Lab Seminar

Fairness-aware Graph Learning

HTET ARKAR

Undergraduate

School of Computer Science and Engineering

Chung-Ang University

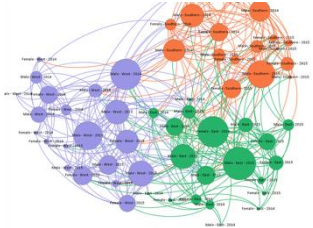
Outline

- Introduction
- Fairness Notation
- Problems in Graph Mining
- Previous Works
- Proposed Method
- Experiments
- Conclusion

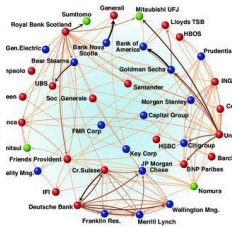
Introduction

❖ Networks

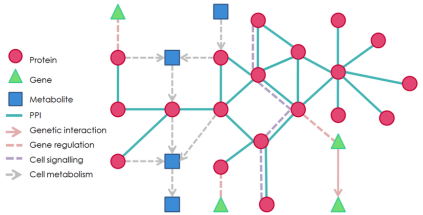
- ❑ A general language for describing and modeling complex systems
- ❑ Many data are networks such as



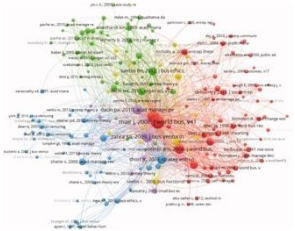
Social Networks



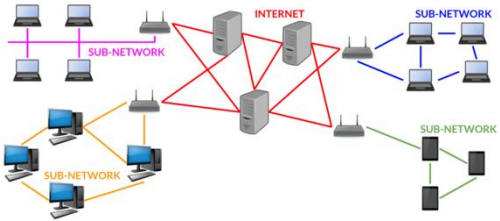
Economic Networks



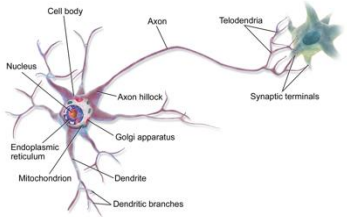
Biological Networks



Citation Networks



Internet



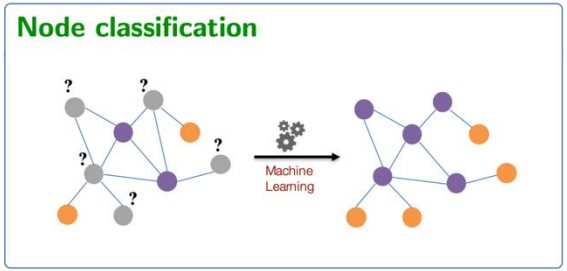
Networks of Neurons

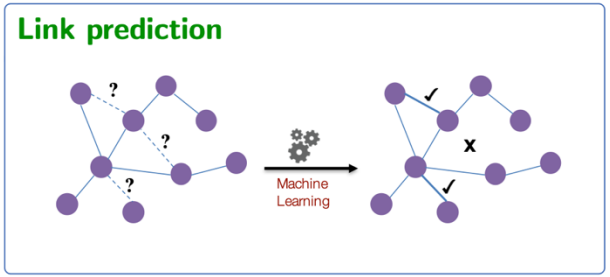
https://sharif.edu/~beigy/courses/13992/40719/Lect-30.pdf

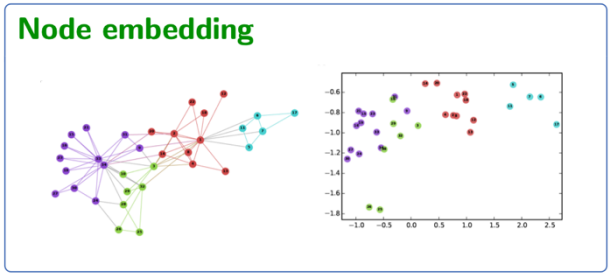
Introduction

❖ Graph Mining

- ❑ Graph-structured data is pervasive in diverse real-world applications
  - ➢ e.g., E-commerce, health care, traffic forecasting , and drug discovery
- ❑ Graph mining algorithms have been proposed to gain a deeper understanding of such data
  - ➢ Promising performance on graph analytical tasks such as node classification and link prediction



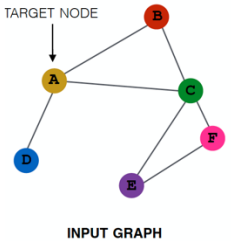


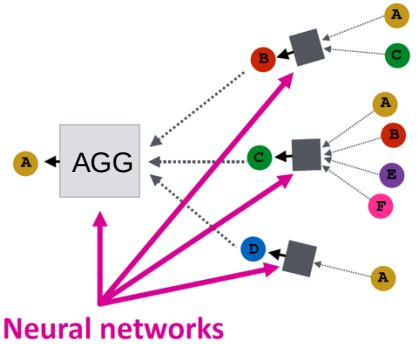


❖ Graph Neural Networks

- ❑ Deep learning architectures for graph-structured data
- ❑ **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

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

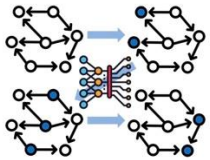


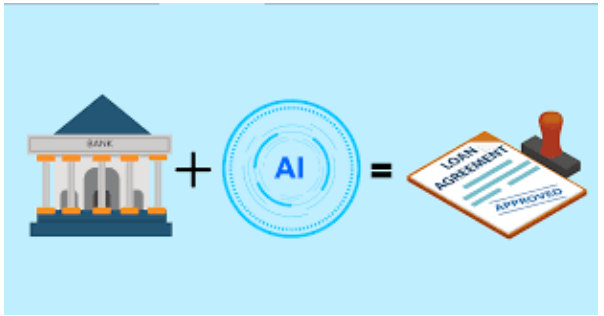


Graph Mining Algorithms Cont'd

❖ Example: A Loan Approval Model

- ❑ To decide whether to approve or deny loan applications
- ❑ The model is trained on **historical data** containing information about applicants, such as:
  - ➢ Income
  - ➢ Credit Score
  - ➢ Employment Status
  - ➢ Debt-to-Income Ratio
  - ➢ Race (Sensitive Attribute)
  - ➢ Gender (Sensitive Attribute)





- ❑ Historical loan decisions were influenced by **systemic biases** (e.g., racial discrimination in lending)
- ❑ The trained model **inherently learns these biases**

Unfair Decision Making

❖ Demographic Bias

- ❑ Suppose historical data shows that
  - ➢ Applicants from a certain racial group were **historically denied loans more often**
  - ➢ Even when they had the same financial credentials as others
- ❑ The model learns this pattern and continues to deny loans at a higher rate to applicants from this group
  - ➢ Reinforcing historical discrimination

❖ Disparate Impact

❖ Fairness vs. Accuracy Trade-off

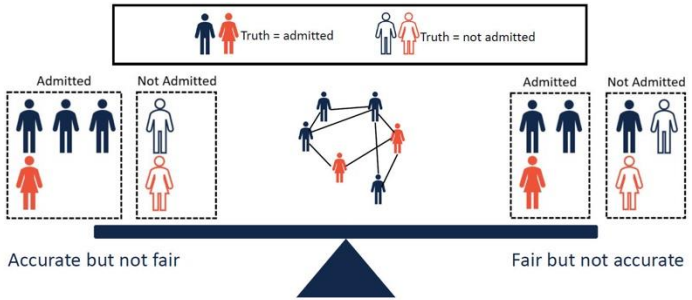
Unfair Decision Making

- **Demographic Bias**
- **Disparate Impact**
  - Even if race is not explicitly included in the model
    - Other correlated attributes (e.g., ZIP code, education level) may indirectly encode race, leading to biased decisions
  - This is an example of proxy bias, where seemingly neutral features capture sensitive information

- **Fairness vs. Accuracy Trade-off**

Unfair Decision Making

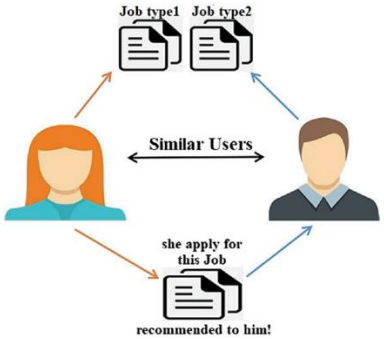
- **Demographic Bias**
- **Disparate Impact**
- **Fairness vs. Accuracy Trade-off**
  - The bank wants the model to be both fair and accurate
  - Simply removing sensitive attributes (like race) does not eliminate bias
    - As the model still relies on correlated features
  - Applying fairness constraints might slightly reduce accuracy
    - But ensure that applicants are treated equitably



Graph Learning

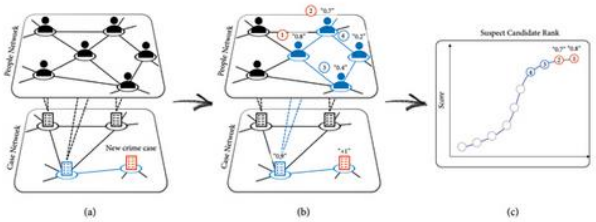
❖ Most of GM algorithms lack of fairness consideration

- ❑ Consequently, they could yield discriminatory results towards certain populations
- ❑ When such algorithms are exploited in human-centered/high-stake applications
  - ➢ e.g., Social network-based job recommendation system/ disaster response, criminal justice, loan approval



Job Recommendation System

Loan approval system

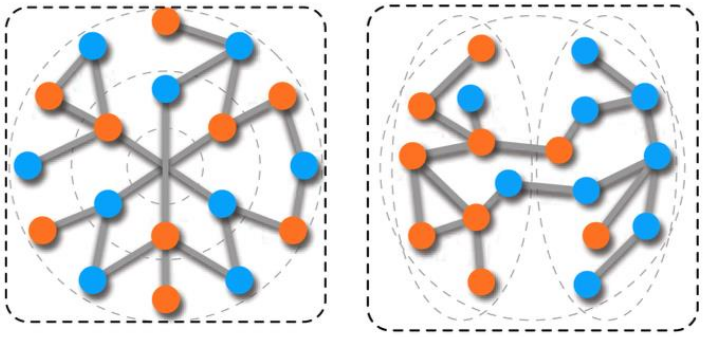


Criminal Susception

Fairness in Graph Mining

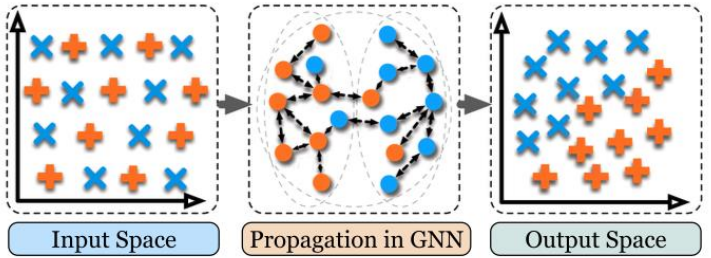
❖ Problems

- ❑ Predictions based on node embeddings learned by GNNs can be unfair
- ❑ (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

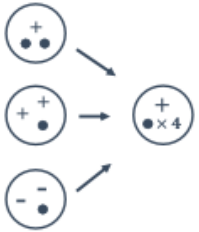


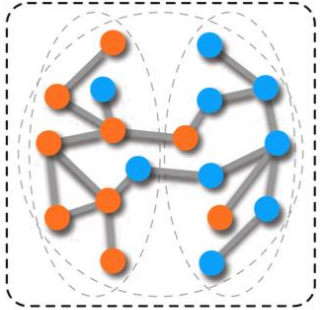
(c) An example of biased node embeddings (learned via information propagation mechanism of GNNs) induced by biased input graph.

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.

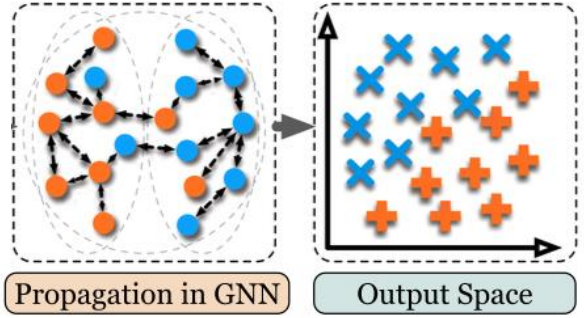
Problem Definition

❖ Biases Definition





(b) Biased graph topology



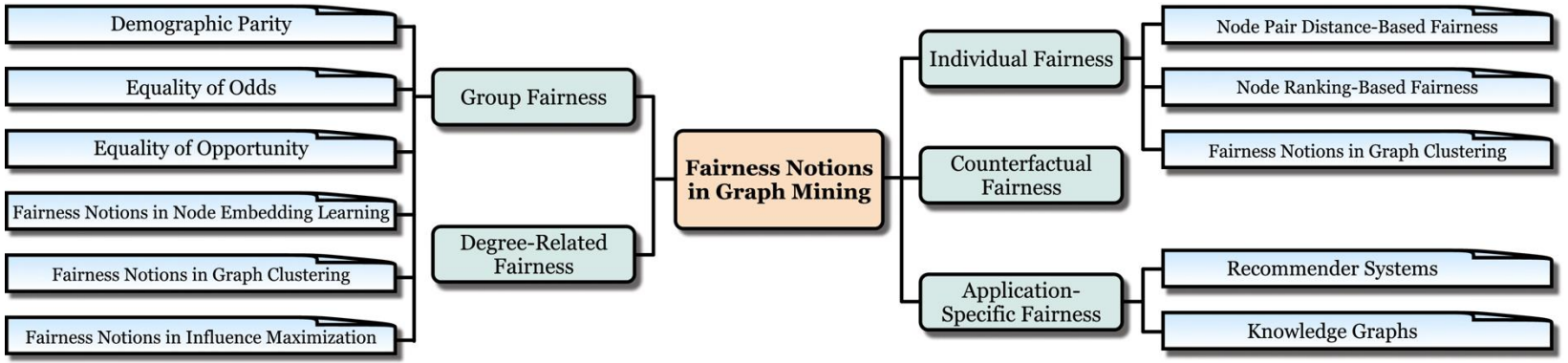
Attribute Bias
(Sensitive Attribute(s))

Structure Bias
(Homophily Effects)

Potential Bias
(Message Passing Mechanism)

Fairness Notation

- **The quality of treating people or in a way that is right or reasonable**
- **Existing Notations:**



Taxonomy of algorithmic fairness notions in graph mining algorithms

Dong, Yushun, et al. "Fairness in graph mining: A survey." *IEEE Transactions on Knowledge and Data Engineering* 35.10 (2023): 10583-10602.

Fairness Notation

❖ Group Fairness

- ❑ Defined upon sensitive subgroups
- ❑ Population can be divided into different demographic subgroups (sensitive subgroups)
- ❑ Subgroups are divided by
  - ➢ Features protected by law to avoid being abused (e.g., race and gender)
  - ➢ Features that users are usually unwilling to share (e.g., occupation, age)
  - ➢ $S \in \{0,1\}$: sensitive features

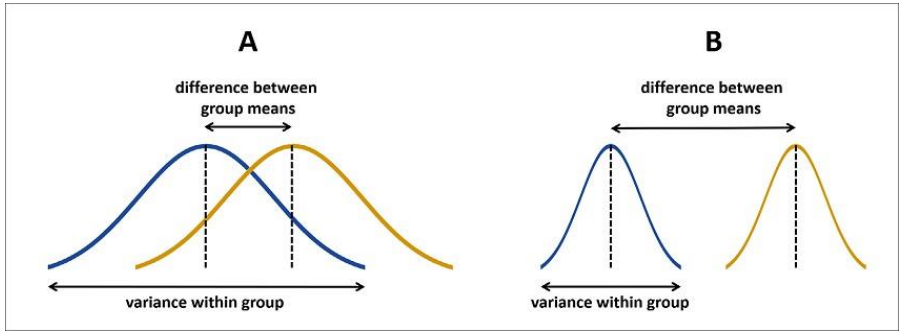
"Generally speaking, group fairness requires that the algorithm should not yield discriminatory predictions or decisions against individuals from any specific sensitive subgroup."

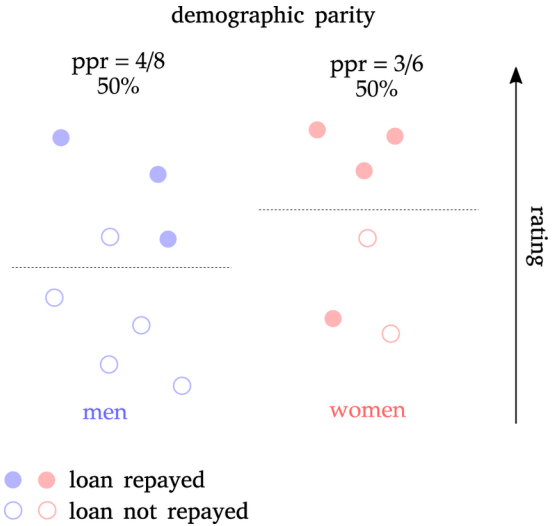
Group Fairness

❖ Demographic Parity (DP) *(a.k.a. Statistical Parity and Independence - **SP**)*

- ❑ Achieved if the model yields the **same acceptance rate** for individuals in both sensitive subgroups
- ❑ ΔDP is defined when both the predicted labels and sensitive feature(s) are binary

$$\Delta DP = |P(\hat{Y} = 1 \mid S = 0) - P(\hat{Y} = 1 \mid S = 1)|$$





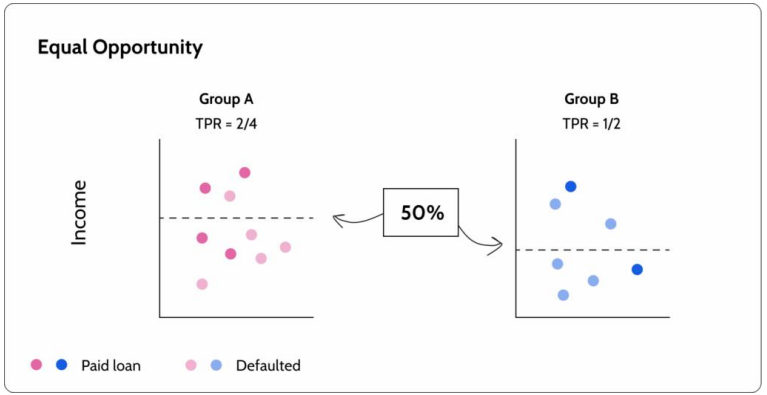
Group Fairness

❖ Equality of Opportunity (EO)

- ❑ Only requires the positive predictions to be independent of sensitive feature(s) for individuals with positive ground truth labels

$$\Delta EO = |P(\hat{Y} = 1 \mid Y = 1, S = 0) - P(\hat{Y} = 1 \mid Y = 1, S = 1)|$$

  - ➢ $\hat{Y} = 1$: an advantaged prediction



FairSIN: Achieving Fairness in Graph Neural Networks through Sensitive Information Neutralization

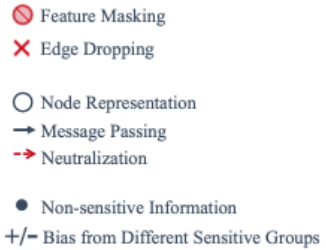
Cheng Yang[1], Jixi Liu[1], Yunhe Yan[1], Chuan Shi[1*]

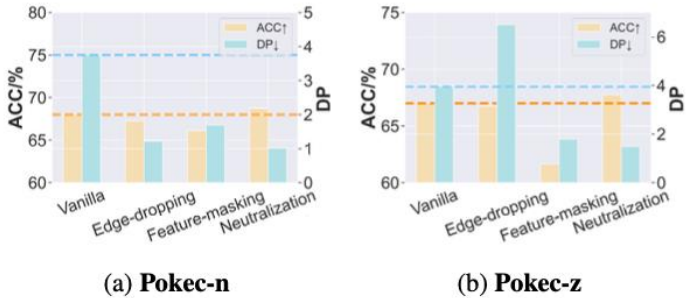
[1]Beijing University of Posts and Telecommunications, China

AAAI-24

❖ Previous Works

- ❑ Recent SOTA methods usually employ feature masking or topology modification
- ❑ Filter out sensitive biases during message passing
- ❑ Unavoidably lead to the loss of useful non-sensitive information
- ❑ Sup-optimal balance between accuracy and fairness





(a) **Pokec-n** (b) **Pokec-z**

Motivation verification on Pokec datasets

(a) Original Message Passing **(b) Filtering-Based Method**

FairSIN

❖ Proposed Methods

- ❑ Introduce extra Fairness-facilitating Features (F3) to node features or representations
  - ➢ The sensitive biases (+/- symbols) can be neutralized
  - ➢ Expected to provide additional non-sensitive feature information (dot symbols)
  - ➢ Enabling a better trade-off between predictive performance and fairness
- ❑ Node features or representations can be debiased before message passing
  - ➢ By emphasizing the features of each node's heterogeneous neighbors as F3
  - ➢ Heterogeneous neighbors: Neighbors with different sensitive attributes

(c) Neutralization-based Method (Ours)

Homogeneous neighbors: same sensitive attribute
Heterogeneous neighbors: different/opposite sensitive attribute

❖ Real World Dataset

- ❑ Some nodes in real-world graphs have very few or even no heterogeneous neighbors
- ❑ Calculation of F3 can be infeasible or very uncertain
- ❑ Propose to train **an estimator**
  - ➢ To predict the average features or representations of a node's heterogeneous neighbors given its own feature
  - ➢ Nodes with rich heterogeneous neighbors can transfer their knowledge to other nodes through the estimator

Dataset	Bail	Pokec-n	Pokec-z
# Nodes	18,876	66,569	67,797
# Features	18	266	277
# Edges	321,308	729,129	882,765
Node label	Bail decision	Working field	Working field
Sens. Attri.	Race	Region	Region
Avg. Deg.	34.04	16.53	19.23
Avg. H-Deg.	15.79	0.73	0.90

Table 1: Dataset statistics. "H-" means "heterogeneous".

Theoretical Analysis

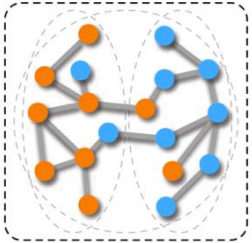
❖ Movitation

- ❑ To measure the sensitive information leakage
  - ➢ By the conditional entropy between sensitive attributes and node representations
- ❑ How the message passing computation exacerbates the leakage problem of sensitive information

❖ Graph Generation

- ❑ Draw node features and sensitive attribute from a joint prior distribution $(x_i, s_i) \sim prior$
- ❑ To obtain graph G, each node v_i samples its in-degree neighbor set N_i by the homophily assumption
- ❑ Homophily assumption, $P_i^{same} > P_i^{diff}$
- ❑ Average feature of v_i's in-degree neighbors,

$$x_i^{neigh} = P_i^{same}\, x_i^{same} + P_i^{diff}\, x_i^{diff}$$

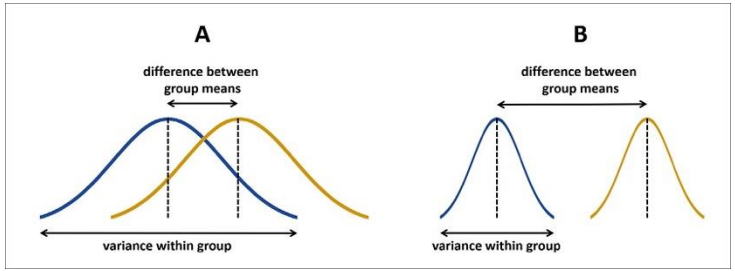


(b) Biased graph topology

Theoretical Analysis

❖ Quantifying Sensitive Information Leakage

- Conditional Entropy, $H(s|x) = -E_{(x,s)\sim prior} \log \boldsymbol{P(s|x)}$
  - $\boldsymbol{P(s \mid x)}$: a predictor that estimates sensitive attributes given node representations
- Adopt a linear intensity function $\boldsymbol{\mathcal{D}_\theta}$ with the parameter θ to define the predictive capability
- $\boldsymbol{\mathcal{D}_\theta}(s^0|x) \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}^2)$ & $\boldsymbol{\mathcal{D}_\theta}(s^1|x) \sim \mathcal{N}(\boldsymbol{\mu}_{ic}, \boldsymbol{\sigma}^2)$
  - $\boldsymbol{\mathcal{D}_\theta}$: More likely to assign larger intensity score to the true sensitive attribute given node representations
  - $\boldsymbol{\mu}_c = \boldsymbol{\mu}_{ic}$: Not distinguish the sensitive attribute from representations
  - Define Predictor $\widehat{\boldsymbol{P}}(\boldsymbol{s}|\boldsymbol{x})$ by normalization the intensity function $\boldsymbol{\mathcal{D}_\theta}$



Theoretical Analysis

❖ Message Passing Can Exacerbate Sensitive Biases

- ❑ Lead to more serious sensitive information leakage problem.
- ❑ Assume that $\boldsymbol{\mu}_{\mathrm{c}} > \boldsymbol{\mu}_{\mathrm{ic}}$ and $x_i' = x_i + x_i^{neigh}$ then

$$\mathbb{E}\{\mathcal{D}_{\boldsymbol{\theta}}(s_i^0|x_i') - \mathcal{D}_{\boldsymbol{\theta}}(s_i^1|x_i')\} > \mathcal{D}_{\boldsymbol{\theta}}(s_i^0|x_i) - \mathcal{D}_{\boldsymbol{\theta}}(s_i^1|x_i)$$

- ❑ The predictor $\widehat{\boldsymbol{P}}(\boldsymbol{s}|\boldsymbol{x})$ can identify the sensitive attributes more accurately

❖ Solution

- ❑ Either modify the graph structure to decrease $\boldsymbol{P_i^{same}} - \boldsymbol{P_i^{diff}}$
- ❑ OR modify the node feature before message passing to decrease $\boldsymbol{\mu}_{\mathrm{c}} - \boldsymbol{\mu}_{\mathrm{ic}}$

Implementations of FairSIN

❖ Proposed Method: FairSIN with two variants

- ❑ Data-centric Perspectives
  - ➢ A pre-processing manner, and modify the graph structure or node features before the training of GNN encoder
- ❑ Model-centric Perspectives
  - ➢ Extend FairSIN-F by jointly learning the MLP_φ and GNN encoder

❖ Data-centric Variant

- ❑ (1) Graph modification (FairSIN-G):
  - ➢ $\boldsymbol{\delta} > 0$, hyper-parameter

$$\mathbf{A}_{ij} = \begin{cases} 1+\delta, & \text{if } (v_i, v_j) \in \mathcal{E} \text{ and } s_i \neq s_j \\ 1, & \text{if } (v_i, v_j) \in \mathcal{E} \text{ and } s_i = s_j \\ 0, & \text{if } (v_i, v_j) \notin \mathcal{E} \end{cases}$$

Data-centric Varient

❖ (2) Feature modification (FairSIN-F)

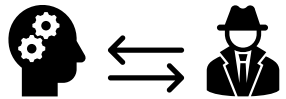
- ❑ Compute average feature of each node v_i's heterogeneous neighbors as $\mathbf{x}_i^{diff} = \frac{1}{|\mathcal{N}_i^{diff}|} \sum_{v_j \in \mathcal{N}_i^{diff}} \mathbf{x}_j$
- ❑ In real world graph, **very few or no heterogeneous neighbors**
  - ➢ Make calculation infeasible or very uncertain
- ❑ Train MLP to **estimate** x_i^{diff} using MSE loss : $\mathcal{L}_F = \frac{1}{|\mathcal{V}|} \sum_{i:|\mathcal{N}_i^{diff}| \geq 1} \| \mathrm{MLP}_\phi(\mathbf{x}_i) - \mathbf{x}_i^{diff} \|^2$
- ❑ Neutralize each node feature as $\tilde{x}_i = x_i + \delta \, MLP_\emptyset(x_i)$
- ❑ Nodes with rich heterogeneous neighbors can transfer their knowledge to other nodes through the MLP

Implementations of FairSIN

❖ Model-centric Variant

- ❑ Extend FairSIN-F by jointly learning the $MLP_{\emptyset}$ and GNN encoder
  - ➢ MSE loss: min min $\mathcal{L}_F^k$
  - ➢ $\boldsymbol{\delta}$ = (0,1]
- ❑ Introduce a **discriminator** module $MLP_{\boldsymbol{\psi}}$
  - ➢ To impose extra fairness constraints on the encoded representations
  - ➢ Train with binary cross-entropy loss: max $\mathcal{L}_D$
- ❑ Downstream classification task: $\mathcal{L}_{\mathrm{T}}$
- ❑ Full Model:
  - ➢ (1) Update each MLP_{ϕ}^k by minimizing $L_F^k - L_D$;
  - ➢ (2) Update GNN encoder by minimizing $L_T - L_D$;
  - ➢ (3) Update discriminator MLP_{ψ} by minimizing $\mathcal{L}_D$;

$$\tilde{\mathbf{H}}^k = \mathbf{H}^k + \delta^k \, \mathrm{MLP}_{\phi}^k(\mathbf{H}^k),$$
$$\mathbf{H}^{k+1} = \mathrm{MessagePassing}(\tilde{\mathbf{H}}^k),$$



estimator discriminator

$$\mathbb{E}\{\mathcal{D}_\theta(s_i|x_i + \delta x_i^{diff}) - \mathcal{D}_\theta(\bar{s}_i|x_i + \delta x_i^{diff})\}$$
$$= (1-\delta)(\mu_c - \mu_{ic}) < \mathbb{E}\{\mathcal{D}_\theta(s_i|x_i) - \mathcal{D}_\theta(\bar{s}_i|x_i)\}$$

How Does FairSIN Work?

❖ Empirical verification

- ❑ Message passing enlarges the sensitive biases for both raw and neutralized features
  - ➢ Validating the theoretical analysis
- ❑ Neutralized features have much less sensitive information leakage
  - ➢ Demonstrating the effectiveness of F3
- ❑ MLP_{ϕ}^{k} can provide additional information when calculating representations
  - ➢ Especially for the nodes with few heterogeneous neighbors

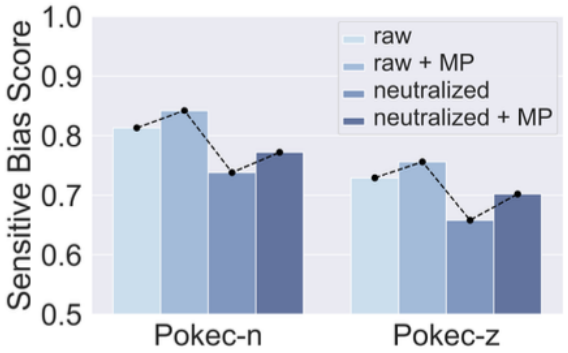


Figure 3: Sensitive biases in four groups of features. The biases are measured by average $\hat{P}_{\theta}(s|x)$, and larger scores indicate more serious sensitive leakage in the representations.

Data-centric v.s. Model-centric Variants

❖ Data-centric variants

- ❑ Task-irrelevant and thus, can be employed for various downstream scenarios
- ❑ More computationally efficient
  - ➢ For example, Debias a graph dataset by neutralizing node features in advance
  - ➢ And then graph machine learning algorithms can be trained as usual

❖ Model-centric variant

- ❑ Model-agnostic, and can be combined with any GNN encoders
- ❑ Allow to further neutralize the internal representations in each GNN layer
- ❑ Enable additional fairness constraint from an adversarial discriminator
- ❑ Different parts of the model can learn and improve together
  - ➢ Thereby achieving better accuracy and fairness

Experiments

❖ Fairness Matrix

- ❑ Demographic (Statistical) Parity (DP)
- ❑ Equal Opportunity (EO)
- ❑ The lower DP and EO, the better the fairness

❖ Datasets

- ❑ Pokec-n/Pokec-z: very few hetero neighbors
- ❑ Bali : fair homo/hetero neighbors

Dataset	Bail	Pokec-n	Pokec-z
# Nodes	18,876	66,569	67,797
# Features	18	266	277
# Edges	321,308	729,129	882,765
Node label	Bail decision	Working field	Working field
Sens. Attri.	Race	Region	Region
Avg. Deg.	34.04	16.53	19.23
Avg. H-Deg.	15.79	0.73	0.90

Table 1: Dataset statistics. “H-” means “heterogeneous”.

Main Results

❖ Effectiveness of Model-centric Variant FairSIN

- ❑ Neutralized-based strategy achieve a better trade-off than SOTA
- ❑ Both the best overall classification performance and group fairness under different GNN encoders
- ❑ Pokec-n/-z : very few hetero neighbors
- ❑ Improvement achieved by FairSIN - aligning with motivation and model design

Encoder	Method	Bail			Pokec_n			Pokec_z		
		ACC↑	DP↓	EO↓	ACC↑	DP↓	EO↓	ACC↑	DP↓	EO↓
GCN	vanilla	87.55±0.54	6.85±0.47	5.26±0.78	68.55±0.51	3.75±0.94	2.93±1.15	66.78±1.09	3.95±1.03	2.76±0.95
	FairGNN	82.94±1.67	6.90±0.17	4.65±0.14	67.36±2.06	3.29±2.95	2.46±2.64	67.65±1.65	1.87±1.95	1.32±1.42
	EDITS	84.49±2.27	6.64±0.39	7.51±1.20	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	82.36±3.91	5.78±1.29	4.72±1.08	67.24±0.49	1.22±0.94	2.79±1.24	66.74±0.93	6.50±2.16	7.64±1.77
	FairVGNN	84.73±0.46	6.53±0.67	4.95±1.22	66.10±1.45	1.69±0.79	1.78±0.70	61.64±4.72	1.79±1.22	1.25±1.01
	FairSIN-G	85.57±1.08	6.57±0.29	5.55±0.84	68.22±0.39	2.56±0.60	1.69±1.29	65.73±1.76	3.53±1.20	2.42±1.43
	FairSIN-F	87.61±0.83	5.54±0.40	3.47±1.03	67.96±1.54	1.16±0.90	0.98±0.70	66.38±1.39	2.53±0.97	2.03±1.23
	FairSIN w/o N.	87.26±0.17	5.93±0.04	4.30±0.20	68.35±0.62	2.51±1.99	2.36±1.89	65.87±1.34	1.98±1.01	1.87±0.64
	FairSIN w/o D.	87.40±0.15	5.65±0.40	4.63±0.52	68.74±0.33	2.22±1.47	1.67±1.70	66.42±1.52	2.73±1.08	2.37±0.69
	FairSIN	**87.67±0.26**	**4.56±0.75**	**2.79±0.89**	**69.34±0.32**	**0.57±0.19**	**0.43±0.41**	**67.76±0.71**	**1.49±0.74**	**0.59±0.50**
GIN	vanilla	83.52±0.87	7.55±0.51	6.17±0.69	69.25±1.75	3.71±1.20	2.55±1.52	65.83±1.31	1.97±1.12	2.17±0.48
	FairGNN	77.90±2.21	6.33±1.49	4.74±1.64	67.10±3.25	3.82±2.44	3.62±2.78	66.49±1.54	3.53±3.90	3.17±3.52
	EDITS	73.74±5.12	6.71±2.35	5.98±3.66	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	74.46±9.98	5.57±1.11	**3.41±1.43**	66.37±1.51	3.84±1.05	3.24±1.60	65.57±1.34	2.70±1.28	3.23±1.92
	FairVGNN	83.86±1.57	5.67±0.76	5.77±0.76	68.37±0.97	1.88±0.99	1.24±1.06	65.46±1.22	1.45±1.13	1.21±1.06
	FairSIN-G	86.10±1.39	6.93±0.16	6.75±0.66	67.73±1.67	1.98±1.54	1.50±1.15	65.09±2.69	1.55±1.23	1.74±0.80
	FairSIN-F	86.48±0.75	5.95±1.85	5.97±2.07	68.92±1.08	1.51±1.11	**0.82±0.79**	65.97±0.82	1.45±1.15	1.14±0.73
	FairSIN w/o N.	85.27±0.70	7.21±0.39	6.75±0.55	68.92±1.13	2.81±1.91	2.12±1.30	65.04±1.56	2.19±1.96	1.23±0.92
	FairSIN w/o D.	86.44±0.80	4.38±1.48	4.23±1.88	**70.04±0.80**	2.44±1.50	1.63±1.24	65.58±0.71	1.40±0.67	1.12±0.24
	FairSIN	**86.52±0.48**	**4.35±0.71**	4.17±0.96	69.58±0.57	**1.11±0.31**	0.97±0.59	**66.74±1.56**	**0.64±0.47**	**1.01±0.64**
SAGE	vanilla	88.13±1.12	1.13±0.48	2.61±1.16	69.03±0.77	3.09±1.29	2.21±1.60	66.55±0.69	4.71±1.05	2.72±0.85
	FairGNN	87.68±0.73	1.94±0.82	1.72±0.70	67.03±2.61	2.97±1.28	2.06±3.02	67.68±1.49	2.86±1.39	2.30±1.33
	EDITS	84.42±2.87	3.74±3.54	4.46±3.50	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	84.11±5.49	5.74±0.38	4.07±1.28	68.48±1.11	3.84±1.05	3.90±2.18	66.68±1.45	6.75±1.84	8.15±0.97
	FairVGNN	88.41±1.29	1.14±0.67	1.69±1.13	68.50±0.71	1.12±0.98	1.13±1.02	66.39±1.95	4.15±1.30	2.31±1.57
	FairSIN-G	**88.79±1.08**	3.97±0.92	1.70±0.66	69.11±0.62	2.00±1.13	1.66±0.70	66.19±1.49	4.96±0.25	2.90±1.21
	FairSIN-F	88.51±0.16	0.67±0.33	1.85±0.50	69.28±0.98	1.80±0.46	1.62±0.84	66.99±1.06	3.25±1.00	1.89±0.79
	FairSIN w/o N.	87.70±0.28	0.64±0.40	2.21±0.22	68.77±0.62	2.35±0.99	1.71±0.99	67.39±1.05	2.92±1.69	1.79±1.16
	FairSIN w/o D.	88.46±0.19	0.82±0.51	2.12±0.55	**69.65±0.32**	1.91±0.82	1.09±1.12	66.78±0.83	3.92±1.02	1.62±0.68
	FairSIN	88.74±0.42	**0.58±0.60**	**1.49±0.34**	69.12±1.16	**1.04±0.83**	**1.04±0.42**	**67.95±0.79**	**1.74±0.73**	**0.68±0.65**

Table 2: Comparison among SOTA methods and different variants of FairSIN. (Bold: the best; underline: the runner-up.)

Main Results

❖ Effectiveness of Data-centric Variant FairSIN-G and FairSIN-F

- Both FairSIN-G and FairSIN-F maintain the accuracy and improve the fairness on average
  - Demonstrate idea of sensitive information neutralization
- FairSIN-G only amplifies the weights of existing heterogeneous neighbors
  - Limit its capacity to furnish as extensive information as FairSIN-F
- FairSIN-F offers a cost-effective, model-agnostic and task-irrelevant solution for fair node representation learning

Encoder	Method	Bail			Pokec_n			Pokec_z		
		ACC↑	DP↓	EO↓	ACC↑	DP↓	EO↓	ACC↑	DP↓	EO↓
GCN	vanilla	87.55±0.54	6.85±0.47	5.26±0.78	68.55±0.51	3.75±0.94	2.93±1.15	66.78±1.09	3.95±1.03	2.76±0.95
	FairGNN	82.94±1.67	6.90±0.17	4.65±0.14	67.36±2.06	3.29±2.95	2.46±2.64	67.65±1.65	1.87±1.95	1.32±1.42
	EDITS	84.49±2.27	6.64±0.39	7.51±1.20	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	82.36±3.91	5.78±1.29	4.72±1.08	67.24±0.49	1.22±0.94	2.79±1.24	66.74±0.93	6.50±2.16	7.64±1.77
	FairVGNN	84.73±0.46	6.53±0.67	4.95±1.22	66.10±1.45	1.69±0.79	1.78±0.70	61.64±4.72	1.79±1.22	1.25±1.01
	FairSIN-G	85.57±1.08	6.57±0.29	5.55±0.84	68.22±0.39	2.56±0.60	1.69±1.29	65.73±1.76	3.53±1.20	2.42±1.43
	FairSIN-F	87.61±0.83	5.54±0.40	3.47±1.03	67.96±1.54	1.16±0.90	0.98±0.70	66.38±1.39	2.53±0.97	2.03±1.23
	FairSIN w/o N.	87.26±0.17	5.93±0.04	4.30±0.20	68.35±0.62	2.51±1.99	2.36±1.89	65.87±1.34	1.98±1.01	1.87±0.64
	FairSIN w/o D.	87.40±0.15	5.65±0.40	4.63±0.52	68.74±0.33	2.22±1.47	1.67±1.70	66.42±1.52	2.73±1.08	2.37±0.69
	FairSIN	**87.67±0.26**	**4.56±0.75**	**2.79±0.89**	**69.34±0.32**	**0.57±0.19**	**0.43±0.41**	**67.76±0.71**	**1.49±0.74**	**0.59±0.50**
GIN	vanilla	83.52±0.87	7.55±0.51	6.17±0.69	69.25±1.75	3.71±1.20	2.55±1.52	65.83±1.31	1.97±1.12	2.17±0.48
	FairGNN	77.90±2.21	6.33±1.49	4.74±1.64	67.10±3.25	3.82±2.44	3.62±2.78	66.49±1.54	3.53±3.90	3.17±3.52
	EDITS	73.74±5.12	6.71±2.35	5.98±3.66	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	74.46±9.98	5.57±1.11	**3.41±1.43**	66.37±1.51	3.84±1.05	3.24±1.60	65.57±1.34	2.70±1.28	3.23±1.92
	FairVGNN	83.86±1.57	5.67±0.76	5.77±0.76	68.37±0.97	1.88±0.99	1.24±1.06	65.46±1.22	1.45±1.13	1.21±1.06
	FairSIN-G	86.10±1.39	6.93±0.16	6.75±0.66	67.73±1.67	1.98±1.54	1.50±1.15	65.09±2.69	1.55±1.23	1.74±0.80
	FairSIN-F	86.48±0.75	5.95±1.85	5.97±2.07	68.92±1.08	1.51±1.11	**0.82±0.79**	65.97±0.82	1.45±1.15	1.14±0.73
	FairSIN w/o N.	85.27±0.70	7.21±0.39	6.75±0.55	68.92±1.13	2.81±1.91	2.12±1.30	65.04±1.56	2.19±1.96	1.23±0.92
	FairSIN w/o D.	86.44±0.80	4.38±1.48	4.23±1.88	**70.04±0.80**	2.44±1.50	1.63±1.24	65.58±0.71	1.40±0.67	1.12±0.24
	FairSIN	**86.52±0.48**	**4.35±0.71**	4.17±0.96	69.58±0.57	**1.11±0.31**	0.97±0.59	**66.74±1.56**	**0.64±0.47**	**1.01±0.64**
SAGE	vanilla	88.13±1.12	1.13±0.48	2.61±1.16	69.03±0.77	3.09±1.29	2.21±1.60	66.55±0.69	4.71±1.05	2.72±0.85
	FairGNN	87.68±0.73	1.94±0.82	1.72±0.70	67.03±2.61	2.97±1.28	2.06±3.02	67.68±1.49	2.86±1.39	2.30±1.33
	EDITS	84.42±2.87	3.74±3.54	4.46±3.50	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	84.11±5.49	5.74±0.38	4.07±1.28	68.48±1.11	3.84±1.05	3.90±2.18	66.68±1.45	6.75±1.84	8.15±0.97
	FairVGNN	88.41±1.29	1.14±0.67	1.69±1.13	68.50±0.71	1.12±0.98	1.13±1.02	66.39±1.95	4.15±1.30	2.31±1.57
	FairSIN-G	**88.79±1.08**	3.97±0.92	1.70±0.66	69.11±0.62	2.00±1.13	1.66±0.70	66.19±1.49	4.96±0.25	2.90±1.21
	FairSIN-F	88.51±0.16	0.67±0.33	1.85±0.50	69.28±0.98	1.80±0.46	1.62±0.84	66.99±1.06	3.25±1.00	1.89±0.79
	FairSIN w/o N.	87.70±0.28	0.64±0.40	2.21±0.22	68.77±0.62	2.35±0.99	1.71±0.99	67.39±1.05	2.92±1.69	1.79±1.16
	FairSIN w/o D.	88.46±0.19	0.82±0.51	2.12±0.55	**69.65±0.32**	1.91±0.82	1.09±1.12	66.78±0.83	3.92±1.02	1.62±0.68
	FairSIN	88.74±0.42	**0.58±0.60**	**1.49±0.34**	69.12±1.16	**1.04±0.83**	**1.04±0.42**	**67.95±0.79**	**1.74±0.73**	**0.68±0.65**

Table 2: Comparison among SOTA methods and different variants of FairSIN. (Bold: the best; underline: the runner-up.)

Ablation Study

❖ Two ablated models:

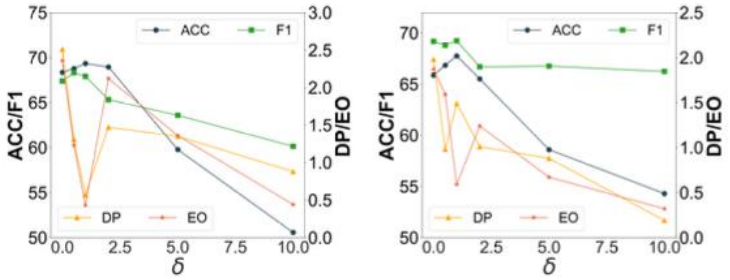
- ❑ FairSIN w/o D. – without discriminator
  - ➢ The neutralization of F3 alone already achieve a favorable trade-off between fairness and accuracy metrics
- ❑ FairSIN w/o N. - FairSIN where $\delta = 0$
  - ➢ Discriminator is employed in isolation rather than as a constraint to guide the learning of F3
  - ➢ It often leads to a decrease in predictive precision

Encoder	Method	Bail			Pokec_n			Pokec_z		
		ACC↑	DP↓	EO↓	ACC↑	DP↓	EO↓	ACC↑	DP↓	EO↓
GCN	vanilla	87.55±0.54	6.85±0.47	5.26±0.78	68.55±0.51	3.75±0.94	2.93±1.15	66.78±1.09	3.95±1.03	2.76±0.95
	FairGNN	82.94±1.67	6.90±0.17	4.65±0.14	67.36±2.06	3.29±2.95	2.46±2.64	67.65±1.65	1.87±1.95	1.32±1.42
	EDITS	84.49±2.27	6.64±0.39	7.51±1.20	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	82.36±3.91	5.78±1.29	4.72±1.08	67.24±0.49	1.22±0.94	2.79±1.24	66.74±0.93	6.50±2.16	7.64±1.77
	FairVGNN	84.73±0.46	6.53±0.67	4.95±1.22	66.10±1.45	1.69±0.79	1.78±0.70	61.64±4.72	1.79±1.22	1.25±1.01
	FairSIN-G	85.57±1.08	6.57±0.29	5.55±0.84	68.22±0.39	2.56±0.60	1.69±1.29	65.73±1.76	3.53±1.20	2.42±1.43
	FairSIN-F	87.61±0.83	5.54±0.40	3.47±1.03	67.96±1.54	1.16±0.90	0.98±0.70	66.38±1.39	2.53±0.97	2.03±1.23
	FairSIN w/o N.	87.26±0.17	5.93±0.04	4.30±0.20	68.35±0.62	2.51±1.99	2.36±1.89	65.87±1.34	1.98±1.01	1.87±0.64
	FairSIN w/o D.	87.40±0.15	5.65±0.40	4.63±0.52	68.74±0.33	2.22±1.47	1.67±1.70	66.42±1.52	2.73±1.08	2.37±0.69
	FairSIN	**87.67±0.26**	**4.56±0.75**	**2.79±0.89**	**69.34±0.32**	**0.57±0.19**	**0.43±0.41**	**67.76±0.71**	**1.49±0.74**	**0.59±0.50**

Hyper-parameter Analysis

❖ Value of δ

- ❑ Control the amount of introduced heterogeneous information
- ❑ Too large may lead to sensitive information leakage in an opposite direction
  - ➢ Contribute to a decrease in predictive performance
- ❑ An optimal value δ= 1 ➔ a favorable trade-off between predictive performance and fairness



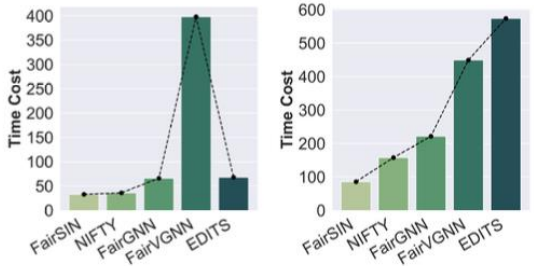
(a) Pokec-n (b) Pokec-z

Figure 4: Classification performance and group fairness under different values of hyper-parameter δ.

Efficiency Analysis

❖ Training time cost

- ❑ FairSIN has the lowest time cost among all methods
- ❑ Both efficient and effective, enabling potential applications in various scenarios
- ❑ FairVGNN - its large number of parameters and the process of adversarial training
- ❑ EDITS - needs to model node similarities between all node pairs for edge addition



(a) Bail (b) Credit

Figure 5: Training time cost on Bail and Credit with GCN backbone (in seconds).

Conclusion

❖ Most of GM algorithms lack of fairness consideration

❖ Fairness Notation about Group Fairness

- ❑ Demographic Parity (DP), Equality of Odds, Equality of Opportunity (EO)

❖ Problem of Fairness in Graph Mining

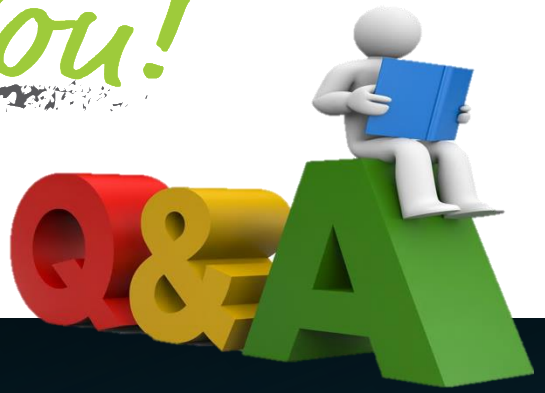
- ❑ Predictions based on node embeddings learned by GNNs can be unfair
- ❑ (1) The raw features of nodes could be statistically correlated to the sensitive attribute
  - ➢ Lead to sensitive information leakage in encoded representations
- ❑ (2) Homophily effects: nodes with the same sensitive attribute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

Conclusion (Cont'd)

❖ FairSIN

- ❑ Previous Works:
  - ➢ Filter out sensitive information from inputs or representations, e.g., edge dropping or feature masking
  - ➢ Such filtering-based strategies may also filter out some non-sensitive feature information
  - ➢ Leading to a sub-optimal trade-off between predictive performance and fairness
- ❑ Proposed Method:
  - ➢ The core idea is to introduce extra Fairness-facilitating Features (F3) to node features or representations
  - ➢ The sensitive biases (+/- symbols) can be neutralized
  - ➢ F3 provide additional non-sensitive feature information (dot symbols)
  - ➢ Thus, enabling a better trade-off between predictive performance and fairness

Thank You!



HTET ARKAR
hak3601@gmail.com