Fairness-aware GNN Link Prediction

AIM5056_41 Machine learning with Graphs

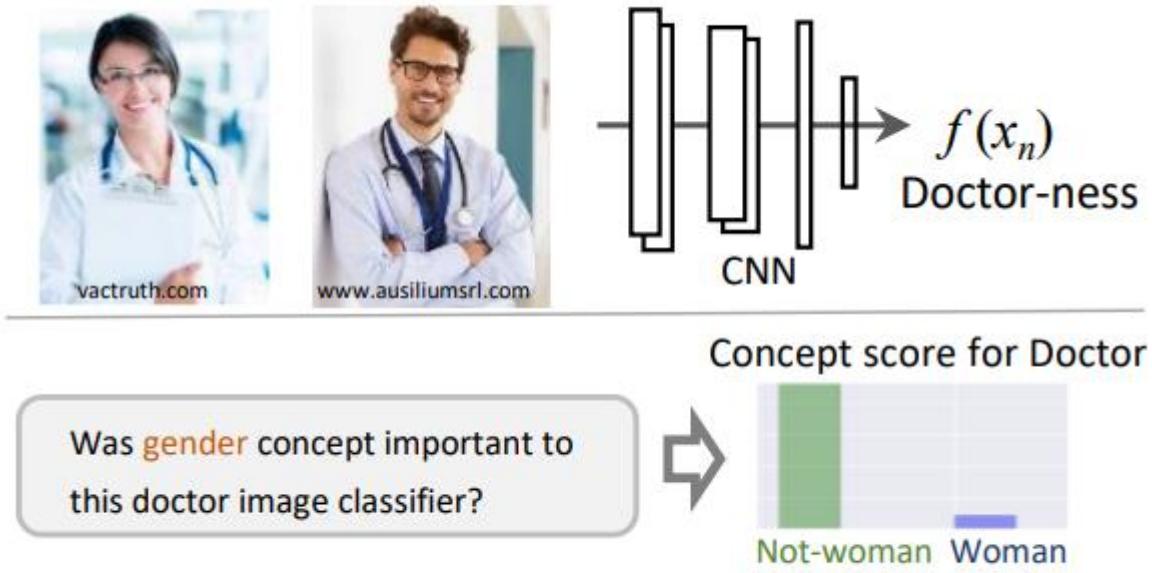
Gahyung Kim, Suhyun Yoon, Yonghoon Kang, Heeyoon Yang, Jiyoung Lim

2021/11/28

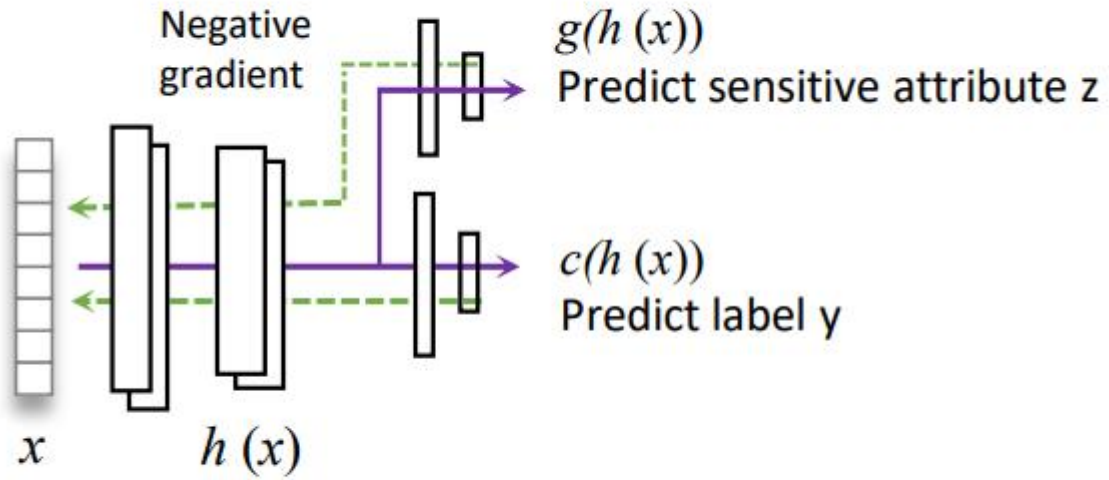
Index

- **Introduction**
  - ✓ What is the sensitivity attribute?
- **Related works**
  - ✓ A recent study on fairness graph link prediction
- **Proposed Method**
  - ✓ Step 1. Fair drop
  - ✓ Step 2. Adj. norm
  - ✓ Step 3. prediction flip
- **Experiment**
- **Conclusion**

What is the sensitive attribute?



(a) Bias Detection



(b) Bias Mitigation

Du, Mengnan, et al. "Fairness in deep learning: A computational perspective." *IEEE Intelligent Systems* (2020).

What is the sensitive attribute?

Bias in real case

NEWS Home > Technology > Machine Learning

AI recruitment tool pulled by Amazon for sex bias

Recruitment software was not favourable towards women

by: Rene Millman 10 Oct 2018

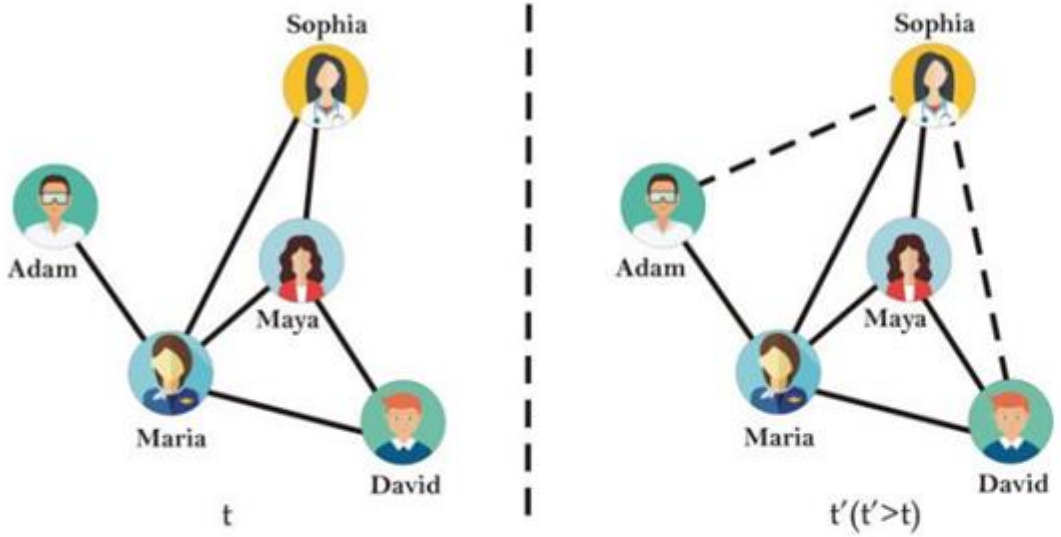
The AI tech was first developed at Amazon in 2014 as a way of quickly filtering out most candidates and providing the firm with the top five people. However, by 2015, it was realised that the system was not rating applicants in a gender-neutral way.

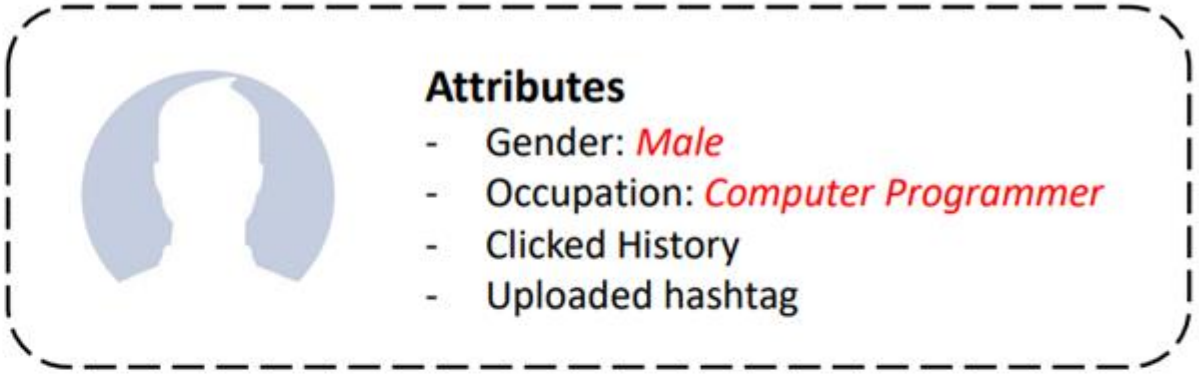
The problem lay in how the system was trained. It was fed CVs to detect patterns in recruiting over a 10-year period. Most of the applications came from men. According to a report from *Reuters*, the system taught itself that male candidates were preferable to women. **It downgraded CVs if found words such as "women's" and penalised graduates of all-female colleges.**

While Amazon recoded the software to make the AI neutral to these terms, this did not guarantee that the technology would find other methods of being discriminatory against women, the report said.

Problem Definition

- **Fairness-aware GNN Link Prediction**





→ *Gender and Occupation are Sensitive attributes*

- **Problem of current Link Prediction**
  Current Graph Learning algorithm promotes links that lead to increased segregation with *intrinsic bias*

- **Definition of Fairness**

$$P(\hat{Y}=1 \mid X^{(S)}) = P(\hat{Y}=1)$$

Related works

ON DYADIC FAIRNESS: Exploring and Mitigating Bias in Graph Connections, ICLR 2021

Algorithm 1: Algorithmic routine for **FairAdj**

Input: vertex features X, adjacency matrix A, GNNs parameters θ, learning rates η_θ and $\eta_{\widetilde{A}}$

Normalize adjacency matrix $\widetilde{A} \leftarrow D^{-1}A$

Fix the elements with zero in $\widetilde{A}$ and select the non-zero elements for optimization

while θ *or* $\widetilde{A}$ *has not converged* **do**

  for $t = 1$ **to** T_1 ▷ `optimize for utility` **do**

    Compute $\mathcal{L}_{\text{util}}$ by Eq. (5), $g_\theta \leftarrow \nabla_\theta \mathcal{L}_{\text{util}}$, $\theta \leftarrow \theta + \eta_\theta \cdot \text{Adam}(\theta, g_\theta)$

  for $t = 1$ **to** T_2 ▷ `optimize for fairness` **do**

    $Z \leftarrow \text{GNN}_\theta(X, \widetilde{A})$, $\hat{A} \leftarrow ZZ^\top$ ▷ *reconstruct graph connections*

    Compute $\mathcal{L}_{\text{fair}}$ by Eq. (6), $g_{\widetilde{A}} \leftarrow \nabla_{\widetilde{A}} \mathcal{L}_{\text{fair}}$

    for $v = 1$ **to** N ▷ `projected gradient descent` **do**

      Sort n non-zero elements in $[\widetilde{A} - \eta_{\widetilde{A}} g_{\widetilde{A}}]_{v,*}$ in descending order: $e_1 \geq e_2 \geq \cdots e_n$

      $\gamma \leftarrow \sum_{j=1}^{n} \mathbf{1}(e_j + \frac{1}{j}(1 - \sum_{i=1}^{j} e_i) \geq 0)$ ▷ $\mathbf{1}(\cdot)$:*the indicator function*

      $\beta \leftarrow \frac{1}{\rho}(1 - \sum_{i=1}^{\gamma} e_i)$

      for $u = 1$ **to** n ▷ *update* $\widetilde{A}$ **do**

        $[\widetilde{A}]_{v,u} \leftarrow \max\{[\widetilde{A} - \eta_{\widetilde{A}} g_{\widetilde{A}}]_{v,u} + \beta,\ 0\}$

Output: Link predictive score between vertex v and $u \leftarrow \text{sigmoid}(\text{GNN}_\theta(v, \widetilde{A})^\top \text{GNN}_\theta(u, \widetilde{A}))$

$$\max_{\theta} \quad \mathcal{L}_{\text{util}} := \mathbb{E}_{\text{GNN}_\theta(Z|X,\widetilde{A})}[\log p(A \mid Z)] - KL[\text{GNN}_\theta(Z \mid X, \widetilde{A}) \| \mathcal{N}(0,1)]. \tag{5}$$

KL-Divergence:
Punishes the discrepancy between latent distribution and a Gaussian prior

Intra links *Inter links*

$$\min_{\widetilde{A}} \quad \mathcal{L}_{\text{fair}} := \|\mathbb{E}_{v,u\sim U\times U}[\hat{a}_{vu} \mid S(v) = S(u)] - \mathbb{E}_{v,u\sim U\times U}[\hat{a}_{vu} \mid S(v) \neq S(u)]\|^2, \tag{6}$$

$$\text{s.t.} \quad (1).\ [\widetilde{A}]_{vu} = 0, \text{ if } [A]_{vu} = 0, \quad (2).\ \widetilde{A}\mathbb{1} = \mathbb{1},\ \widetilde{A} \geq 0,$$

Restrictions

Related works

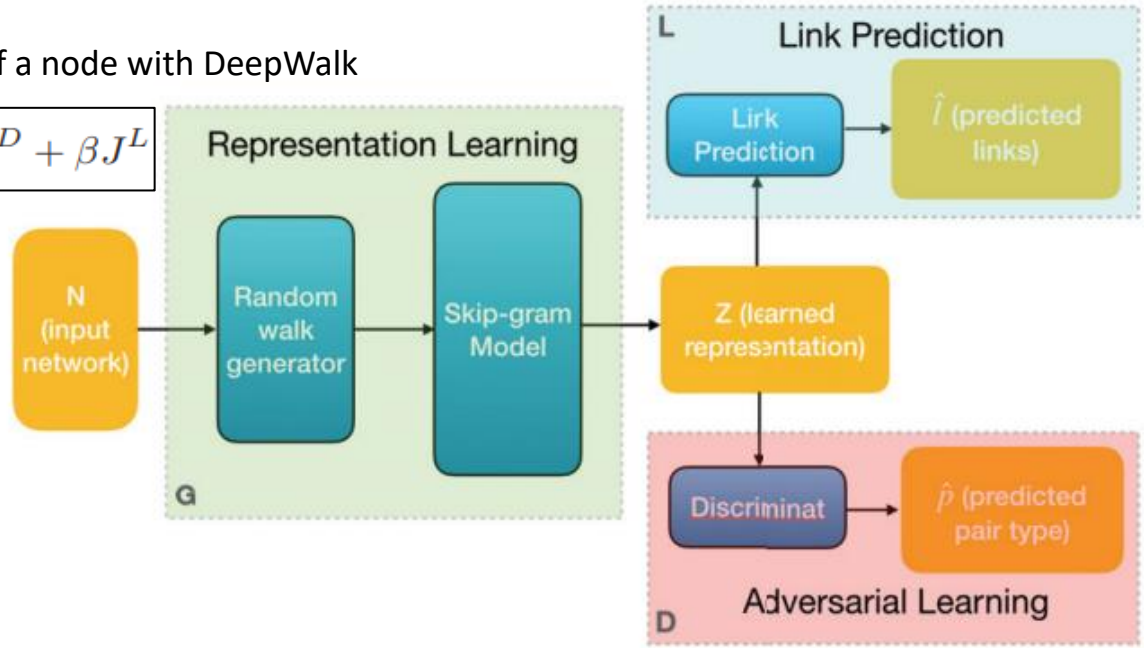
Link Prediction

Predict whether or not a link exists by looking at the representations of two given nodes

Generator

Learning the representation of a node with DeepWalk

$$J^G = (1-\alpha)J^{Skip} - \alpha J^D + \beta J^L$$



$$J^L = -\frac{1}{|\mathcal{T}|}\sum_{(u,v)\in\mathcal{T}}\left[e_{uv}log(\hat{e}_{uv}) + (1-e_{uv})\log(1-\hat{e}_{uv})\right]$$

$$J^D = -\frac{1}{|\mathcal{T}|}\sum_{(u,v)\in\mathcal{T}}\left[p_{uv}log(\hat{p}_{uv}) + (1-p_{uv})\log(1-\hat{p}_{uv})\right]$$

Discriminator

Given two node embeddings, predict whether these two nodes are intra-group or inter-group (making the generator more robust)

Related works

Biased Edge Dropout for Enhancing Fairness in Graph Representation Learning, *arXiv (2021)*

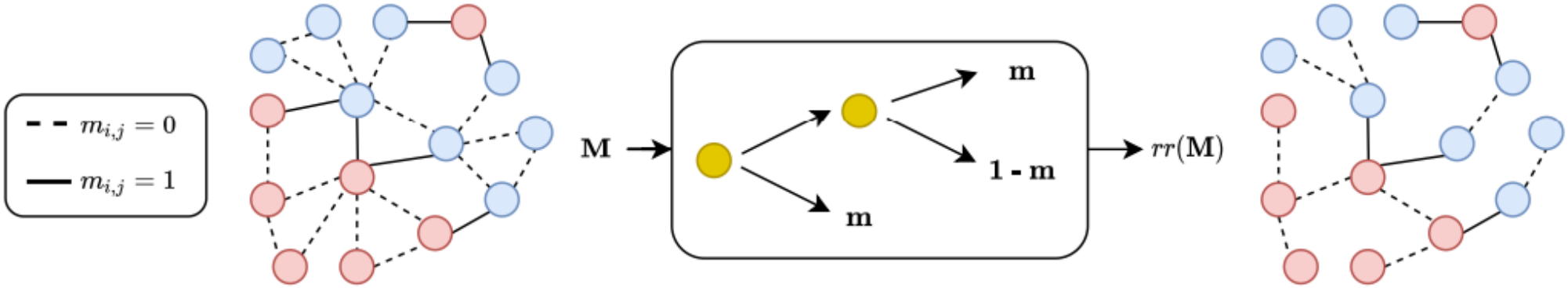


Fig. 1. Schematics of FairDrop. From left to right: (i) Construction of the sensitive attribute homophily mask (Section IV-A); (ii) Randomized response used to inject randomness and to regulate the bias imposed by the constraint (Section IV-B); (iii) Final graph obtained by dropping the connections according to the randomized mask (Section IV-C).

$$rr(m_{ij}) = \begin{cases} m_{ij} & \text{with probability : } \frac{1}{2} + \delta \\ 1 - m_{ij} & \text{with probability : } \frac{1}{2} - \delta \end{cases}$$

$$\mathbf{A}_{fair} = \mathbf{A} \circ rr(\mathbf{M})$$

Limitations of Recent Studies

- Conducted experiments under different environment set up
  - ✓ Metrics
  - ✓ Questionable choice on sensitive attribute selection
  - ✓ Arbitrary selection of datasets

	Metric	N2VEC	FAIRWALK	N2VEC^{EMD}	N2VEC^{LAP}	CNE	DEBAYES	CNE^{EMD}	CNE^{LAP}	RANDOM
POLBLOGS	AUC	**.75 ± .01**	**.75 ± .01**	.66 ± .01	.73 ± .01	**.93 ± .01**	.88 ± .01	.86 ± .01	.91 ± .02	.53 ± .01
	RB	.97 ± .01	.96 ± .01	**.78 ± .01**	.94 ± .01	.97 ± .01	**.64 ± .04**	.73 ± .03	.94 ± .04	.63 ± .01
	DI	.10 ± .02	.20 ± .01	.54 ± .07	.25 ± .02	.03 ± .02	.53 ± .05	**.83 ± .05**	.19 ± .03	.43 ± .02
	Cons.	.75 ± .02	.73 ± .01	.77 ± .10	**.91 ± .01**	.89 ± .01	.89 ± .01	.90 ± .01	**.93 ± .01**	.90 ± .04
FACEBOOK	AUC	**.98 ± .01**	.85 ± .00	.96 ± .00	.96 ± .00	**.99 ± .01**	**.99 ± .03**	**.99 ± .01**	.98 ± .01	.49 ± .04
	RB	.64 ± .01	**.61 ± .01**	**.61 ± .00**	.63 ± .00	.58 ± .02	.57 ± .02	**.54 ± .03**	.58 ± .02	.56 ± .02
	DI	.80 ± .01	**.83 ± .00**	.80 ± .01	.80 ± .00	.93 ± .03	.91 ± .03	.98 ± .01	**.99 ± .05**	.84 ± .02
	Cons.	.96 ± .00	.94 ± .00	.96 ± .01	.96 ± .00	**.97 ± .01**	.96 ± .00	**.97 ± .01**	**.97 ± .00**	.89 ± .01
DBLP	AUC	**.98 ± .01**	**.98 ± .01**	.78 ± .03	.81 ± .04	**.98 ± .01**	**.98 ± .01**	.77 ± .03	.82 ± .05	.54 ± .01
	RB	.77 ± .00	.77 ± .01	**.58 ± .04**	**.58 ± .02**	.55 ± .02	**.51 ± .02**	.52 ± .01	**.51 ± .02**	.59 ± .01
	DI	.14 ± .01	.14 ± .01	**1.26 ± .04**	1.02 ± .05	.03 ± .01	.04 ± .01	**1.29 ± .04**	.98 ± .05	.43 ± .03
	Cons.	.91 ± .01	.91 ± .01	.93 ± .02	**.95 ± .01**	.91 ± .01	.90 ± .02	.94 ± .01	**.97 ± .01**	.86 ± .01

Table 1: Statistic for datasets in experiments.

Dataset	# Vertex	# Edge	# Class	# Intra	# Inter	Intra Ratio	Inter Ratio	Dis. Ratio
Oklahoma97	3,111	73,230	2	46,368	26,862	1.92e-2	1.11e-2	1.73
UNC28	4,018	65,287	2	36,212	29,075	8.76e-3	7.38e-3	1.19
Facebook#1684	786	14,024	2	7,989	6,035	4.76e-2	4.30e-2	1.11
Cora	2,708	5,278	7	4,275	1,003	6.51e-3	3.30e-4	19.73
Citeseer	3,312	4,660	6	2,089	2,571	2.13e-3	5.70e-4	3.74
Pubmed	19,717	44,327	3	33,443	10,884	4.80e-4	9.00e-5	5.33

Table 2: Experimental results on *UNC28*.

Method	AUC ↑	AP ↑	Δ_{DP} ↓	Δ_{true} ↓	Δ_{false} ↓	Δ_{FNR} ↓	Δ_{TNR} ↓
VGAE	**87.63** ± 0.56	**88.69** ± 0.65	2.24 ± 0.42	1.50 ± 0.41	0.44 ± 0.36	7.62 ± 0.84	2.18 ± 0.72
node2vec	87.22 ± 0.30	87.10 ± 0.37	2.75 ± 0.78	1.30 ± 0.53	1.05 ± 0.93	12.56 ± 1.12	2.24 ± 0.92
Fairwalk	87.18 ± 0.30	87.07 ± 0.37	2.79 ± 0.70	1.17 ± 0.49	0.90 ± 0.92	12.71 ± 1.11	2.20 ± 0.96
FairAdj$_{T2=5}$	86.98 ± 0.54	87.75 ± 0.65	**1.53** ± 0.35	**0.32** ± 0.29	**0.41** ± 0.35	2.84 ± 0.74	2.22 ± 0.68
FairAdj$_{T2=20}$	87.04 ± 0.55	87.80 ± 0.65	1.57 ± 0.36	0.34 ± 0.31	0.42 ± 0.35	**2.76** ± 0.75	**2.16** ± 0.73

TABLE II
LINK PREDICTION ON CORA

Method	Accuracy ↑	AUC ↑	ΔDP_m ↓	ΔEO_m ↓	ΔDP_g ↓	ΔEO_g ↓	ΔDP_s ↓	ΔEO_s ↓
GCN+EdgeDrop	**82.4** ± 0.9	**90.1** ± 0.7	56.4 ± 2.4	36.5 ± 4.3	12.3 ± 2.6	15.4 ± 3.3	90.2 ± 2.7	100.0 ± 0.0
GAT+EdgeDrop	80.5 ± 1.2	88.3 ± 0.8	53.7 ± 2.5	37.1 ± 3.2	18.8 ± 3.6	22.5 ± 4.2	93.6 ± 2.9	100.0 ± 0.0
GCN+FairDrop	**82.4** ± 0.9	**90.1** ± 0.7	52.9 ± 2.5	**31.0** ± 4.9	**11.8** ± 3.2	**14.9** ± 3.7	**89.4** ± 3.4	100.0 ± 0.0
GAT+FairDrop	79.2 ± 1.2	87.8 ± 1.0	**48.9** ± 2.8	31.9 ± 4.3	15.3 ± 3.2	18.1 ± 3.5	94.5 ± 2.0	100.0 ± 0.0

Project Goals

Goal

The **goal** of this project is to **evaluate** and **compare** the proposed methods using the **same datasets** and a fixed number of evaluation methods.

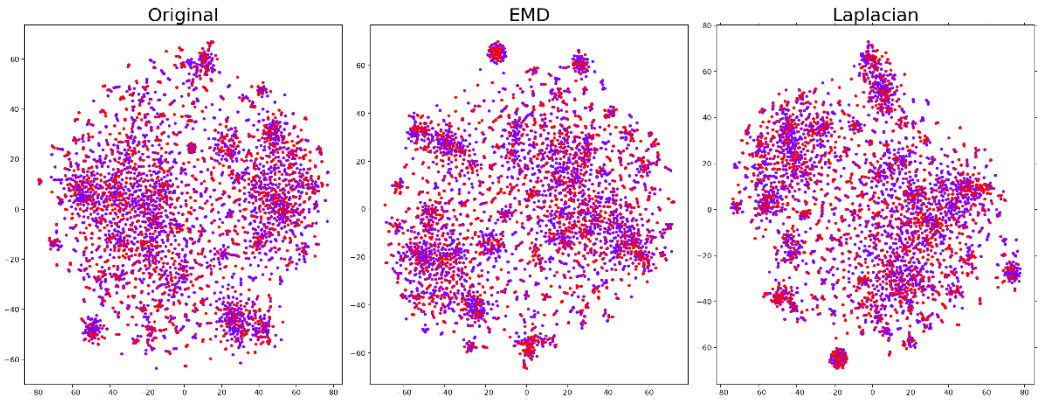
Contribution

1. Evaluate the effectiveness of proposed **fairness** evaluating Metrics, such as **Modred** [2], and examine its applicability to other models
2. By constructing the comparison table of recently proposed methods, we are going to make a foundation study to establish future research directions on fairness-aware link prediction

Limitations of Recent Studies

- Conducted experiments under different environment set up
  - ✓ Metrics
  - ✓ Questionable choice on sensitive attribute selection
  - ✓ Arbitrary selection of datasets

$$\hat{h} = \frac{1}{C-1} \sum_{k=0}^{C-1} \left[h_k - \frac{|C_k|}{n} \right]_+$$



Twitch Node embeddings

Table 5: Statistics for homophilic graph datasets. # C is the number of node classes.

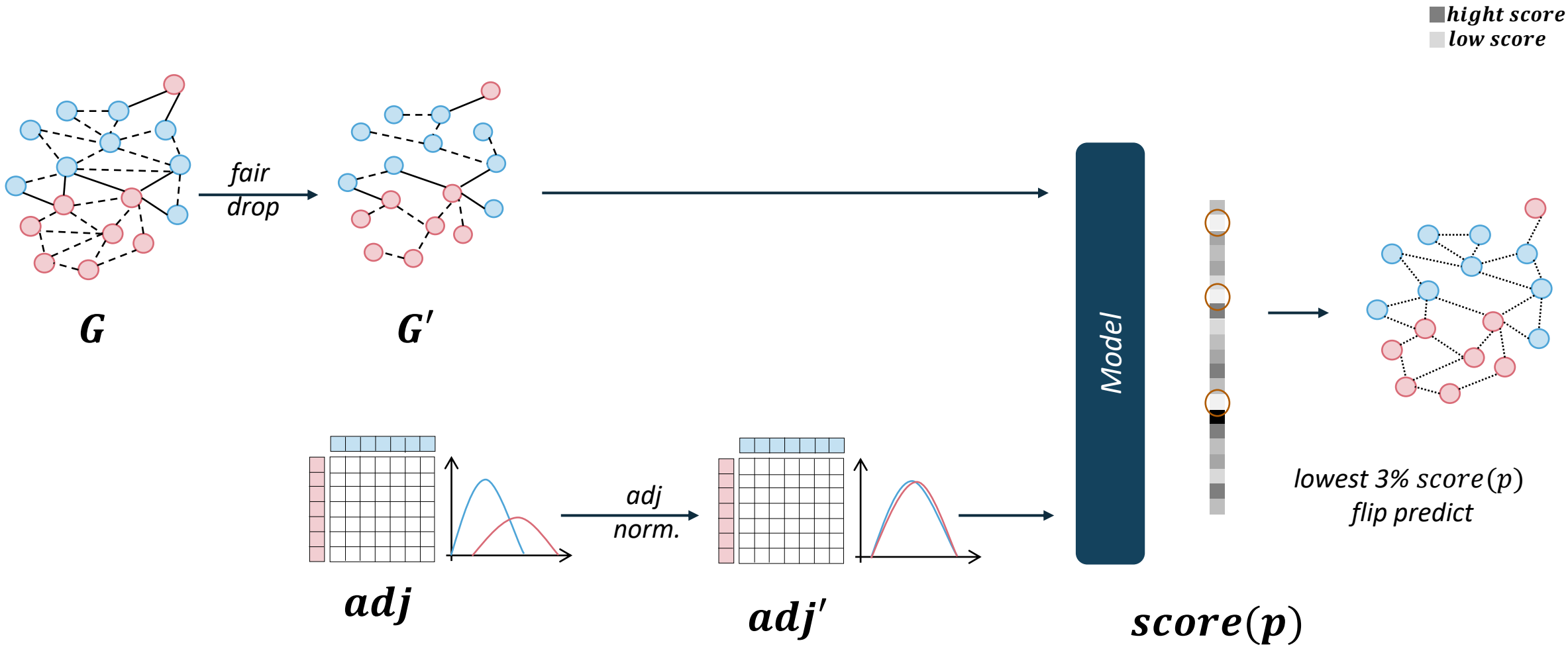
Dataset	# Nodes	# Edges	# C	Edge hom.	$\hat{h}$ (ours)
Cora	2,708	5,278	7	.81	.766
Citeseer	3,327	4,552	6	.74	.627
Pubmed	19,717	44,324	3	.80	.664
ogbn-arXiv	169,343	1,166,243	40	.66	.416
ogbn-products	2,449,029	61,859,140	47	.81	.459
oeis	226,282	761,687	5	.50	.532

Table 2: Statistics of our proposed non-homophilous graph datasets. # C is the number of distinct node classes. Note that our datasets come from more diverse applications areas and are much larger than those shown in Table 1, with up to 384x more nodes and 1398x more edges.

Dataset	# Nodes	# Edges	# Feat.	# C	Class types	Edge hom.	$\hat{h}$ (ours)
Penn94	41,554	1,362,229	5	2	gender	.470	.046
pokec	1,632,803	30,622,564	65	2	gender	.445	.000
arXiv-year	169,343	1,166,243	128	5	pub year	.222	.272
snap-patents	**2,923,922**	13,975,788	269	5	time granted	.073	.100
genius	421,961	984,979	12	2	marked act.	.618	.080
twitch-gamers	168,114	6,797,557	7	2	mature content	.545	.090
wiki	1,925,342	**303,434,860**	600	5	views	.389	.107

Lim, Derek, et al. "Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods." Advances in Neural Information Processing Systems 34 (2021).

Proposed Method



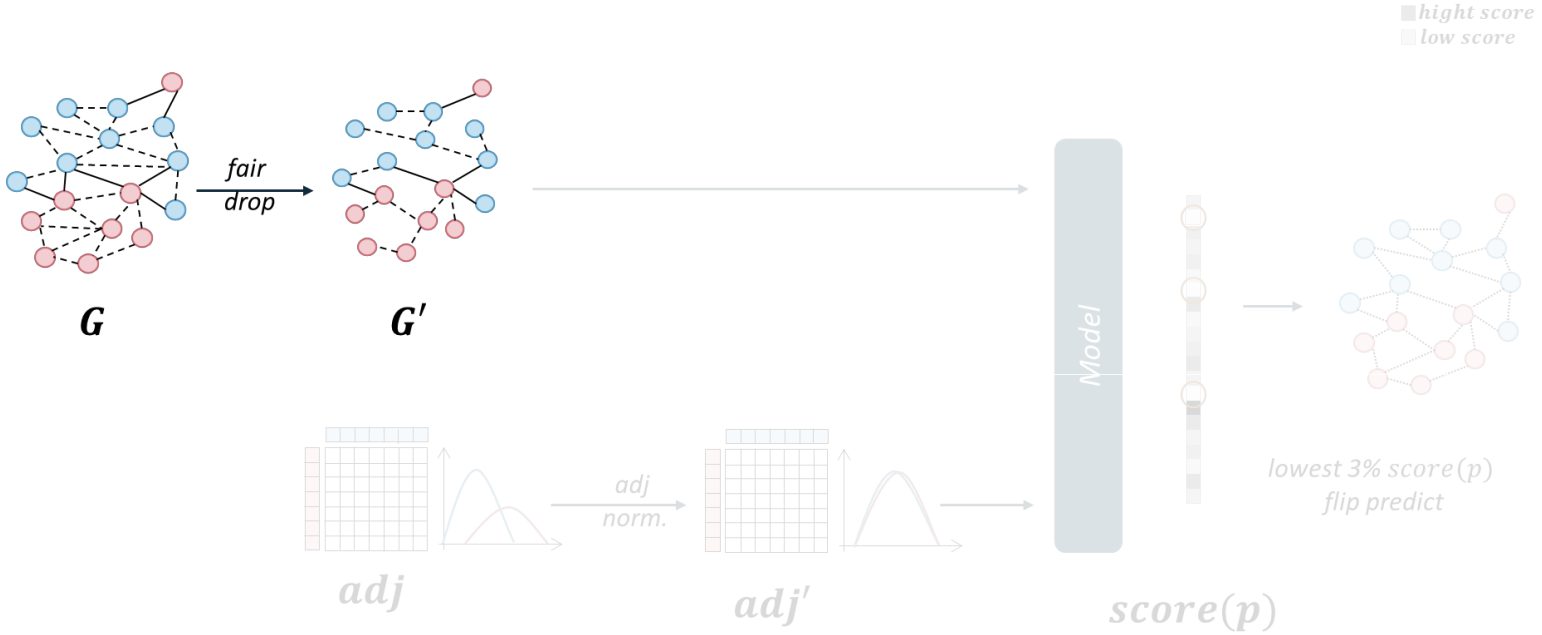
Proposed Method

1) Fairly drop edges

Algorithm 1: Fairness-aware GNN

Input: vertex features X, adjacency matrix A, Graph G, initialized with zeros M, $\delta \in [0, \frac{1}{2}]$

for edges (i, j) **in** G **do**
  $m_{ij} = \mathrm{S}[i] \neq S[j]$
  if RANDOM $\leq \frac{1}{2} + \delta$ **then**
    $fd(m_{ij}) = m_{ij}$
  else
    $fd(m_{ij}) = 1 - m_{ij}$
  end if
end for
$A_{fair} = A \circ fd(M)$
$A_{fairnorm} = D^{-1} A_{fair}$
for epoch 1, 2, 3, ... e **do**
  $P \leftarrow \mathrm{GNN}(X, A_{fairnorm})$
end for
$score \leftarrow \frac{(-1)^{\gamma(P)}}{2m}(-1 + \frac{d_x + d_y - 1}{2m})\gamma(X_x^{(p)}, X_y^{(p)}) + (\sum_{v \in V, X_v^{(p)} \neq X_x^{(p)}, v \neq y} d_v + \sum_{v \in V, X_v^{(p)} \neq X_x^{(p)}, v \neq x} d_i)/4m^2$
$s_{sort} \leftarrow$ flip the edges with the $\lambda score$
for s **in** s_{sort} **do**
  if $s \geq 0.5$ **then**
    $\hat{P}_s \leftarrow 0$
  else
    $\hat{P}_s \leftarrow 1$
  end if
end for
$R \leftarrow EvaluationMetric(\hat{P})$



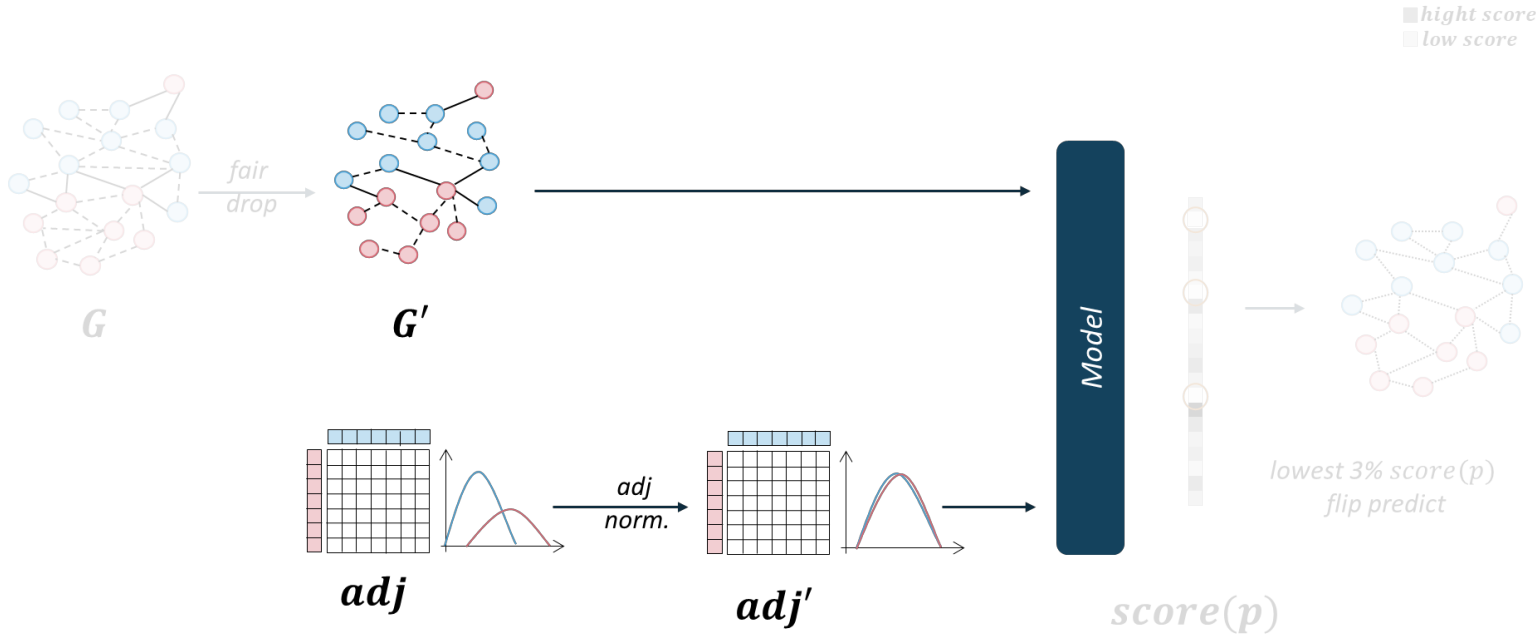
Proposed Method

2) Learning Adjacency Matrix

Algorithm 1: Fairness-aware GNN

Input: vertex features X, adjacency matrix A, Graph G, initialized with zeros M, $\delta \in [0, \frac{1}{2}]$

```
for edges (i, j) in G do
    m_ij = S[i] ≠ S[j]
    if RANDOM ≤ 1/2 + δ then
        fd(m_ij) = m_ij
    else
        fd(m_ij) = 1 − m_ij
    end if
end for
A_fair = A ∘ fd(M)
A_fairnorm = D^{-1} A_fair
for epoch 1, 2, 3, ... e do
    P ← GNN(X, A_fairnorm)
end for
score ← ((−1)^{f(p)} / 2m)(−1 + (d_x + d_y − 1) / 2m) γ(X_x^(p), X_y^(p)) +
    (Σ_{v∈V, X_v^(p) ≠ X_x^(p), v≠y} d_v + Σ_{v∈V, X_v^(p) ≠ X_x^(p), v≠x} d_i) / 4m^2
s_sort ← flip the edges with the λscore
for s in s_sort do
    if s ≥ 0.5 then
        P̂_s ← 0
    else
        P̂_s ← 1
    end if
end for
R ← EvaluationMetric(P̂)
```



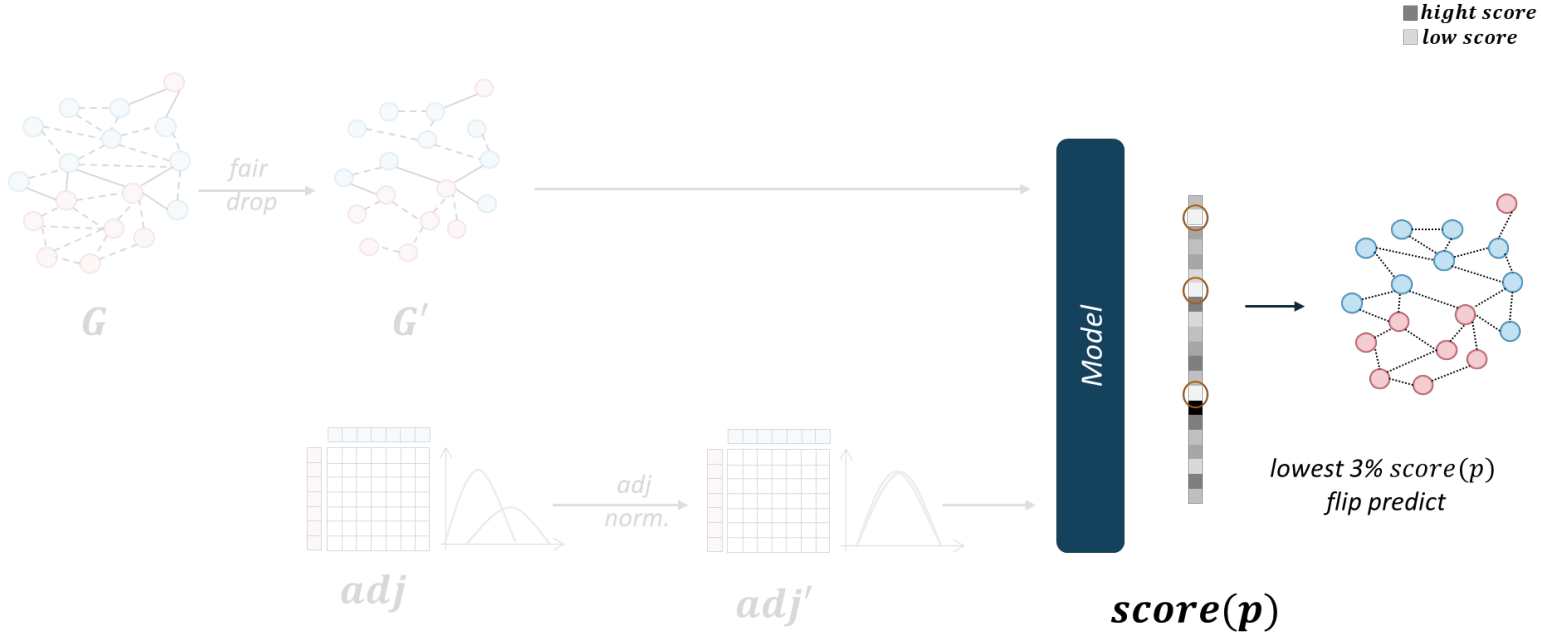
Proposed Method

3) Greedy Postprocessing

Algorithm 1: Fairness-aware GNN

Input: vertex features X, adjacency matrix A, Graph G, initialized with zeros M, $\delta \in [0, \frac{1}{2}]$

1: **for** edges (i, j) **in** G **do**
2: $m_{ij} = S[i] \neq S[j]$
3: **if** RANDOM $\leq \frac{1}{2} + \delta$ **then**
4: $fd(m_{ij}) = m_{ij}$
5: **else**
6: $fd(m_{ij}) = 1 - m_{ij}$
7: **end if**
8: **end for**
9: $A_{fair} = A \circ fd(M)$
10: $A_{fairnorm} = D^{-1} A_{fair}$
11: **for** epoch 1, 2, 3, ... e **do**
12: $P \leftarrow \text{GNN}(X, A_{fairnorm})$
13: **end for**
14: $score \leftarrow \frac{(-1)^{\gamma(P)}}{2m}(-1 + \frac{d_x + d_y - 1}{2m})\gamma(X_x^{(p)}, X_y^{(p)}) + (\sum_{v \in V, X_v^{(p)} \neq X_x^{(p)}, v \neq y} d_v + \sum_{v \in V, X_v^{(p)} \neq X_x^{(p)}, v \neq x} d_i)/4m^2$
15: $s_{sort} \leftarrow$ flip the edges with the $\lambda score$
16: **for** s **in** s_{sort} **do**
17: **if** $s \geq 0.5$ **then**
18: $\hat{P}_s \leftarrow 0$
19: **else**
20: $\hat{P}_s \leftarrow 1$
21: **end if**
22: **end for**
23: $R \leftarrow EvaluationMetric(\hat{P})$



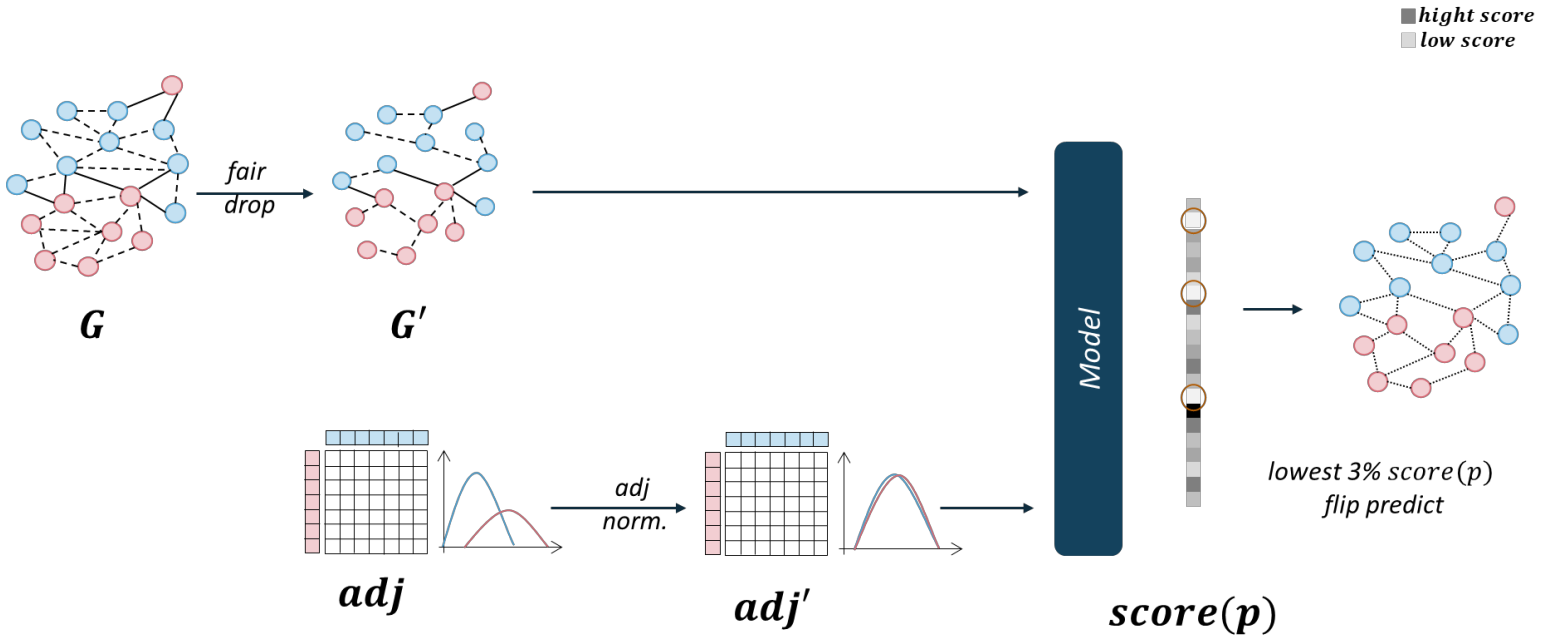
Proposed Method

Algorithm

Algorithm 1: Fairness-aware GNN

Input: vertex features X, adjacency matrix A, Graph G, initialized with zeros M, $\delta \in [0, \frac{1}{2}]$

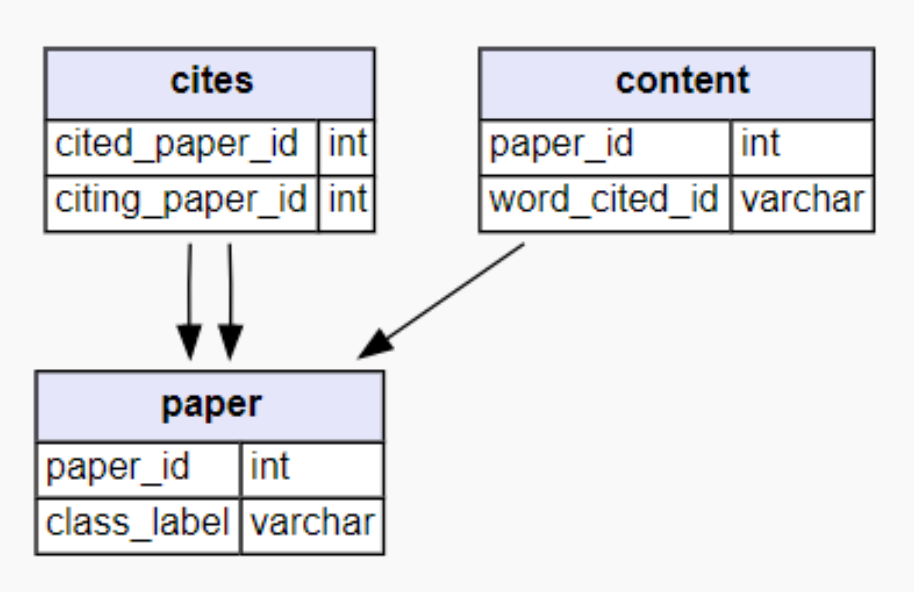
for edges (i, j) **in** G **do**
  m_{ij} =S$[i] \neq S[j]$
  if RANDOM $\leq \frac{1}{2} + \delta$ **then**
    $fd(m_{ij}) = m_{ij}$
  else
    $fd(m_{ij}) = 1 - m_{ij}$
  end if
end for
$A_{fair} = A \circ fd(M)$
$A_{fairnorm} = D^{-1} A_{fair}$
for epoch 1, 2, 3, ... e **do**
  $P \leftarrow \text{GNN}(X, A_{fairnorm})$
end for
$score \leftarrow \frac{(-1)^{\gamma(P)}}{2m}(-1 + \frac{d_x + d_y - 1}{2m})\gamma(X_x^{(p)}, X_y^{(p)}) + (\sum_{\substack{v \in V, X_v^{(p)} \neq X_x^{(p)} \\ v \neq y}} d_v + \sum_{\substack{v \in V, X_v^{(p)} \neq X_x^{(p)} \\ v \neq x}} d_i)/4m^2$
$s_{sort} \leftarrow$flip the edges with the $\lambda score$
for s **in** s_{sort} **do**
  if $s \geq 0.5$ **then**
    $\hat{P}_s \leftarrow 0$
  else
    $\hat{P}_s \leftarrow 1$
  end if
end for
$R \leftarrow EvaluationMetric(\hat{P})$

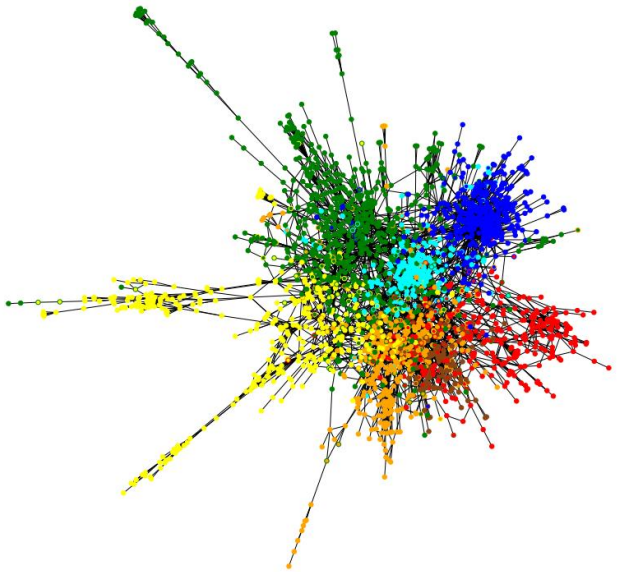


Datasets

Cora

The **Cora** dataset consists of **2,708 scientific publications** classified into one of **seven classes**. The citation network consists of **5,429 links**. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of **1,433 unique words.**

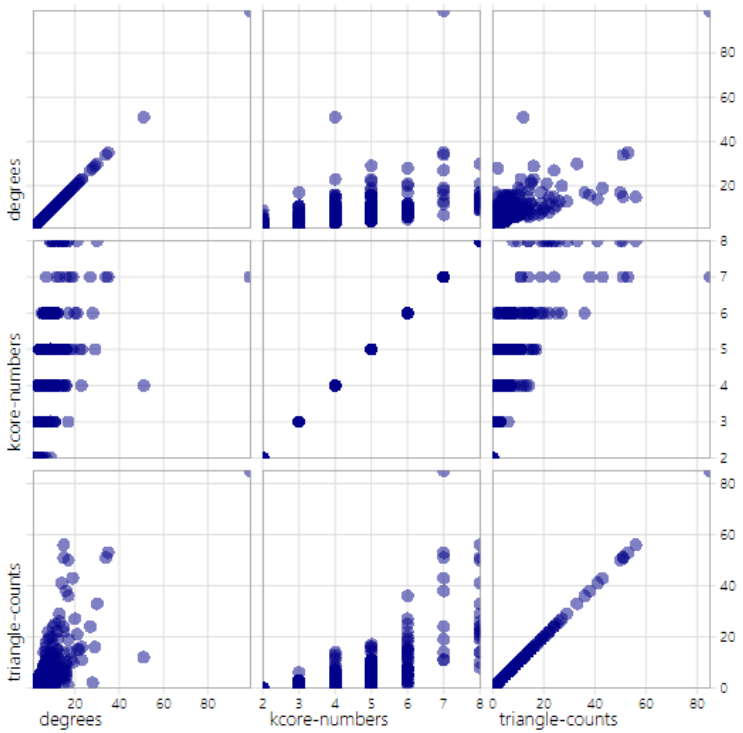


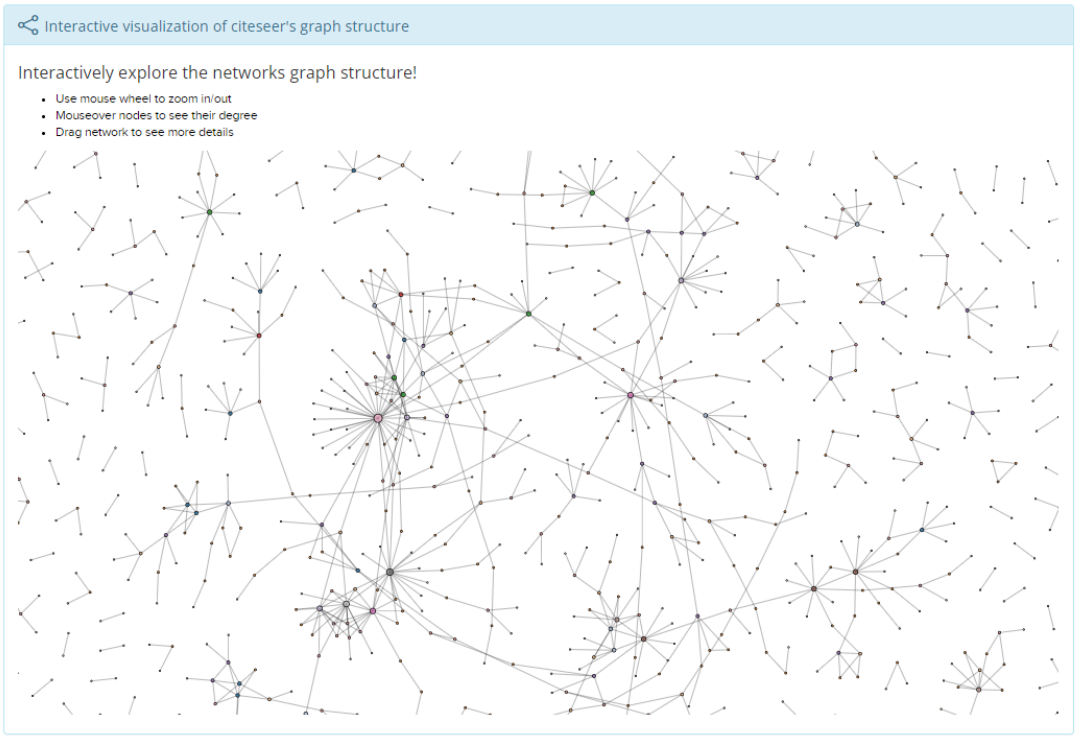


Datasets

Citeseer

The **CiteSeer** dataset consists of **3,327 scientific publications** classified into one of **six classes**, the citation network consists of **4,676 links**. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of **3703 unique words.**

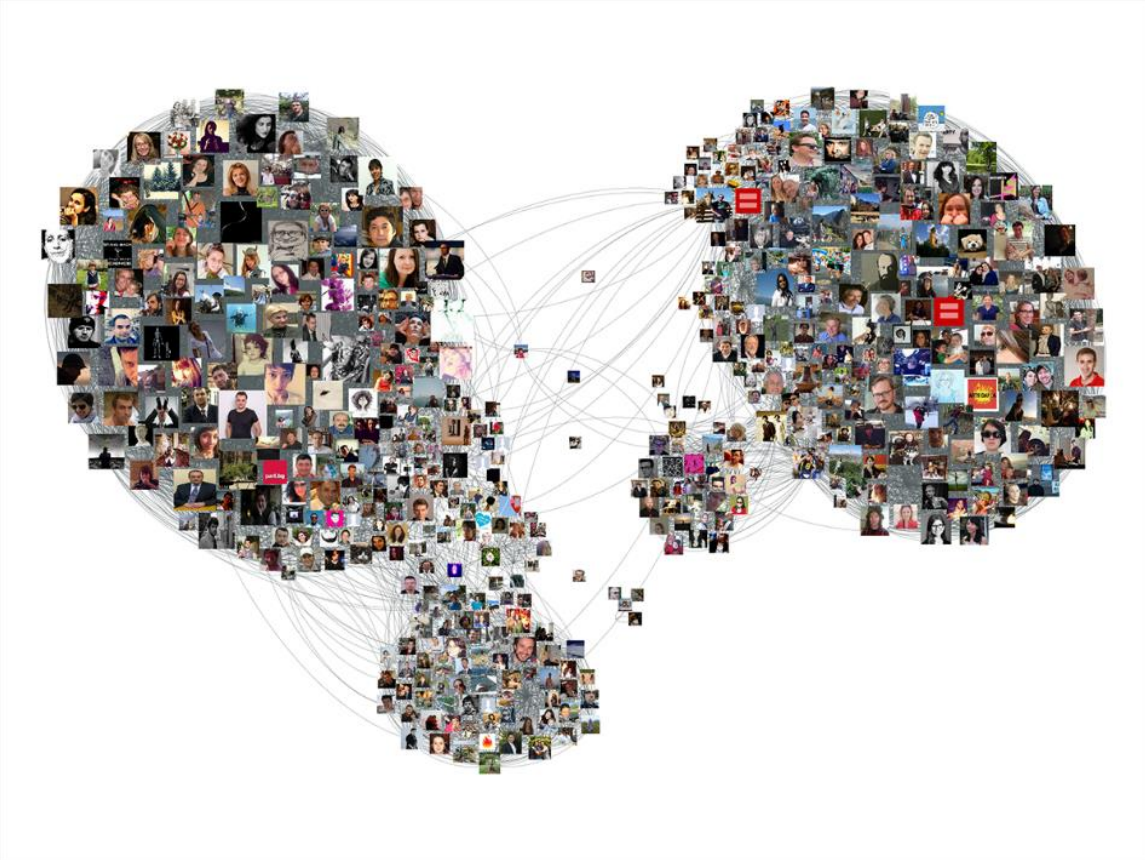




Datasets

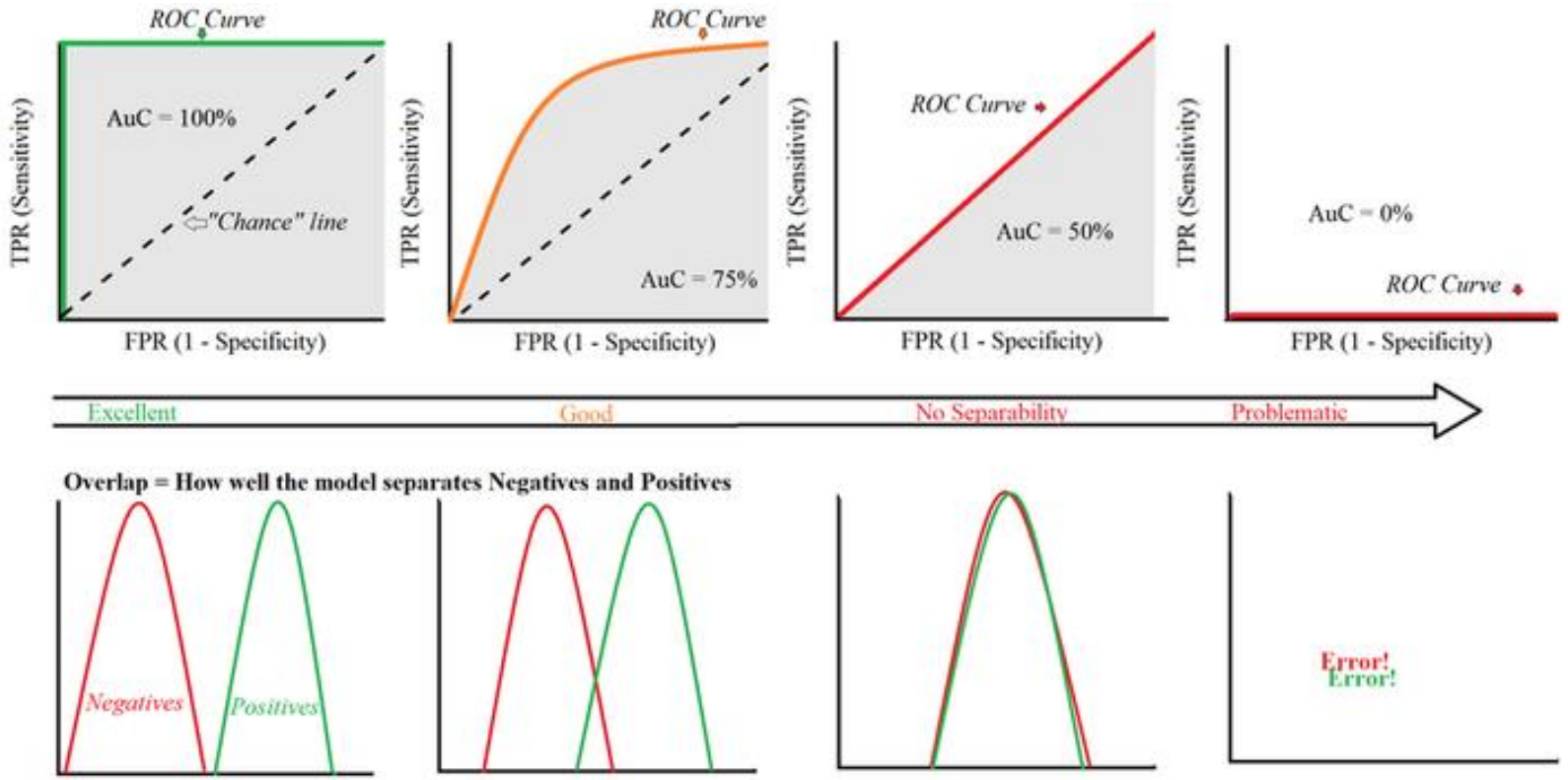
Facebook Ego

The **Facebook Ego** dataset consists of **4,039 nodes** and **79,411 edges**. Each node's features are based on the user's profile and the link represents the relationship in the social network



Metrics for Accuracy

- **Area Under the Curve (AUC) and AveragePrecision (AP)**



Metrics for Fairness

- $\boldsymbol{modred}$

  The reduction in the modularity measure to determine whether the modified network obtained from the link prediction results is biased towards creating more inter-group or intra-group links.

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{d_i d_j}{2m}\right)\delta(c_i, c_j) \qquad (1) \qquad modred = \frac{Q_{\text{ref}} - Q_{\text{pred}}}{Q_{\text{ref}}}, \qquad (2)$$

- $\boldsymbol{\Delta_{DP}}$ **: Demographic Parity**

  The demographic parity difference of 0 means that all groups have the same selection rate.

$$\Delta_{\text{DP}} := |\mathbb{E}_{(v,u)\sim U\times U}[g(v,u) \mid S(v) = S(u)] - \mathbb{E}_{(v,u)\sim U\times U}[g(v,u) \mid S(v) \neq S(u)]| \leq Q\|\Sigma\|_2 \cdot \delta. \qquad (2)$$

intra *inter*

- $\boldsymbol{\Delta_{True}}$, $\boldsymbol{\Delta_{False}}$

  Fairness is evaluated towards Δ_{DP}, as well as the disparity on the expected score on all the true samples Δ_{True} and false samples Δ_{False}

Experiments

Dataset	Method	AUC↑	AP↑	modred↑	Δ_{DP}↓	Δ_{TRUE}↓	Δ_{FALSE}↓
Cora	FairAdj	0.8491	0.8680	0.0147	0.1908	0.0437	0.0427
	FLIP	0.5619	0.5509	0.4458	0.0002	0.0000	0.0001
	GreedyPostProcessing	0.9726	0.9616	0.0030	0.1430	0.0084	0.0226
	FairDrop	0.6743	0.6317	0.0560	0.0140	0.0324	0.0278
	Ours	0.7057	0.6391	0.0693	0.0201	0.0248	0.0073
	Ours + Adversarial Network	0.7513	0.7435	0.0693	0.0231	0.0199	0.0079
Citeseer	FairAdj	0.7999	0.8387	0.0108	0.0854	0.0118	0.0023
	FLIP	0.5191	0.6707	0.2796	0.0004	0.0005	0.0000
	GreedyPostProcessing	0.9751	0.9812	0.0000	0.1061	0.0110	0.0143
	FairDrop	0.7418	0.6809	0.0398	0.3115	0.0742	0.0413
	Ours	0.7211	0.6492	0.0411	0.3058	0.1272	0.0873
	Ours + Adversarial Network	0.7646	0.7322	0.0533	0.2314	0.0932	0.0895
Facebook	FairAdj	0.9673	0.9587	0.0519	0.0068	0.0086	0.0025
	FLIP	0.7112	0.6519	0.3241	0.0004	0.0000	0.0002
	GreedyPostProcessing	0.9721	0.9472	0.0704	0.0017	0.0000	0.0003
	FairDrop	0.9598	0.9379	0.0371	0.0086	0.0080	0.0046
	Ours	0.7344	0.6532	0.0540	0.0192	0.0004	0.0010
	Ours + Adversarial Network	0.9255	0.9097	0.0457	0.0075	0.0061	0.0063

Reference

[1] Li, Peizhao, et al. "On dyadic fairness: Exploring and mitigating bias in graph connections." International Conference on Learning Representations. 2020.

[2] Masrour, Farzan, et al. "Bursting the filter bubble: Fairness-aware network link prediction." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 01. 2020.

[3] Laclau, Charlotte, et al. "All of the Fairness for Edge Prediction with Optimal Transport." International Conference on Artificial Intelligence and Statistics. PMLR, 2021.

[4] Indro Spinelli, et al. “Biased Edge Dropout for Enhancing Fairness in Graph Representation Learning”, arXiv:2104.14210, 2021.