Each bias causes sensitive attributes to affect the model:

1) Attribute bias affects how node attributes are distributed across subgroups;

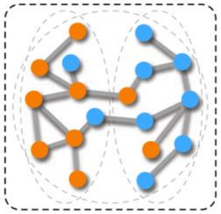
2) Structure bias stems from connections between nodes with similar sensitive attributes;

3) Potential bias arises because of the **interplay** between node attributes and graph structure

   - ➢ The interplay makes neutral attributes **strongly correlated with sensitive attributes**

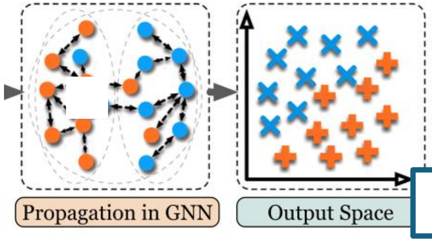


**Attribute Bias**
(Sensitive Attribute(s))



(b) Biased graph topology

**Structure Bias**
(Homophily Effects)



**Potential Bias**
(Message Passing Mechanism)

# Motivation

## ❖ Existing methods

- ❑ Often overlook a critical aspect in removing sensitive information from the final node embeddings

## ❖ Fail to address the unique nature of each bias

- ❑ Leading to inadequate debiasing and persistent unfairness

➢ **Effectively disentangling these biases within node embeddings remains a significant challenge**

# Proposed Method

## ❖ DAB-GNN

- **D**isentangle, **A**mplify, and de**B**ias the attribute, structure, and potential biases through a **GNN** framework
- Operate with two key modules: disentanglement and amplification, and debiasing

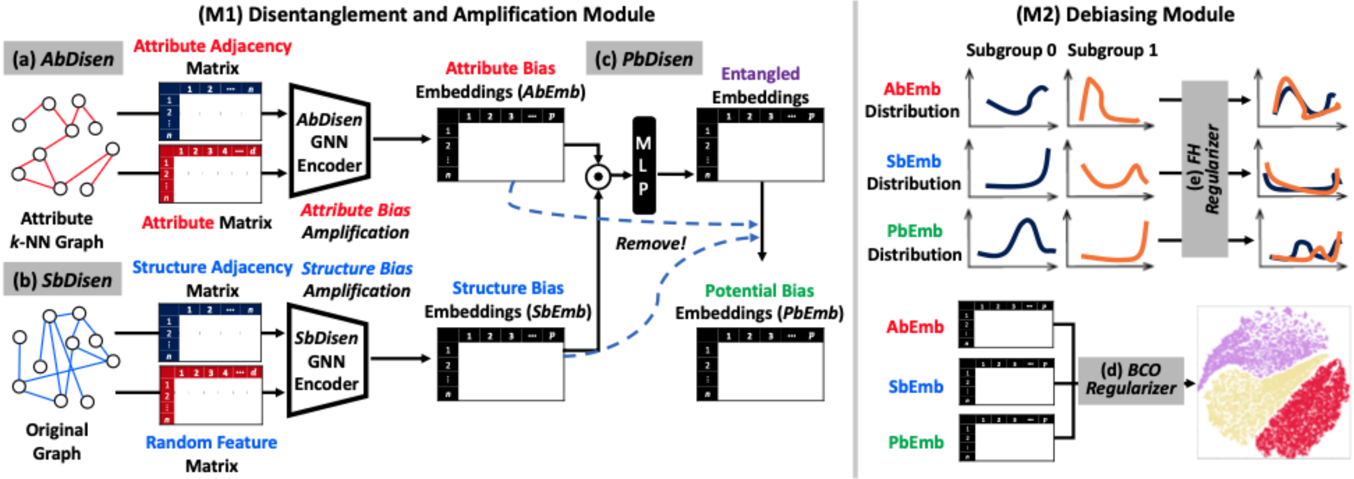
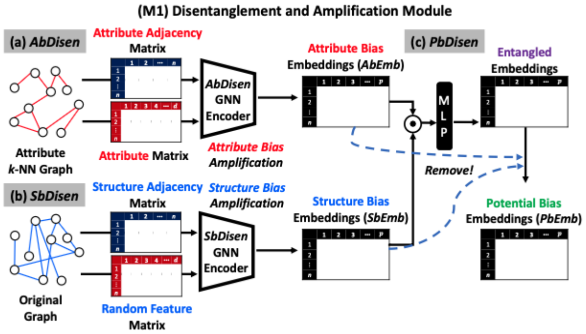


Figure 1: Overview of **DAB-GNN**, which consists of (M1) disentanglement and amplification module, and (M2) debiasing module.

# DAB-GNN Architecture

## ❖ (M1) Disentanglement and Amplification Module

- ❑ DAB-GNN separates/moves away node embeddings into three components:
  - ➢ Attribute bias, structure bias, and potential bias
- ❑ Each component is handled by a specialized disentangler
  - ➢ That identifies and amplifies the corresponding bias
- ❑ Then these disentangled embeddings are concatenated into a comprehensive representation
  - ➢ Which is used for training in various downstream tasks like node classification or link prediction

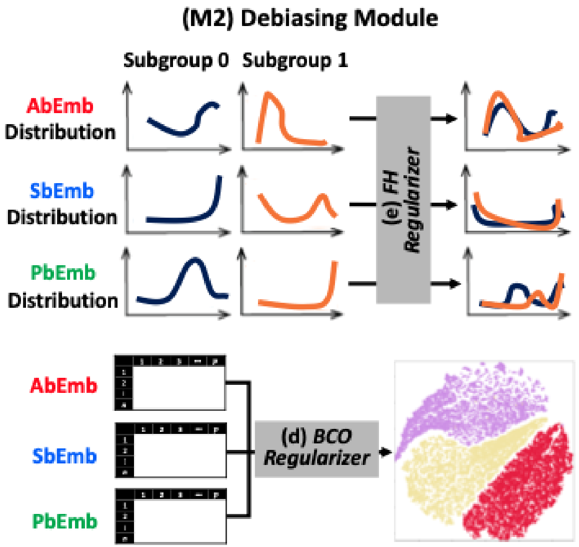

(M1) Disentanglement and Amplification Module
(a) AbDisen
Attribute Adjacency Matrix
Attribute k-NN Graph
Attribute Matrix
AbDisen GNN Encoder
Attribute Bias Amplification
Attribute Bias Embeddings (AbEmb)
(c) PbDisen
MLP
Entangled Embeddings
Remove!
Potential Bias Embeddings (PbEmb)
(b) SbDisen
Structure Adjacency Matrix
Original Graph
Random Feature Matrix
SbDisen GNN Encoder
Structure Bias Amplification
Structure Bias Embeddings (SbEmb)

Proposed Method (2)

# DAB-GNN Architecture

## ❖ M2) Debiasing Module

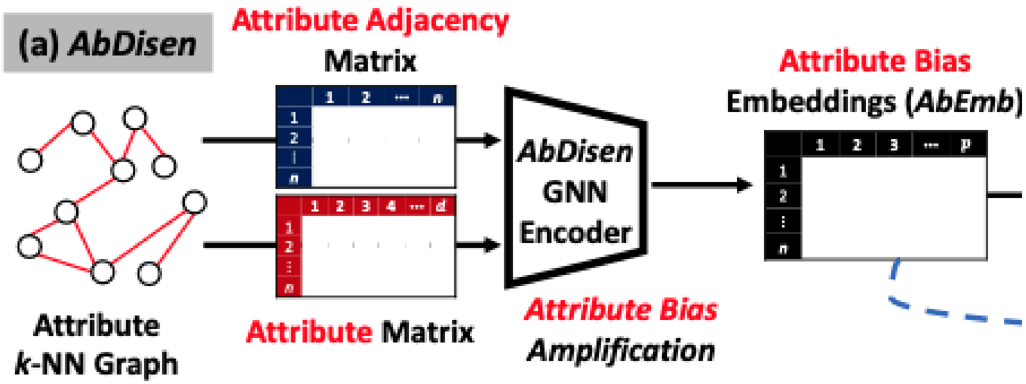
- DAB-GNN refines the disentangled embeddings to ensure they are distinct and fair
- Two key regularizers:
  - Bias contrast optimizer (BCO)
    - Enforce clear separation between different bias embeddings
  - Fairness harmonizer (FH)
    - Reduce the impact of sensitive attributes
    - By minimizing the distance between subgroup distributions



# M1 Module

## ❖ Attribute Bias Disentangler (AbDisen)

- ❑ Leverage a specialized GNN to effectively capture the attribute bias
- ❑ Amplify the attributes bias, as message passing mechanism
  - ➢ Only the information related to node attributes in $X_{attr}$ and $A_{attr}$

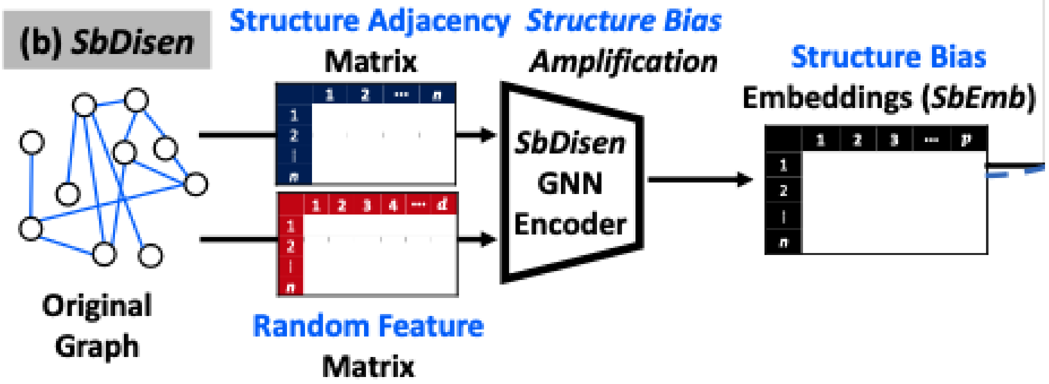


$$\mathbf{H}_{\text{attr}}^{(l+1)} = \sigma\left(\mathbf{A}_{\text{attr}} \cdot \mathbf{H}_{\text{attr}}^{(l)} \cdot \mathbf{W}_{\text{attr}}^{(l)} + \mathrm{b}\right)$$

# M1 Module

## ❖ Structure Bias Disentangler (SbDisen)

- ❑ Leverage another specialized GNN to capture the structure bias
- ❑ Message Passing Mechanism
  - ➢ Update node embedding based solely on the graph’s structure



$$\mathbf{H}_{\text{stru}}^{(l+1)} = \sigma\left(\mathbf{A}_{\text{stru}} \cdot \mathbf{H}_{\text{stru}}^{(l)} \cdot \mathbf{W}_{\text{stru}}^{(l)} + \mathrm{b}\right)$$

DAB-GNN Architecture

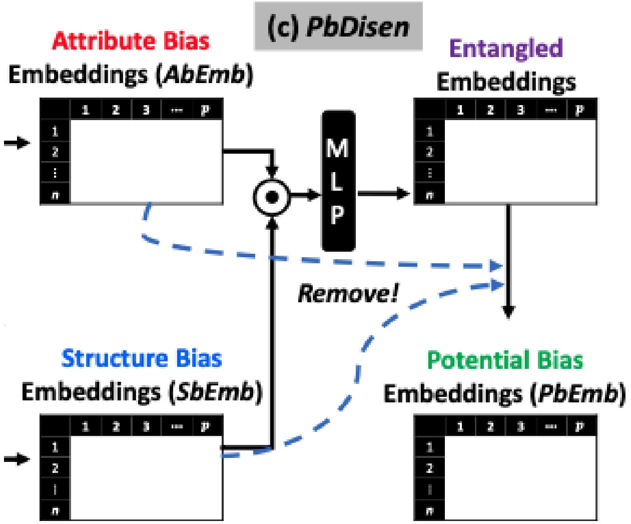
# M1 Module

## ❖ Potential Bias Disentangler (PbDisen)

- ❑ Address the potential bias that arises from the interaction between attribute and structure biases

$$H_{ent} = MLP\ ([H_{attr}|H_{stru}])$$

$$H_{pot} = H_{ent} - H_{attr} - H_{stru}$$



- ❑ Downstream task: Node classification

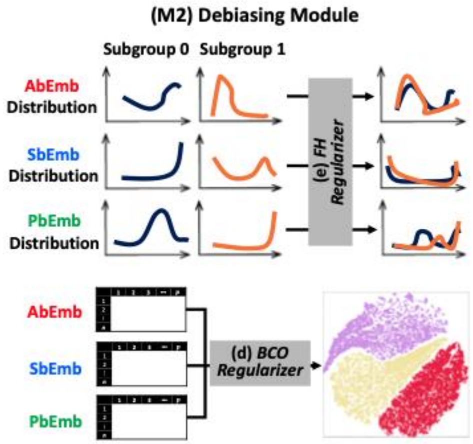
$$H_{final} = [\ H_{attr}|H_{stru}\ |\ H_{pot}]$$

$$\mathcal{L}_{\text{primary}} = -\sum_{i=1}^{n}[\mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i)\log(1 - \hat{\mathbf{y}}_i)]$$

# M2 Debiasing Module

- **Goal: Refining the disentangled embeddings**
  - □ Ensure that predictions are free from biases related to sensitive attributes

- **Embeddings for different bias types**
  - □ May still overlap in the embedding space
  - □ Due to residual similarities or interdependencies
  - □ Crucial to achieve a clear separation of each bias in the embedding space
  - □ Eliminate any sensitive information from the corresponding embeddings



# M2 Debiasing Module

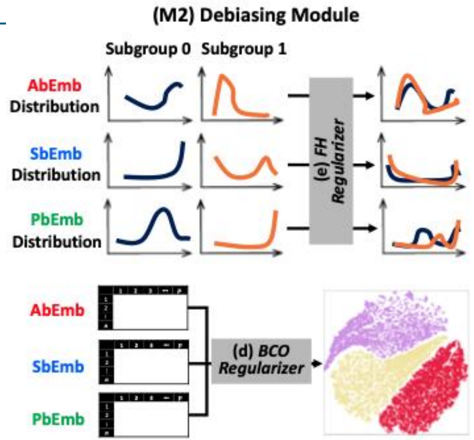
- **Two regularizers**
  - □ Bias Contrast Optimizer (BCO):
    - ➢ Enforce a strong separation between embeddings from different bias components

$$\mathcal{L}_{\text{bco}} = -\sum_{q \neq r} \mathtt{D}_f\left(\mathbf{H}_q, \mathbf{H}_r\right) \Leftarrow \mathtt{D}_f\left(\mathbf{H}_q, \mathbf{H}_r\right) = |\mathbf{H}_q - \mathbf{H}_r|_F.$$

$$q, r \in \{\text{attr}, \text{stru}, \text{pot}\}$$

  - □ Fairness Harmonizer (FH)
    - ➢ Reduce sensitive information in the disentangled embeddings by minimizing the Wasserstein-1 distance

$$\mathcal{L}_{\text{fh}} = \sum_{q \in \{\text{attr},\text{stru},\text{pot}\}} \mathtt{W}\left(\mathcal{P}(\mathbf{H}_q(0)), \mathcal{P}(\mathbf{H}_q(1))\right) \Leftarrow \mathtt{W}(\mathcal{P}, \mathcal{Q}) = \inf_{\gamma \in \Gamma(\mathcal{P},\mathcal{Q})} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|_1],$$



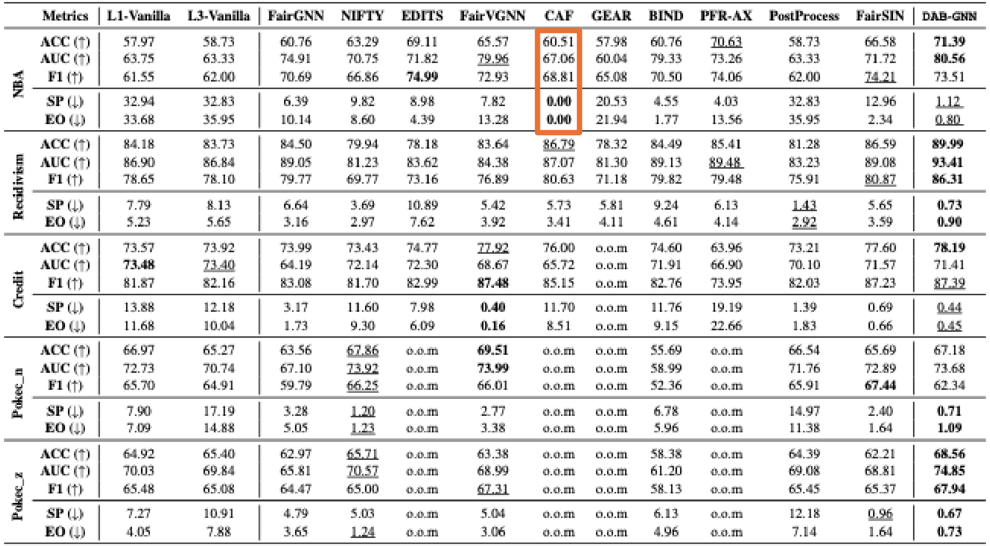
- **Training**:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{primary}} + \alpha \cdot \mathcal{L}_{\text{fh}} + \beta \cdot \mathcal{L}_{\text{bco}};$$

# Experiments

## ❖ Address fairness effectively and enhance accuracy

- ❑ A few cases have slightly lower accuracy/higher fairness values
- ❑ CAF : Underscore difficulty in balancing accuracy and fairness



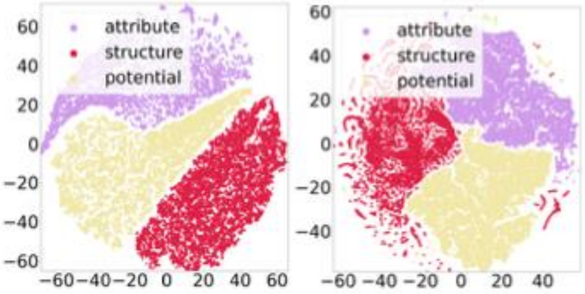
| | Metrics | L1-Vanilla | L3-Vanilla | FairGNN | NIFTY | EDITS | FairVGNN | CAF | GEAR | BIND | PFR-AX | PostProcess | FairSIN | DAB-GNN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NBA | ACC (↑) | 57.97 | 58.73 | 60.76 | 63.29 | 69.11 | 65.57 | 60.51 | 57.98 | 60.76 | 70.63 | 58.73 | 66.58 | **71.39** |
| | AUC (↑) | 63.75 | 63.33 | 74.91 | 70.75 | 71.82 | 79.96 | 67.06 | 60.04 | 79.33 | 73.26 | 63.33 | 71.72 | **80.56** |
| | F1 (↑) | 61.55 | 62.00 | 70.69 | 66.86 | **74.99** | 72.93 | 68.81 | 65.08 | 70.50 | 74.06 | 62.00 | 74.21 | 73.51 |
| | SP (↓) | 32.94 | 32.83 | 6.39 | 9.82 | 8.98 | 7.82 | **0.00** | 20.53 | 4.55 | 4.03 | 32.83 | 12.96 | 1.12 |
| | EO (↓) | 33.68 | 35.95 | 10.14 | 8.60 | 4.39 | 13.28 | **0.00** | 21.94 | 1.77 | 13.56 | 35.95 | 2.34 | 0.80 |
| Recidivism | ACC (↑) | 84.18 | 83.73 | 84.50 | 79.94 | 78.18 | 83.64 | 86.79 | 78.32 | 84.49 | 85.41 | 81.28 | 86.59 | **89.99** |
| | AUC (↑) | 86.90 | 86.84 | 89.05 | 81.23 | 83.62 | 84.38 | 87.07 | 81.30 | 89.13 | 89.48 | 83.23 | 89.08 | **93.41** |
| | F1 (↑) | 78.65 | 78.10 | 79.77 | 69.77 | 73.16 | 76.89 | 80.63 | 71.18 | 79.82 | 79.48 | 75.91 | 80.87 | **86.31** |
| | SP (↓) | 7.79 | 8.13 | 6.64 | 3.69 | 10.89 | 5.42 | 5.73 | 5.81 | 9.24 | 6.13 | 1.43 | 5.65 | **0.73** |
| | EO (↓) | 5.23 | 5.65 | 3.16 | 2.97 | 7.62 | 3.92 | 3.41 | 4.11 | 4.61 | 4.14 | 2.92 | 3.59 | **0.90** |
| Credit | ACC (↑) | 73.57 | 73.92 | 73.99 | 73.43 | 74.77 | 77.92 | 76.00 | o.o.m | 74.60 | 63.96 | 73.21 | 77.60 | **78.19** |
| | AUC (↑) | **73.48** | 73.40 | 64.19 | 72.14 | 72.30 | 68.67 | 65.72 | o.o.m | 71.91 | 66.90 | 70.10 | 71.57 | 71.41 |
| | F1 (↑) | 81.87 | 82.16 | 83.08 | 81.70 | 82.99 | **87.48** | 85.15 | o.o.m | 82.76 | 73.95 | 82.03 | 87.23 | 87.39 |
| | SP (↓) | 13.88 | 12.18 | 3.17 | 11.60 | 7.98 | **0.40** | 11.70 | o.o.m | 11.76 | 19.19 | 1.39 | 0.69 | 0.44 |
| | EO (↓) | 11.68 | 10.04 | 1.73 | 9.30 | 6.09 | **0.16** | 8.51 | o.o.m | 9.15 | 22.66 | 1.83 | 0.66 | 0.45 |
| Pokec_n | ACC (↑) | 66.97 | 65.27 | 63.56 | 67.86 | o.o.m | **69.51** | o.o.m | o.o.m | 55.69 | o.o.m | 66.54 | 65.69 | 67.18 |
| | AUC (↑) | 72.73 | 70.74 | 67.10 | 73.92 | o.o.m | **73.99** | o.o.m | o.o.m | 58.99 | o.o.m | 71.76 | 72.89 | 73.68 |
| | F1 (↑) | 65.70 | 64.91 | 59.79 | 66.25 | o.o.m | 66.01 | o.o.m | o.o.m | 52.36 | o.o.m | 65.91 | **67.44** | 62.34 |
| | SP (↓) | 7.90 | 17.19 | 3.28 | 1.20 | o.o.m | 2.77 | o.o.m | o.o.m | 6.78 | o.o.m | 14.97 | 2.40 | **0.71** |
| | EO (↓) | 7.09 | 14.88 | 5.05 | 1.23 | o.o.m | 3.38 | o.o.m | o.o.m | 5.96 | o.o.m | 11.38 | 1.64 | **1.09** |
| Pokec_z | ACC (↑) | 64.92 | 65.40 | 62.97 | 65.71 | o.o.m | 63.38 | o.o.m | o.o.m | 58.38 | o.o.m | 64.39 | 62.21 | **68.56** |
| | AUC (↑) | 70.03 | 69.84 | 65.81 | 70.57 | o.o.m | 68.99 | o.o.m | o.o.m | 61.20 | o.o.m | 69.08 | 68.81 | **74.85** |
| | F1 (↑) | 65.48 | 65.08 | 64.47 | 65.00 | o.o.m | 67.31 | o.o.m | o.o.m | 58.13 | o.o.m | 65.45 | 65.37 | **67.94** |
| | SP (↓) | 7.27 | 10.91 | 4.79 | 5.03 | o.o.m | 5.04 | o.o.m | o.o.m | 6.13 | o.o.m | 12.18 | 0.96 | **0.67** |
| | EO (↓) | 4.05 | 7.88 | 3.65 | 1.24 | o.o.m | 3.06 | o.o.m | o.o.m | 4.96 | o.o.m | 7.14 | 1.64 | **0.73** |

Table 4: Accuracy and fairness results of DAB-GNN and competitors across five real-world datasets. (↑) and (↓) mean higher and lower values are better, respectively; 'o.o.m' denotes 'out of memory.'

# Results

## ❖ Disentangled Embeddings Analysis

- ❑ Different colors indicate different types of bias embeddings
- ❑ Bias embeddings are well-separated into distinct clusters
- ❑ Successfully isolates various biases present in graph data



(a) **Recidivism** (b) **Credit**

Figure 2: Visualization of disentangled node embeddings by using t-SNE: *AbEmb*, *SbEmb*, and *PbEmb*.

# DAB-GNN - Conclusion

- **Existing fairness-aware GNN methods:**
  - ❑ Entanglement of different bias types in the final node embeddings
  - ❑ Lead to difficulty in their comprehensive debiasing

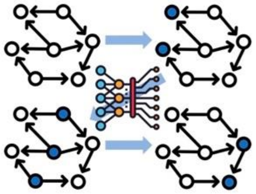
- **DAB-GNN, a novel GNN framework**
  - ❑ Disentangle, amplify, and debias the attribute, structure, and potential biases within node embeddings

- **Extensive experiments on five real-world graph datasets**
  - ❑ DAB-GNN outperforms ten state-of-the-art competitors in balancing accuracy and fairness,
  - ❑ While validating the effectiveness of design choices

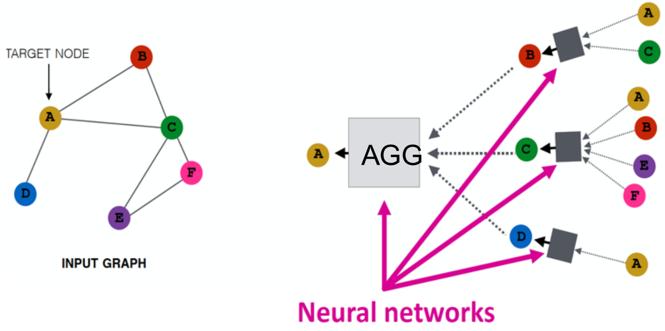
# Conclusion

## ❖ Graph Learning

- ❑ Gain a deeper understanding of graph-structured data
- ❑ Has achieved remarkable success in a myriad of high-impact real-world applications
- ❑ GNNs also can give unfair predictions due to the societal bias in the data
- ❑ The bias in the training data even can be magnified
  - ➢ By the graph topology and message-passing mechanism of GNNs



Pipeline of Graph Mining



Graph Neural Networks

# Conclusion

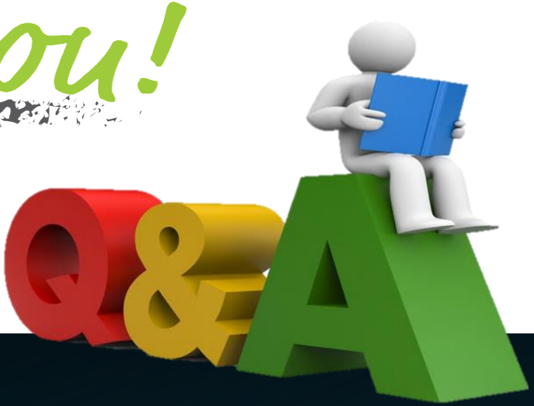
## ❖ Fairness-aware Graph Learning

- ❑ Problem Definition
  - ➢ (1)The **raw features** of nodes could be statistically **correlated to the sensitive attribute**
    - ▪ Lead to sensitive information leakage in encoded representations
  - ➢ (2) **Homophily effects**: nodes with the same sensitive attribute tend to link with each other
    - ▪ Make the node representations in the same sensitive group more similar during message passing

- ❑ Solution Methods
  - ➢ FMP: A new fairness-aware GNN architecture called **F**air **M**essage **P**assing
  - ➢ DAB-GNN: **D**isentangle, **A**mplify, and de**B**ias the attribute, structure, and potential biases through a **GNN** framework

# Thank You!

바이



HTET ARKAR
hak3601@gmail.com