UNIVERSITY of VIRGINIA

Individual Fairness for Graph Neural Networks: A Ranking based Approach

Yushun Dong
University of Virginia
06/20/2021

Outline

Background Introduction

Previous Works

Existing Problems & Challenges

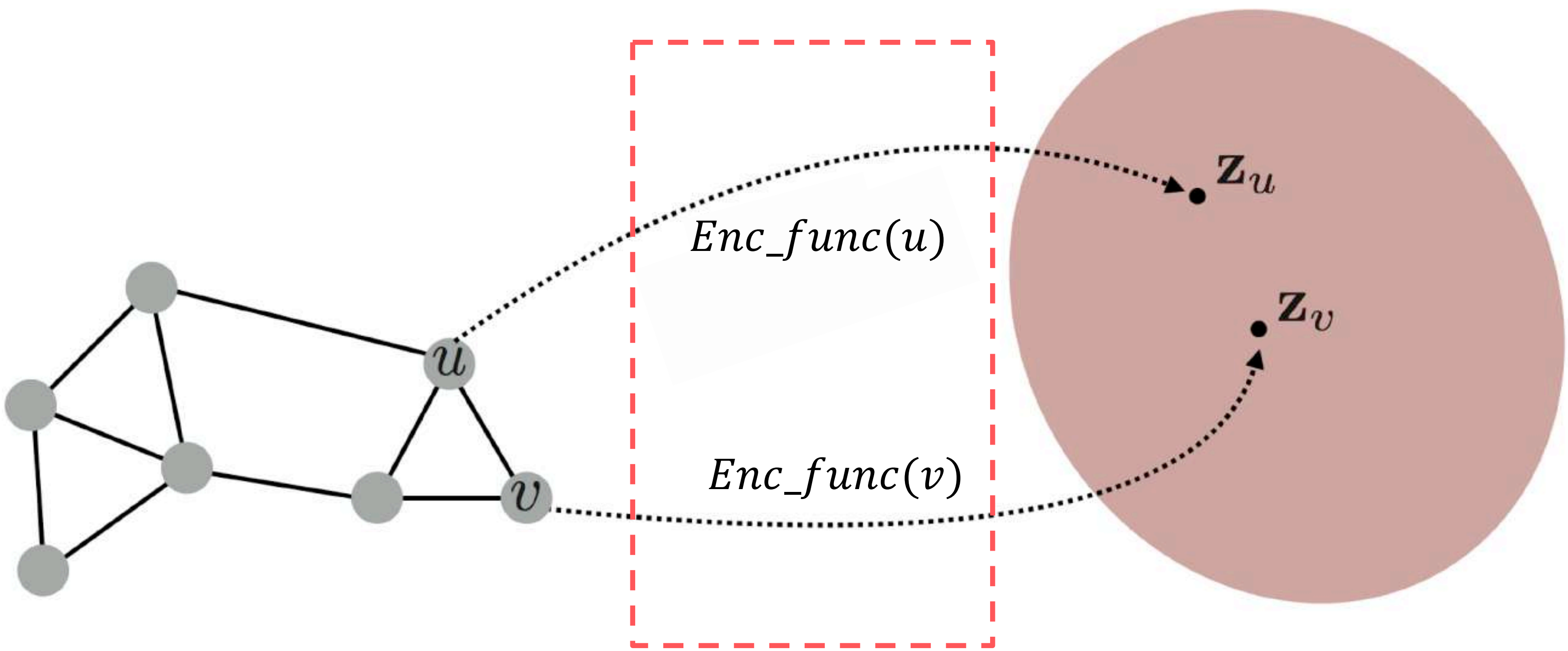
Our Solutions

Experiments & Conclusion

Future Works

Background Introduction: Graph Neural Networks

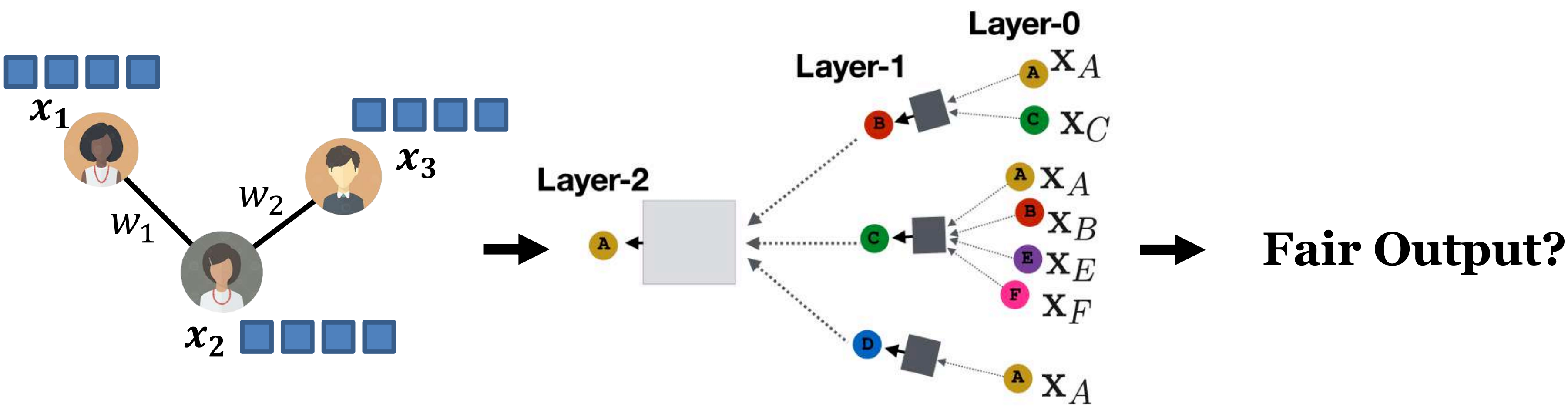
Goal of Graph Neural Networks (GNNs): to encode nodes so that similarity in the embedding space (e.g., dot product) approximates similarity in the original network.



GNNs work here

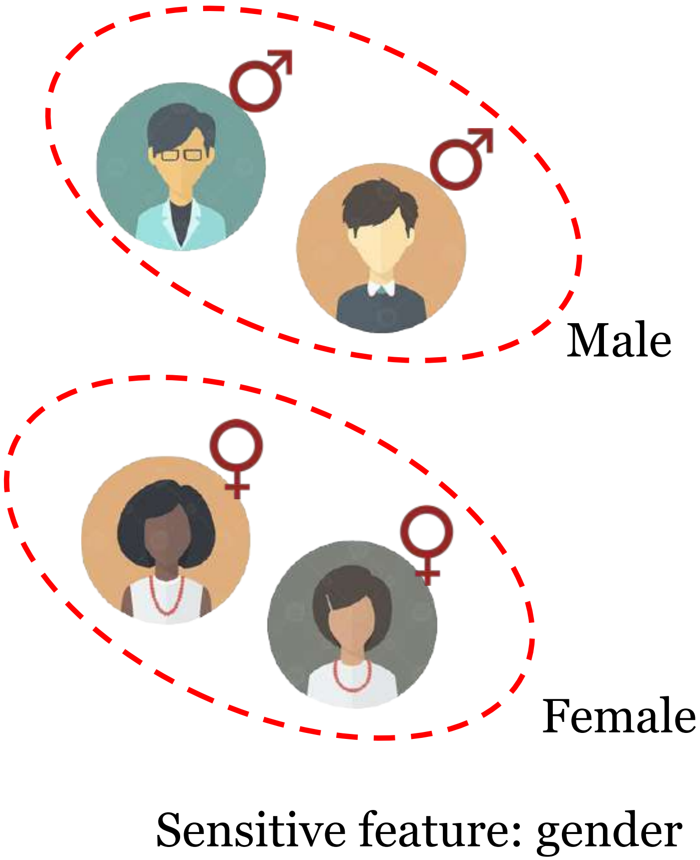
Background Introduction: Different Fairness Notions

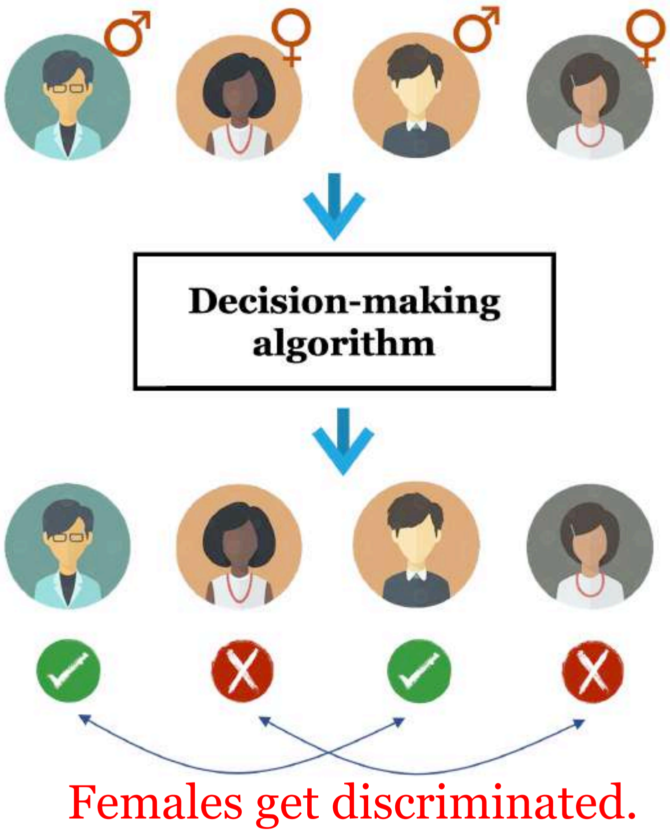
- Traditional GNNs usually lack fairness consideration.
- Introducing different fairness notions to graphs and promoting fairness for GNNs become urgent needs.
- So what are common fairness notions?



Different Fairness Notions: Group v.s. Individual

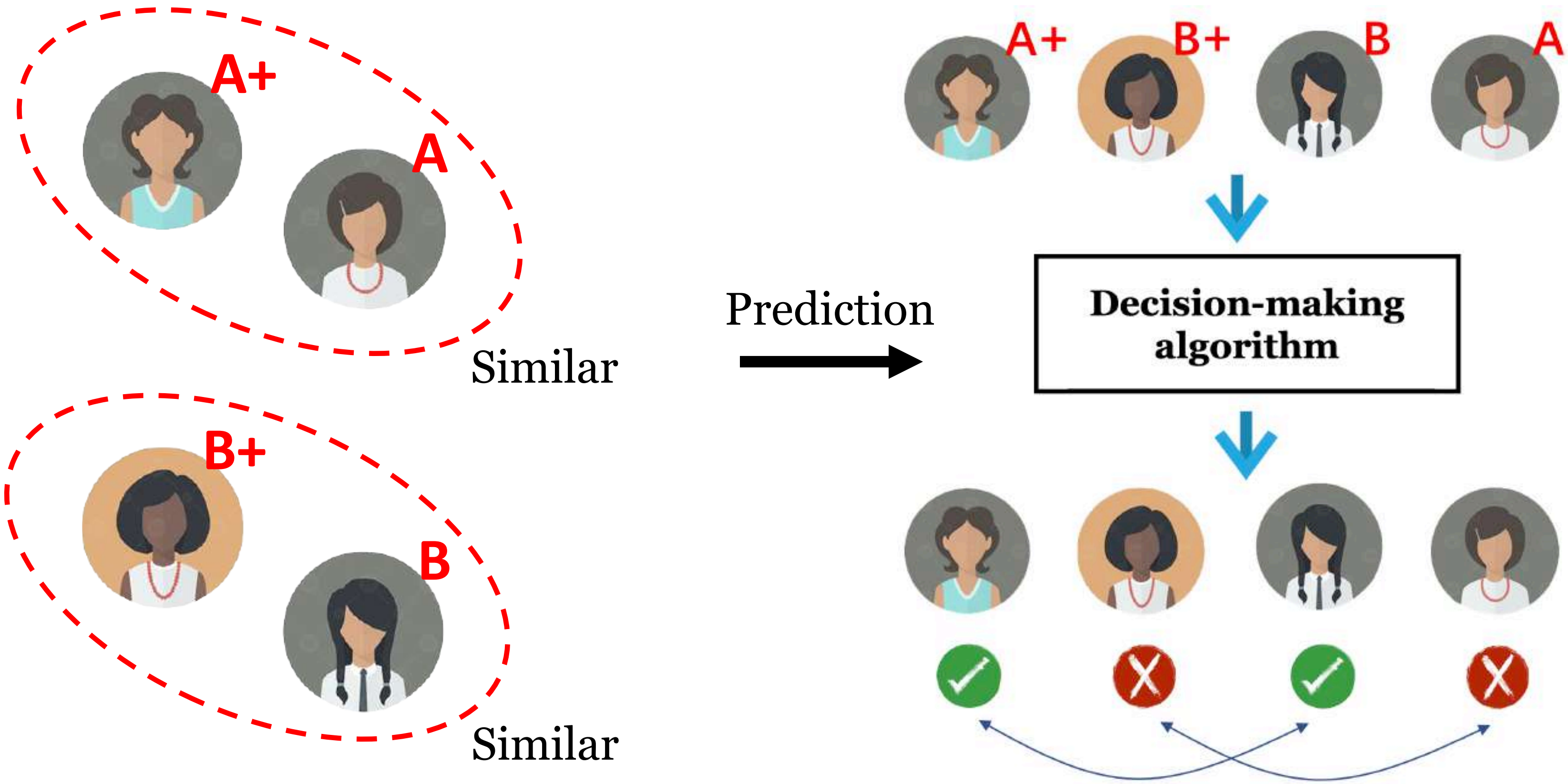
- **Group Fairness**
  - Decision-making algorithm should not make biased prediction towards people with certain sensitive feature.





Different Fairness Notions: Group v.s. Individual

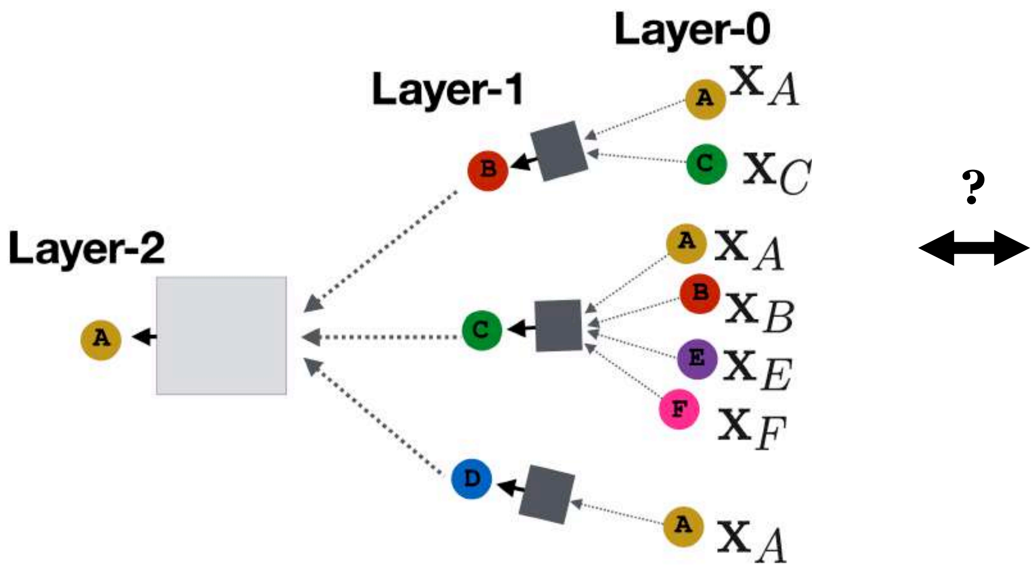
- **Individual Fairness: higher granularity**
  - Similar people should be treated similarly.

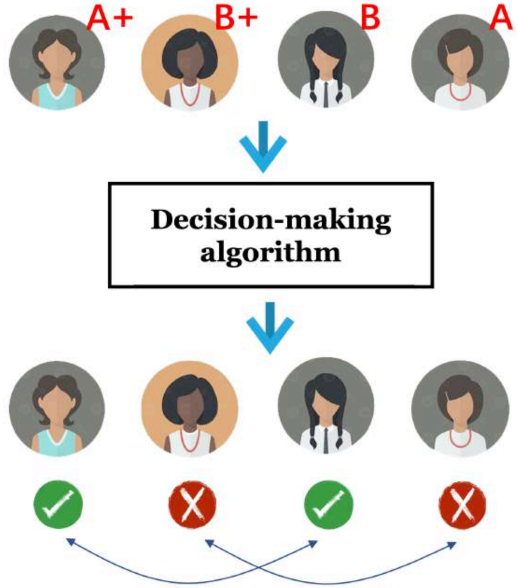


Similar pairs do not receive similar output.

Goal of this work

- Up to now, group fairness has been thoroughly explored in GNNs.
- As a more granular fairness notion, individual fairness has not been studied in GNNs.





Goal: Promoting individual fairness in GNNs.

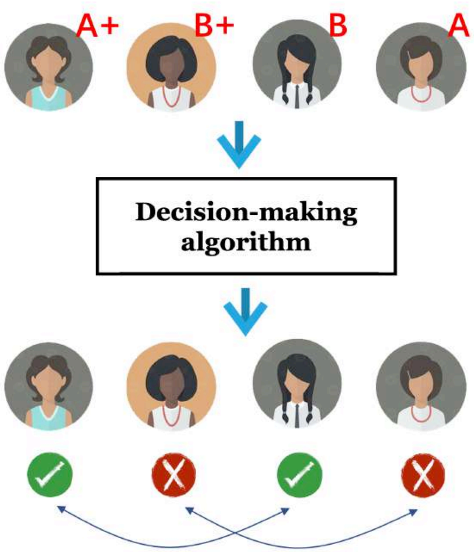
Outline

- Background Introduction
- Previous Works
- Existing Problems & Challenges
- Our Solutions
- Experiments & Conclusion
- Future Works

Previous works

- [Kang et al., 2020] formulate the individual fairness optimization problem in graphs based on *Lipschitz condition*.

- Output distance between pairs ≤ scaled input distance between pairs : similar people are treated similarly.



Lipschitz condition:

Lipschitz constant

$$D(u_1, u_4) \le Ld(u_1, u_4)$$

Output distance   Input distance

[Kang et al., 2020] Kang J, He J, Maciejewski R, et al. InFoRM: Individual Fairness on Graph Mining[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 379-389.

Previous works

- Here the oracle similarity matrix **S** is given as side information.
- Output distance : ℓ_2 distance between output vectors;
- Input distance: inverse of similarity between individual pairs;

Lipschitz condition:

Lipschitz constant

$$\|\mathrm{Y}[i,:]-\mathrm{Y}[j,:]\|_F^2 \leq \frac{\epsilon}{\mathrm{S}[i,j]}\ \forall i,j=1,\ldots,n \quad \longleftrightarrow \quad D(u_1, u_4) \leq L d(u_1, u_4)$$

Output distance

Input distance

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathrm{Y}[i,:]-\mathrm{Y}[j,:]\|_F^2\mathrm{S}[i,j]=2\mathrm{Tr}(\mathrm{Y}'\mathrm{L}_{\mathrm{S}}\mathrm{Y})$$

Outline

Background Introduction

Previous Works

Existing Problems & Challenges

Our Solutions

Experiments & Conclusion

Future Works

Existing Problems & Challenges

- **Constraint Formulation.**

Lipschitz constant

$$D(u_1, u_4) \leq L d(u_1, u_4)$$

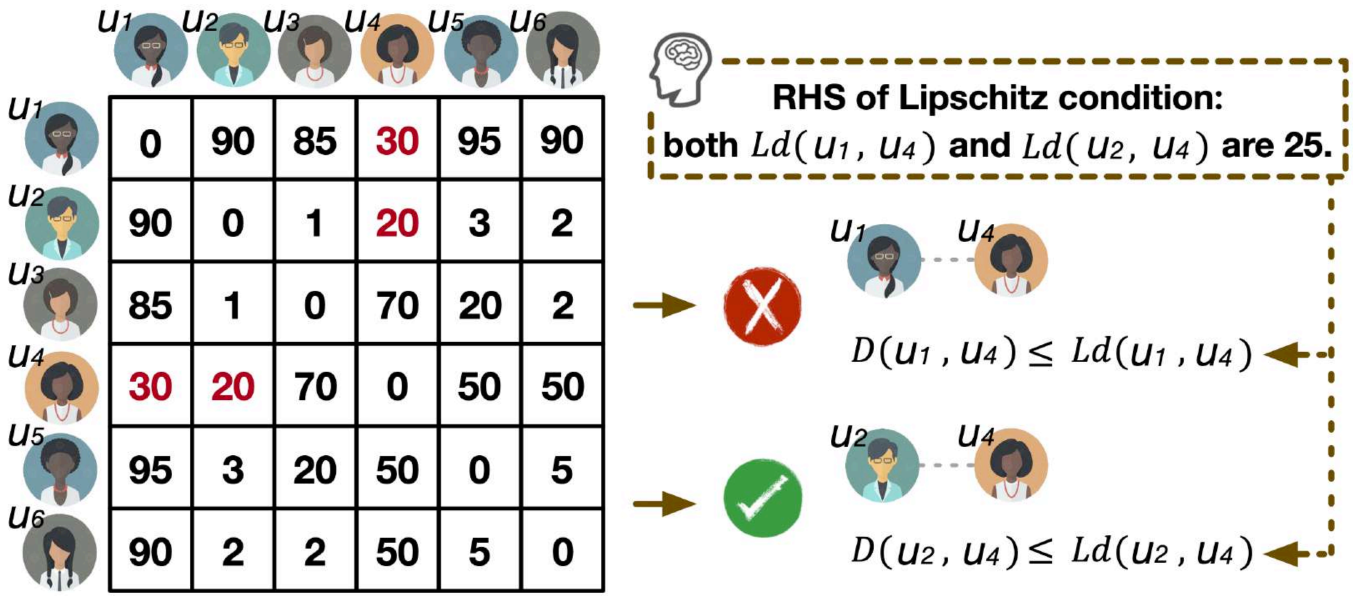
Output distance   Input distance

From different domains and different distance metrics.

How can we achieve individual fairness without specifying such a constant and avoid distance comparison across domains?

Existing Problems & Challenges

- **Distance Calibration.**



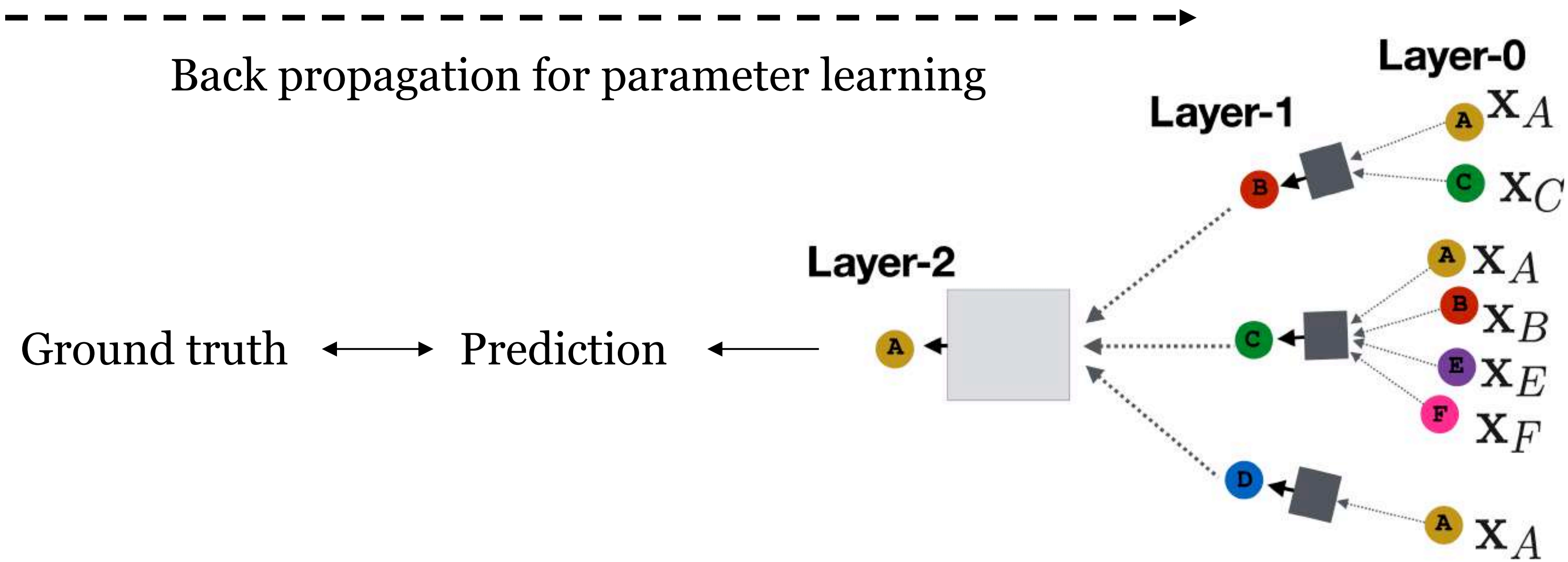
(a) Outcome distance matrix from distance metric D

(b) Lipschitz condition judgement based on human knowledge

How can we achieve individual fairness with natural calibration across different individuals?

Existing Problems & Challenges

- **End-to-End Learning Paradigm.**



An important advantage of GNN is its end-to-end learning* paradigm. **How can we achieve individual fairness without jeopardizing such advantage?**

*End-to-end learning usually refers to omitting any hand-crafted intermediary algorithms and directly learning the solution of a given problem from the sampled dataset.

Outline

- Background Introduction
- Previous Works
- Existing Problems & Challenges
- Our Solutions
- Experiments & Conclusion
- Future Works

Our Solutions: Outline

- **Constraint Formulation.**
- **Distance Calibration.**

New individual fairness definition from a ranking perspective.

↓

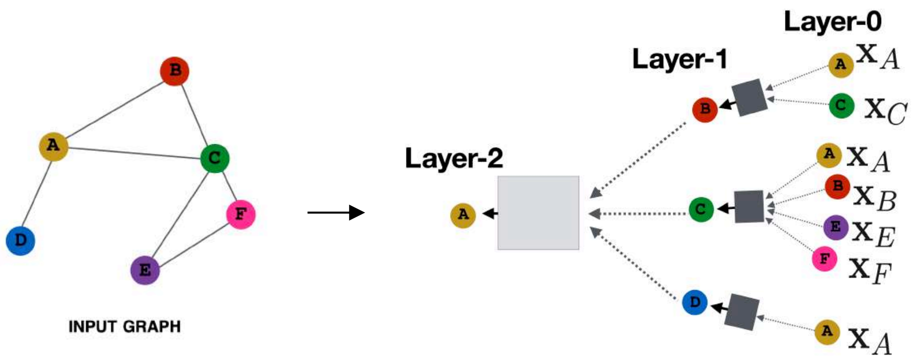
Problem formulation.

↓

- **End-to-End Learning Paradigm** → REDRESS framework.

Our Solutions: Ranking-based Individual Fairness

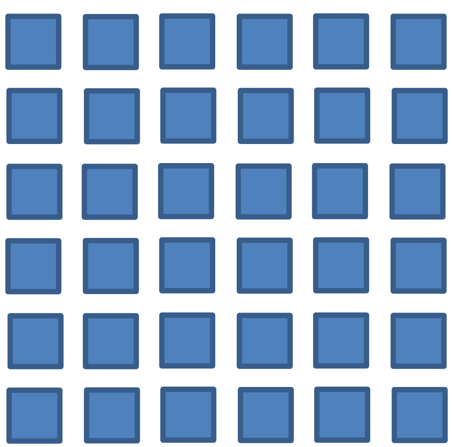
Traditional GNN based graph mining data flow without fairness consideration:



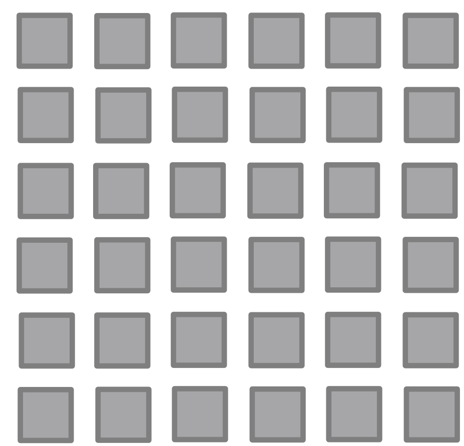
→ Prediction $\hat{\mathbf{Y}}$ ↔ Ground truth $\mathbf{Y}$

Extra fairness indicators for input and output data:

Oracle pairwise similarity matrix $\mathbf{S}_{\mathcal{G}}$

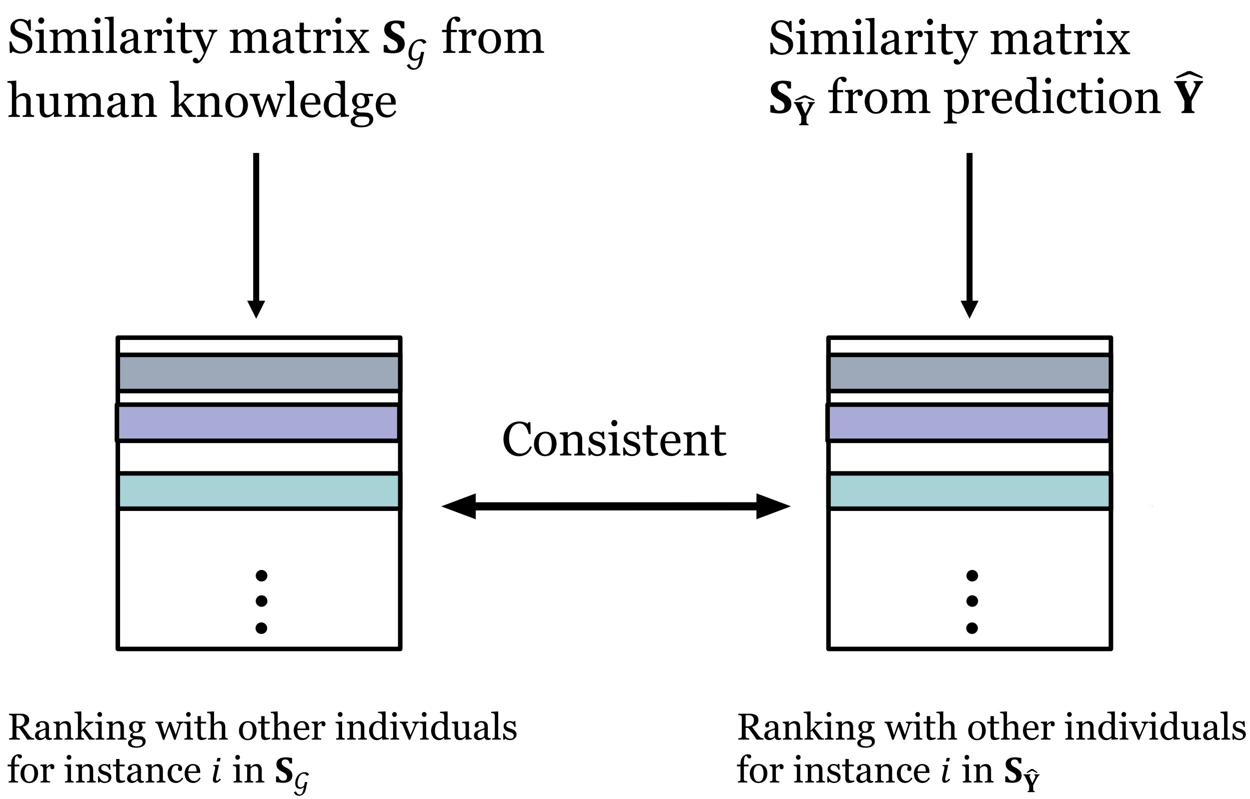


Pairwise similarity matrix $\mathbf{S}_{\hat{\mathbf{Y}}}$ obtained from $\hat{\mathbf{Y}}$



Our Solutions: Ranking-based Individual Fairness

Ranking based individual fairness: for each individual i, if we have

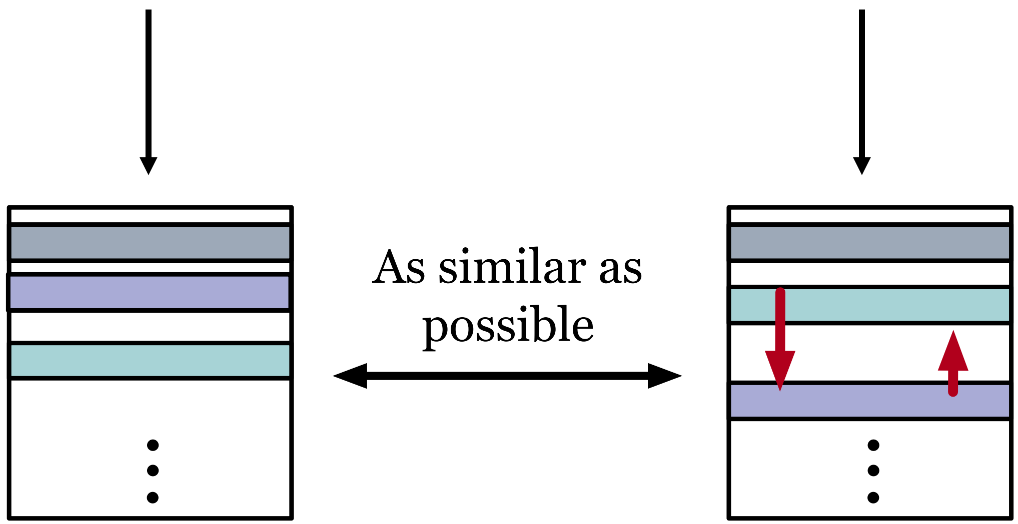


Our Solutions: Ranking-based Problem Formulation

Promoting individual fairness from a ranking perspective: for each individual i, our goal is to

Similarity matrix $\mathbf{S}_{\mathcal{G}}$ from human knowledge

Similarity matrix $\mathbf{S}_{\hat{\mathbf{Y}}}$ from prediction $\hat{\mathbf{Y}}$



Ranking with other individuals for instance i in $\mathbf{S}_{\mathcal{G}}$

Ranking with other individuals for instance i in $\mathbf{S}_{\hat{\mathbf{Y}}}$

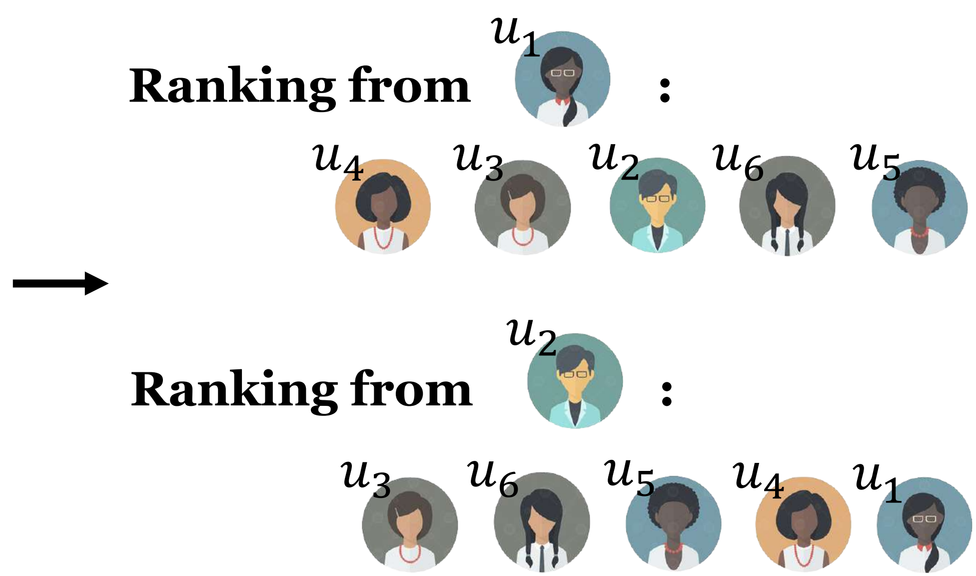
Our Solutions: Ranking-based Problem Formulation

The newly proposed Ranking-based individual fairness definition & corresponding problem formulation:

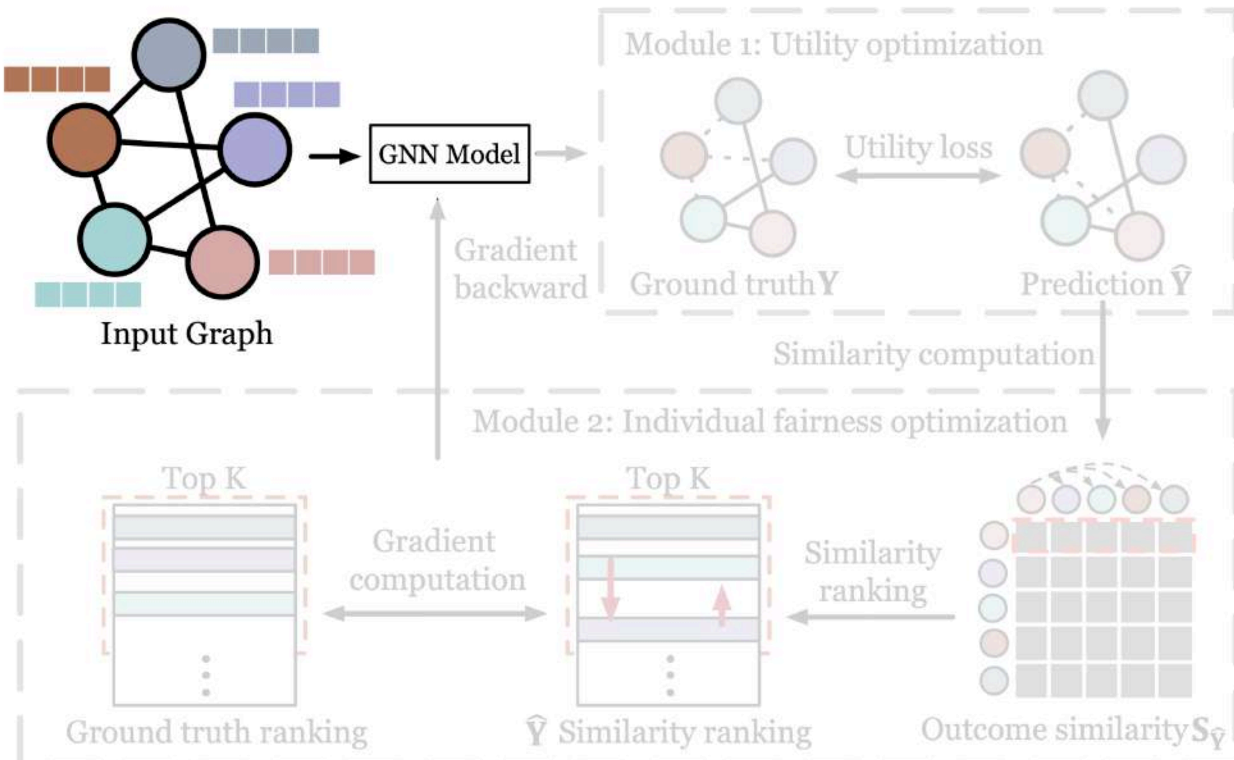
- Naturally calibrate across individuals;
- Provide a new constraint criterion to achieve individual fairness;

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	0	90	85	30	95	90
u_2	90	0	1	20	3	2
u_3	85	1	0	70	20	2
u_4	30	20	70	0	50	50
u_5	95	3	20	50	0	5
u_6	90	2	2	50	5	0

Outcome distance matrix from distance metric D

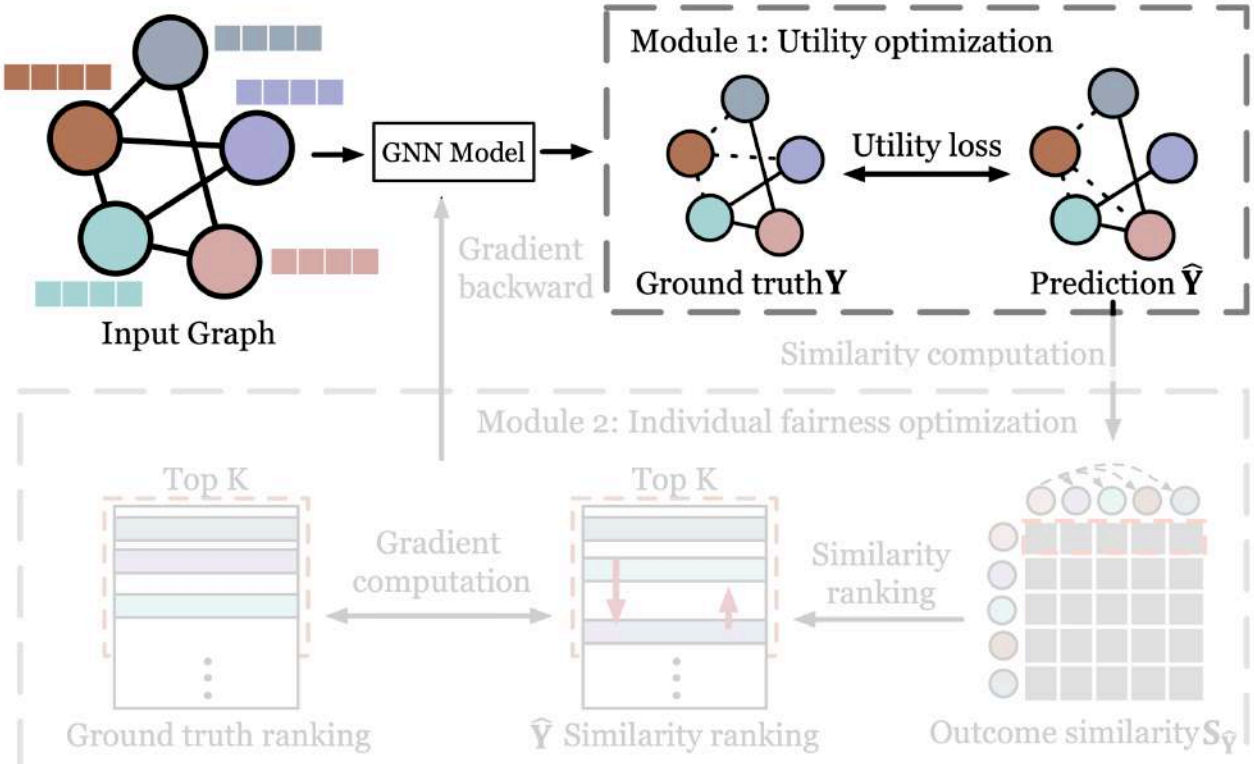


Our Solutions: Proposed Framework—REDRESS



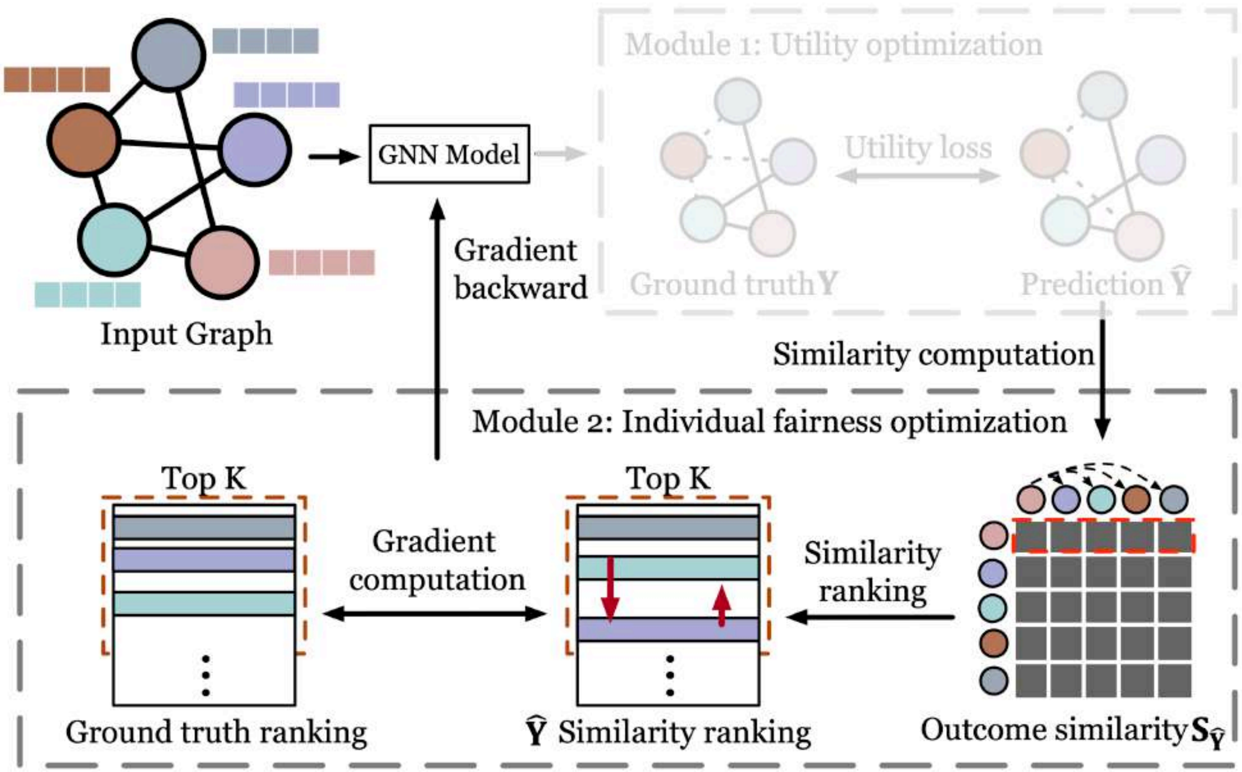
- **GNN backbone model.**
  Basic GNN structure to achieve downstream tasks.
- **Utility maximization.**
  It aims to minimize the downstream task loss.
- **Individual fairness optimization.**
  It aims to enforces the similarity rankings from $\mathbf{S}_{\hat{Y}}$ and $\mathbf{S}_{\mathcal{G}}$ to be similar.

Our Solutions: Proposed Framework—REDRESS



- **GNN backbone model.**
  Basic GNN structure to make downstream tasks.
- **Utility maximization.**
  It aims to minimize the downstream task loss.
- **Individual fairness optimization.**
  It aims to enforces the similarity rankings from $\mathbf{S}_{\hat{Y}}$ and $\mathbf{S}_{\mathcal{G}}$ to be similar.

Our Solutions: Proposed Framework—REDRESS



- **GNN backbone model.**
  Basic GNN structure to make downstream tasks.
- **Utility maximization.**
  It aims to minimize the downstream task loss.
- **Individual fairness optimization.**
  It aims to enforces the similarity rankings from $\mathbf{S}_{\hat{Y}}$ and $\mathbf{S}_{\mathcal{G}}$ to be similar.

REDRESS–GNN Backbone

GNN backbone: The basic operation of GNN between l–th layer and $(l+1)$–th layer can be summarized as:

$$\mathbf{h}_v^{(l+1)} = \sigma(\mathrm{COMBINE}(\mathbf{h}_v^{(l)}, f(\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v)\})))$$

$\mathbf{h}_v^{(l+1)}$: Embedding of node v at $(l+1)$-th layer;

$f(.)$: Information aggregation function;

$\mathrm{COMBINE}(.)$: Information combine function;

$\sigma(.)$: Activation function;

$\mathcal{N}(v)$: Neighborhood set of node v;

REDRESS–Utility Maximization

Utility maximization: loss function can be initialized as the cross-entropy between predictions and ground truth.

$$\mathcal{L} = -\sum_{(i,j)\in\mathcal{T}} \mathrm{Y}_{ij} \ln \hat{\mathrm{Y}}_{ij}$$

Node classification: $\mathcal{T}$ is the (node, class) tuple set for training nodes.

Link prediction : $\mathcal{T}$ is the (node, node) tuple set for the vertices of training edges.

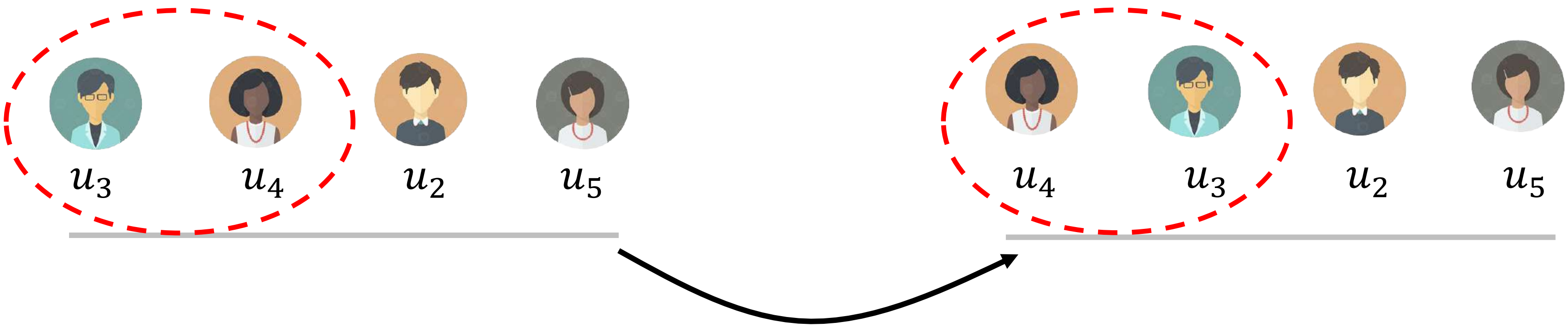
REDRESS–Individual Fairness Optimization

Individual fairness optimization: for every node, the two similarity ranking lists with other nodes derived from $\mathbf{S}_{\hat{\mathbf{Y}}}$ and $\mathbf{S}_{\mathcal{G}}$ should be similar.

Example: Ranking* from u_1

Ranking in $\mathbf{S}_{\hat{\mathbf{Y}}}$ derived from prediction $\hat{\mathbf{Y}}$

Ground truth ranking from human knowledge $\mathbf{S}_{\mathcal{G}}$



Goal: to make the two ranking lists as similar as possible.

*We omit u_6 for simplification purpose.

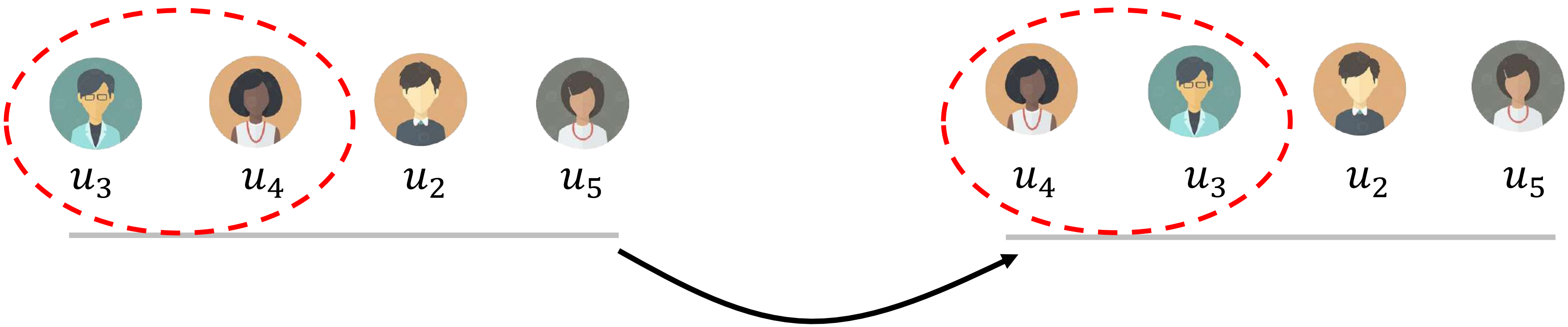
REDRESS–Individual Fairness Optimization

Individual fairness optimization: for every node, the two similarity ranking lists with other nodes derived from $\mathbf{S}_{\hat{\mathbf{Y}}}$ and $\mathbf{S}_{\mathcal{G}}$ should be similar.

Example: Ranking* from

Ranking in $\mathbf{S}_{\hat{\mathbf{Y}}}$ derived from prediction $\hat{\mathbf{Y}}$

Ground truth ranking from human knowledge $\mathbf{S}_{\mathcal{G}}$



Directly defining a loss between the two ranking lists: **loss is non-differentiable**.

*We omit u_6 for simplification purpose.

REDRESS–Individual Fairness Optimization

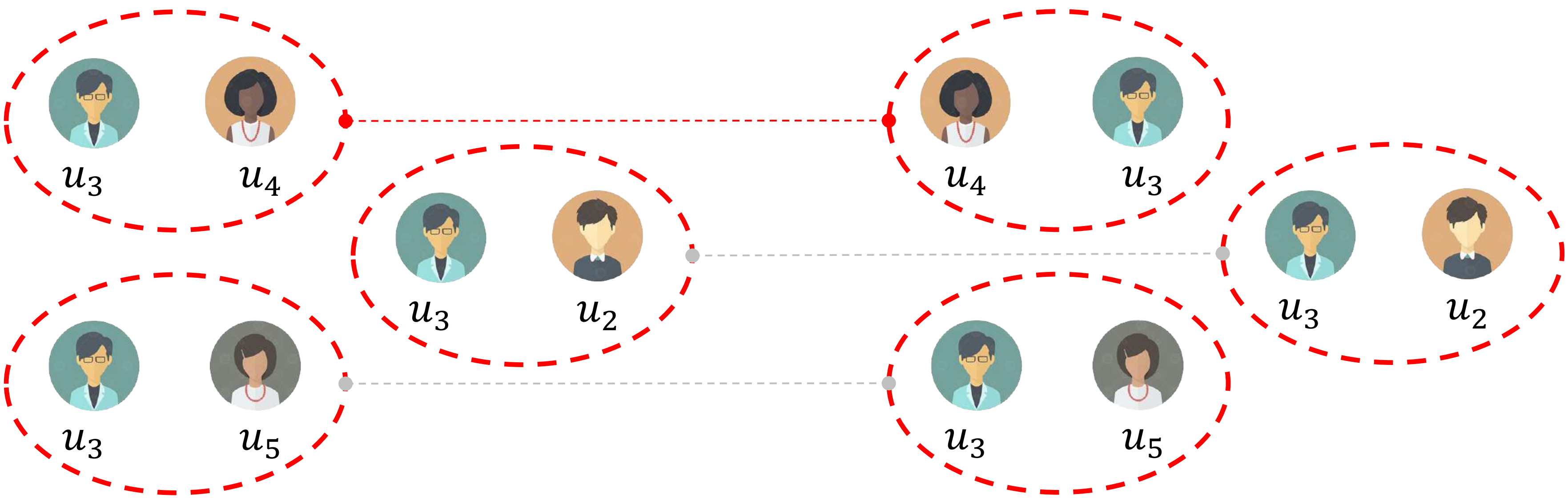
We turn to make the **relative ranking order of every pair** to be consistent with that in the ground truth ranking.

Take  as an example:

u_3

Ranking in $\mathbf{S}_{\hat{\mathbf{Y}}}$ derived from prediction $\hat{\mathbf{Y}}$

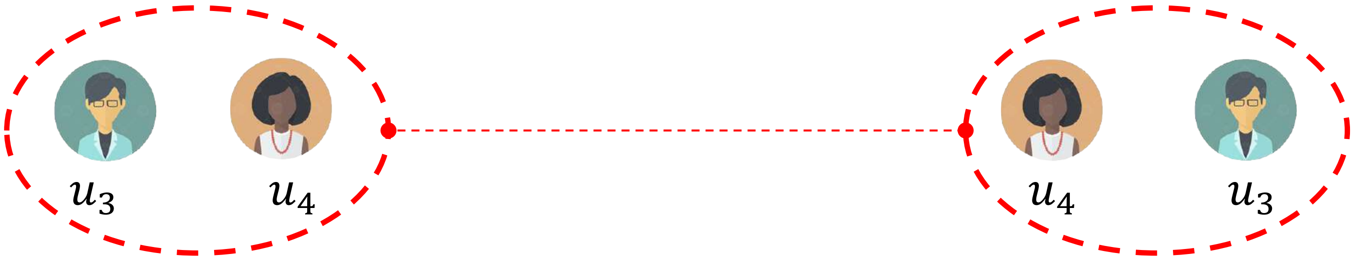
Ground truth ranking from human knowledge $\mathbf{S}_{\mathcal{G}}$



REDRESS–Individual Fairness Optimization

Ranking in $\mathbf{S}_{\hat{\mathbf{Y}}}$ derived from prediction $\hat{\mathbf{Y}}$

Ground truth ranking from human knowledge $\mathbf{S}_{\mathcal{G}}$



Probability of 'u is ahead of v' is modeled as a logistic sigmoid function:

$$\hat{P}_{u,v}(i) = \frac{1}{1 + e^{-\alpha(\hat{s}_{i,u} - \hat{s}_{i,v})}}$$

We set the corresponding ground truth as

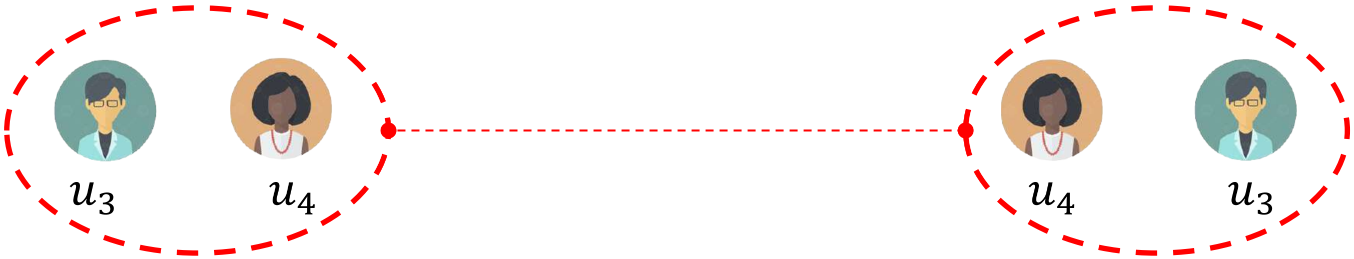
$$P_{u,v}(i) = \frac{1}{2}\left(1 + T_{u,v}(i)\right)$$

$$T_{u,v}(i) = \begin{cases} +1 & \text{the } u\text{-th one ranks higher} \\ 0 & \text{the } u\text{-th and } v\text{-th one rank samely} \\ -1 & \text{the } v\text{-th one ranks higher} \end{cases}$$

REDRESS–Individual Fairness Optimization

Ranking in $\mathbf{S}_{\hat{\mathbf{Y}}}$ derived from prediction $\hat{\mathbf{Y}}$

Ground truth ranking from human knowledge $\mathbf{S}_{\mathcal{G}}$



$$\hat{P}_{u,v}(i) = \frac{1}{1 + e^{-\alpha(\hat{s}_{i,u} - \hat{s}_{i,v})}} \qquad P_{u,v}(i) = \frac{1}{2}(1 + T_{u,v}(i))$$

Cross-entropy loss can be an example:

$$\mathcal{L}_{j,m}(i) = -P_{j,m}\log\hat{P}_{j,m} - (1 - P_{j,m})\log(1 - \hat{P}_{j,m})$$

REDESS–Individual Fairness Optimization

An example: Assume ranked $\hat{\mathbf{s}}_{u_1} = [0.7, 0.3, 0.2, 0.1]$.

Ranking in $\mathbf{S}_{\hat{\mathbf{Y}}}$ derived from prediction $\hat{\mathbf{Y}}$

u_3 u_4 u_2 u_5

Ground truth ranking from human knowledge $\mathbf{S}_{\mathcal{G}}$

u_4 u_3 u_2 u_5

For u_3:

(u_3, u_4)= 0.98		(u_3, u_4)= 0
(u_3, u_5)= 0.99	Cross-entropy loss	(u_3, u_5)= 1
(u_3, u_2)= 0.99		(u_3, u_2)= 1

Assume hyper-parameter $\alpha = 10$

REDESS–Individual Fairness Optimization

An example: Assume **optimized** $\hat{\mathbf{s}}_{u_1} = [0.7, 0.6, 0.2, 0.1]$.

Ranking in $\mathbf{S}_{\hat{\mathbf{Y}}}$ derived from prediction $\hat{\mathbf{Y}}$

u_3 u_4 u_2 u_5

Ground truth ranking from human knowledge $\mathbf{S}_{\mathcal{G}}$

u_4 u_3 u_2 u_5

For u_3:

$(u_3, u_4) = 0.73$		$(u_3, u_4) = 0$
$(u_3, u_5) = 0.99$	Cross-entropy loss	$(u_3, u_5) = 1$
$(u_3, u_2) = 0.99$		$(u_3, u_2) = 1$

Assume hyper-parameter $\alpha = 10$

REDRESS–Individual Fairness Optimization

An example: Assume **optimized** $\hat{\mathbf{s}}_{u_1} = [0.4, 0.7, 0.2, 0.1]$.

Ranking in $\mathbf{S}_{\hat{\mathbf{Y}}}$ derived from prediction $\hat{\mathbf{Y}}$

u_3 u_4 u_2 u_5

Ground truth ranking from human knowledge $\mathbf{S}_{\mathcal{G}}$

u_4 u_3 u_2 u_5

For u_3:

$(u_3, u_4) = 0.05$		$(u_3, u_4) = 0$
$(u_3, u_5) = 0.88$	Cross-entropy loss	$(u_3, u_5) = 1$
$(u_3, u_2) = 0.95$		$(u_3, u_2) = 1$

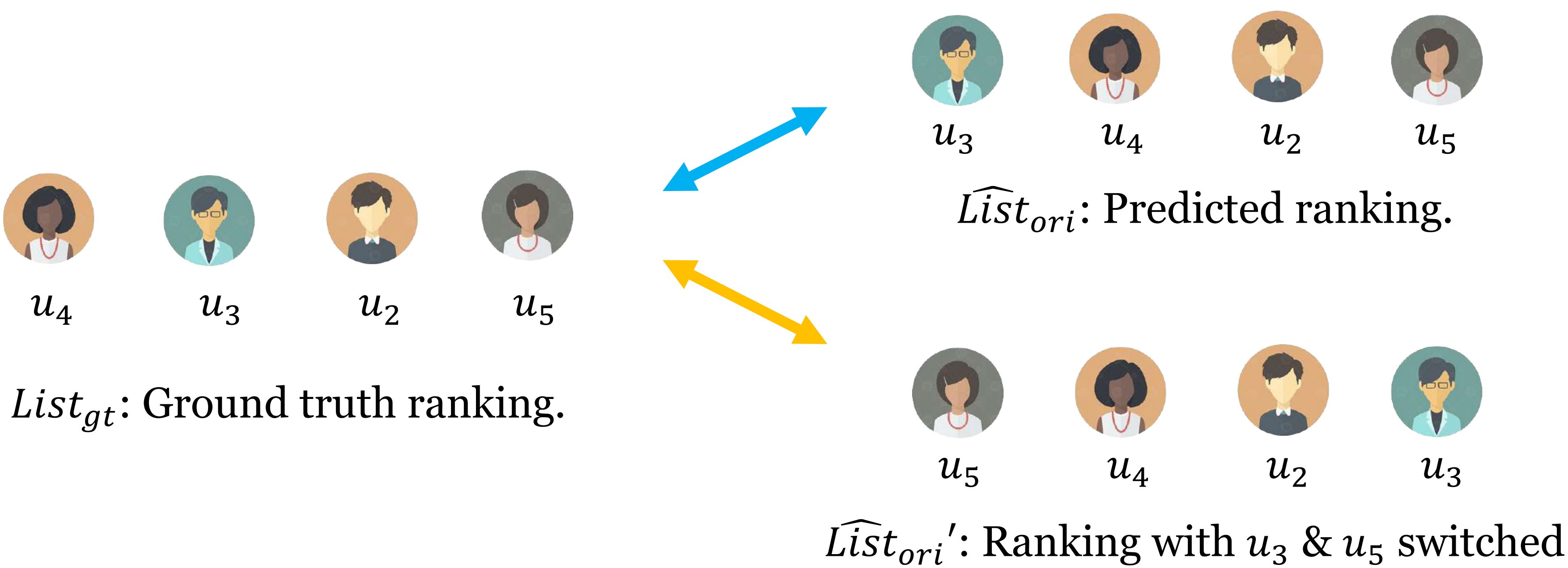
Assume hyper-parameter $\alpha = 10$

Total Loss Formulation

Simple sum of each individual loss: $\mathcal{L}_{\text{fairness}}(i) = \sum_{j,m} \mathcal{L}_{j,m}(i)$

Loss with training facilitation: $\mathcal{L}_{\text{fairness}}(i) = \sum_{j,m} \mathcal{L}_{j,m}(i) |\Delta z_{@k}|_{j,m}$

$$|\Delta z_{@k}|_{3,5} = |z_{@k}(List_{gt}, \widehat{List}_{ori}) - z_{@k}(List_{gt}, \widehat{List}_{ori}')|$$



$z_{@k}$ choices: NDCG@K, ERR@K, etc.

Computational Simplification

Loss with training facilitation: $\mathcal{L}_{\text{fairness}}(i) = \sum_{j,m} \mathcal{L}_{j,m}(i)|\Delta z_{@k}|_{j,m}$

Disadvantage: High computational cost when n (the length of the ranking) is large ($\mathcal{O}(n^2k)$).

Computational Simplification: Restrict node j and m only within the top-k ranked nodes (reduced from $\mathcal{O}(n^2k)$ to $\mathcal{O}(k^3)$).

$$\mathcal{L}_{\text{fairness}}(i) = \sum_{j,m:\, j,m \in \mathcal{K}(i)} \mathcal{L}_{j,m}(i)|\Delta z_{@k}|_{j,m}$$

$\mathcal{K}(i)$ is the set of the top-k ranked nodes for node i.

Total Loss Formulation

Utility loss: cross-entropy loss for model utility (sum of all training nodes);

Individual Fairness loss: cross-entropy loss for predicted ranking (sum of all training nodes);

Total loss formulation:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{utility}} + \gamma \mathcal{L}_{\text{fairness}}$$

$$\mathcal{L}_{\text{utility}} = - \sum_{(i,j) \in \mathcal{T}} \mathrm{Y}_{ij} \ln \hat{\mathrm{Y}}_{ij}$$

$$\mathcal{L}_{\text{fairness}} = \sum_{i} \sum_{j,m: j,m \in \mathcal{K}(i)} \mathcal{L}_{j,m}(i) |\Delta z_{@k}|_{j,m}$$

Outline

Background Introduction

Previous Works

Existing Problems & Challenges

Our Solutions

Experiments & Conclusion

Future Works

Experimental Settings

Downstream tasks:

- Node classification;
- Link prediction;

Datasets:

- ACM [Tang et al., 2008] , Coauthor CS and Coauthor Phy [Shchur et al., 2018] for node classification;
- BlogCatalog [Tang et al., 2009] , Flickr [Huang et al., 2017] and Facebook [McAuley et al., 2012] for link prediction;

Table 1: Statistics of datasets.

	Dataset	# Nodes	# Edges	# Features	# Classes
NC	ACM	16,484	71,980	8,337	9
	CS	18,333	81,894	6,805	15
	Phy	34,493	247,962	8,415	5
LP	BlogCatalog	5,196	171,743	8,189	N/A
	Flickr	7,575	239,738	12,047	N/A
	Facebook	4,039	88,234	1,406	N/A

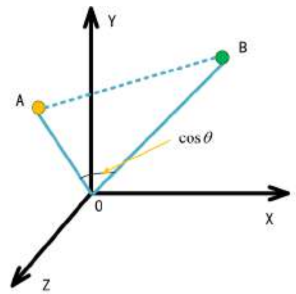
Experimental Settings

GNN backbones:

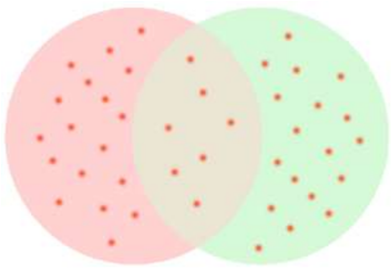
- GCN [Kipf et al., 2016] and SGC [Wu et al., 2019] for node classification;
- GCN [Kipf et al., 2016] and GAE [Kipf et al., 2016] for link prediction;

Oracle Similarity Matrix:

- Cosine similarity (feature-based);
- Jaccard similarity (structure-based);



Cosine similarity

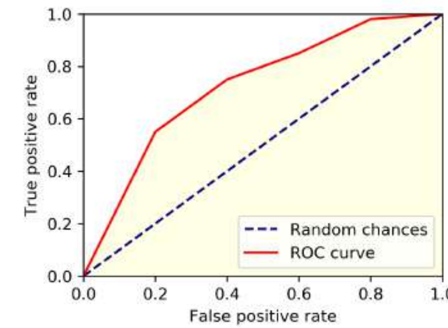


Jaccard similarity

Experimental Settings

Baselines:

- PFR [Lahoti et al., 2019] (not specially designed for graphs);
- InFoRM [Kang et al., 2020] (introduced before);

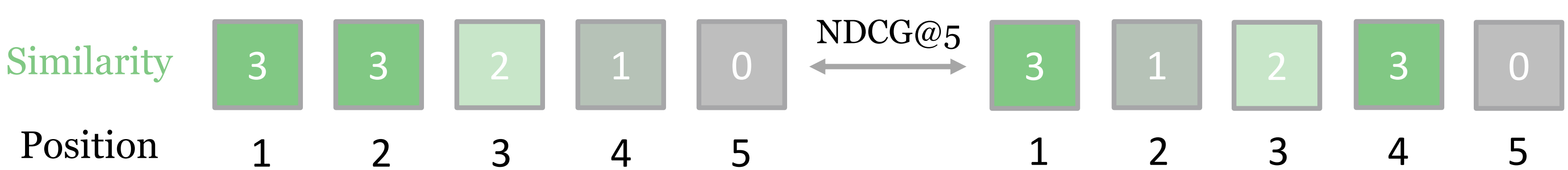


Evaluation Metrics:

- Model utility:

  Node classification accuracy (ACC);

  Area under receiver operating characteristic curve (AUC);
- Individual fairness: NDCG@10 as $z_{@k}$ ($k = 10$) for ranking similarity evaluation;



NDCG: Normalized Discounted Cumulative Gain

Research Questions

- **RQ1:** How well can REDRESS balance the GNN model utility and individual fairness compared with other baselines?
- **RQ2:** How will the individual fairness promotion hyperparameter γ affect the performance of REDRESS?
- **RQ3:** How will the choice of parameter k affect the performance of REDRESS?

Experimental Results

RQ1: How well can REDRESS balance the GNN model utility and individual fairness compared with other baselines?

- Take performance on ACM as an example.

Our model achieves **comparable** performance on model utility compared with the best ones.

Table 2: **Node classification** results on ACM.

		Feature Similarity		Structural Similarity	
BB	**Model**	**Utility: ACC**	**Fairness: NDCG@10**	**Utility: ACC**	**Fairness: NDCG@10**
ACM GCN	Vanilla	**72.49 ± 0.6** (—)	47.33 ± 1.0 (—)	**72.49 ± 0.6** (—)	25.42 ± 0.6 (—)
	InFoRM	68.03 ± 0.3 (−6.15%)	39.79 ± 0.3 (−15.9%)	69.13 ± 0.5 (−4.64%)	12.02 ± 0.4 (−52.7%)
	PFR	67.88 ± 1.1 (−6.36%)	31.20 ± 0.2 (−34.1%)	69.00 ± 0.7 (−4.81%)	23.85 ± 1.3 (−6.18%)
	REDRESS (Ours)	71.75 ± 0.4 (−1.02%)	**49.13 ± 0.4** (+3.80%)	72.03 ± 0.9 (−0.63%)	**29.09 ± 0.4** (+14.4%)
ACM SGC	Vanilla	68.40 ± 1.0 (—)	55.75 ± 1.1 (—)	**68.40 ± 1.0** (—)	37.18 ± 0.6 (—)
	InFoRM	**68.81 ± 0.5** (+0.60%)	48.25 ± 0.5 (−13.5%)	66.71 ± 0.6 (−2.47%)	28.33 ± 0.6 (−23.8%)
	PFR	67.97 ± 0.7 (−0.62%)	34.71 ± 0.1 (−37.7%)	67.78 ± 0.1 (−0.91%)	37.15 ± 0.6 (−0.08%)
	REDRESS (Ours)	67.16 ± 0.2 (−1.81%)	**58.64 ± 0.4** (+5.18%)	67.77 ± 0.4 (−0.92%)	**38.95 ± 0.1** (+4.76%)

Experimental Results

RQ1: How well can REDRESS balance the GNN model utility and individual fairness compared with other baselines?

- Take performance on ACM as an example.

Our model achieves **best** performance on individual fairness.

Table 2: **Node classification** results on ACM.

	BB	Model	Feature Similarity		Structural Similarity	
			Utility: ACC	Fairness: NDCG@10	Utility: ACC	Fairness: NDCG@10
ACM	GCN	Vanilla	**72.49 ± 0.6** (—)	47.33 ± 1.0 (—)	**72.49 ± 0.6** (—)	25.42 ± 0.6 (—)
		InFoRM	68.03 ± 0.3 (−6.15%)	39.79 ± 0.3 (−15.9%)	69.13 ± 0.5 (−4.64%)	12.02 ± 0.4 (−52.7%)
		PFR	67.88 ± 1.1 (−6.36%)	31.20 ± 0.2 (−34.1%)	69.00 ± 0.7 (−4.81%)	23.85 ± 1.3 (−6.18%)
		REDRESS (Ours)	71.75 ± 0.4 (−1.02%)	**49.13 ± 0.4** (+3.80%)	72.03 ± 0.9 (−0.63%)	**29.09 ± 0.4** (+14.4%)
	SGC	Vanilla	68.40 ± 1.0 (—)	55.75 ± 1.1 (—)	**68.40 ± 1.0** (—)	37.18 ± 0.6 (—)
		InFoRM	**68.81 ± 0.5** (+0.60%)	48.25 ± 0.5 (−13.5%)	66.71 ± 0.6 (−2.47%)	28.33 ± 0.6 (−23.8%)
		PFR	67.97 ± 0.7 (−0.62%)	34.71 ± 0.1 (−37.7%)	67.78 ± 0.1 (−0.91%)	37.15 ± 0.6 (−0.08%)
		REDRESS (Ours)	67.16 ± 0.2 (−1.81%)	**58.64 ± 0.4** (+5.18%)	67.77 ± 0.4 (−0.92%)	**38.95 ± 0.1** (+4.76%)

Experimental Results

RQ1: How well can REDRESS balance the GNN model utility and individual fairness compared with other baselines?

- Similar conclusion in link prediction (on Blog as an example).

Our model achieves **comparable** performance on model utility compared with the best ones.

Table 2: **Link prediction** results on BlogCatalog (Blog).

	BB	Model	Feature Similarity		Structural Similarity	
			Utility: AUC	Fairness: NDCG@10	Utility: AUC	Fairness: NDCG@10
Blog	GCN	Vanilla	85.87 ± 0.1 (—)	16.73 ± 0.1 (—)	85.87 ± 0.1 (—)	32.47 ± 0.5 (—)
		InFoRM	79.85 ± 0.6 (−7.01%)	15.57 ± 0.2 (−6.93%)	84.00 ± 0.1 (−2.18%)	26.18 ± 0.3 (−19.4%)
		PFR	84.25 ± 0.2 (−1.89%)	16.37 ± 0.0 (−2.15%)	83.88 ± 0.0 (−2.32%)	29.60 ± 0.4 (−8.84%)
		REDRESS (Ours)	**86.49 ± 0.8** (+0.72%)	**17.66 ± 0.2** (+5.56%)	**86.25 ± 0.3** (+0.44%)	**34.62 ± 0.7** (+6.62%)
	GAE	Vanilla	**85.72 ± 0.1** (—)	17.13 ± 0.1 (—)	85.72 ± 0.1 (—)	41.99 ± 0.4 (—)
		InFoRM	80.01 ± 0.2 (−6.66%)	16.12 ± 0.2 (−5.90%)	82.86 ± 0.0 (−3.34%)	27.29 ± 0.3 (−35.0%)
		PFR	83.83 ± 0.1 (−2.20%)	16.64 ± 0.0 (−2.86%)	83.87 ± 0.1 (−2.16%)	35.91 ± 0.4 (−14.5%)
		REDRESS (Ours)	84.67 ± 0.9 (−1.22%)	**18.19 ± 0.1** (+6.19%)	**86.36 ± 1.5** (+0.75%)	**43.51 ± 0.7** (+3.62%)

Experimental Results

RQ1: How well can REDRESS balance the GNN model utility and individual fairness compared with other baselines?

- Similar conclusion in link prediction (on Blog as an example).

Our model achieves **best** performance on individual fairness.

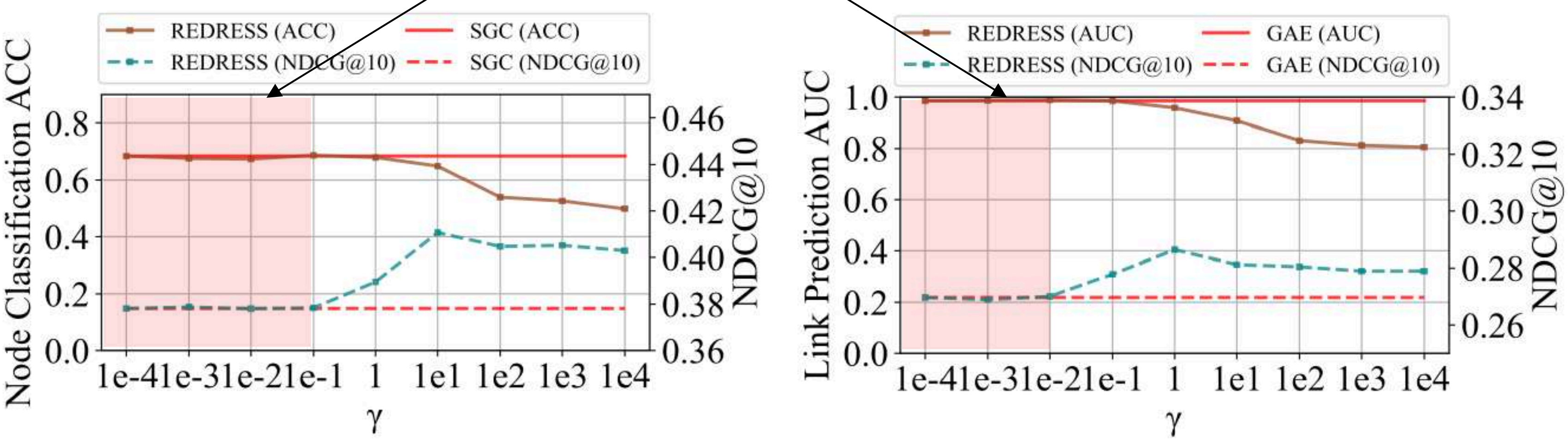
Table 2: **Link prediction** results on BlogCatalog (Blog).

	BB	Model	Feature Similarity		Structural Similarity	
			Utility: AUC	Fairness: NDCG@10	Utility: AUC	Fairness: NDCG@10
Blog	GCN	Vanilla	85.87 ± 0.1 (−)	16.73 ± 0.1 (−)	85.87 ± 0.1 (−)	32.47 ± 0.5 (−)
		InFoRM	79.85 ± 0.6 (−7.01%)	15.57 ± 0.2 (−6.93%)	84.00 ± 0.1 (−2.18%)	26.18 ± 0.3 (−19.4%)
		PFR	84.25 ± 0.2 (−1.89%)	16.37 ± 0.0 (−2.15%)	83.88 ± 0.0 (−2.32%)	29.60 ± 0.4 (−8.84%)
		REDRESS (Ours)	**86.49 ± 0.8** (+0.72%)	**17.66 ± 0.2** (+5.56%)	**86.25 ± 0.3** (+0.44%)	**34.62 ± 0.7** (+6.62%)
	GAE	Vanilla	**85.72 ± 0.1** (−)	17.13 ± 0.1 (−)	85.72 ± 0.1 (−)	41.99 ± 0.4 (−)
		InFoRM	80.01 ± 0.2 (−6.66%)	16.12 ± 0.2 (−5.90%)	82.86 ± 0.0 (−3.34%)	27.29 ± 0.3 (−35.0%)
		PFR	83.83 ± 0.1 (−2.20%)	16.64 ± 0.0 (−2.86%)	83.87 ± 0.1 (−2.16%)	35.91 ± 0.4 (−14.5%)
		REDRESS (Ours)	84.67 ± 0.9 (−1.22%)	**18.19 ± 0.1** (+6.19%)	**86.36 ± 1.5** (+0.75%)	**43.51 ± 0.7** (+3.62%)

Experimental Results

RQ2: How will the individual fairness promotion hyperparameter γ affect the performance of REDRESS?

Relatively small: the individual fairness constraint makes little difference;



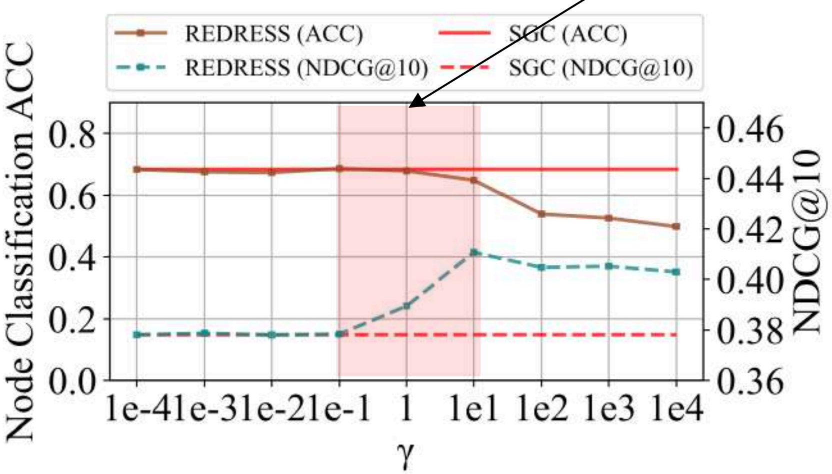
(a) ACM

(b) Facebook

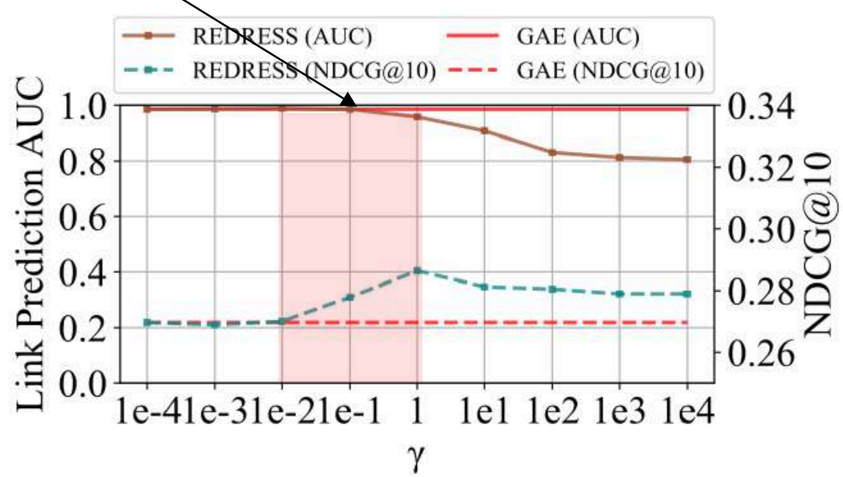
Experimental Results

RQ2: How will the individual fairness promotion hyperparameter γ affect the performance of REDRESS?

Modest value : NDCG@10 can be improved with little sacrifice on ACC or AUC;



(a) ACM

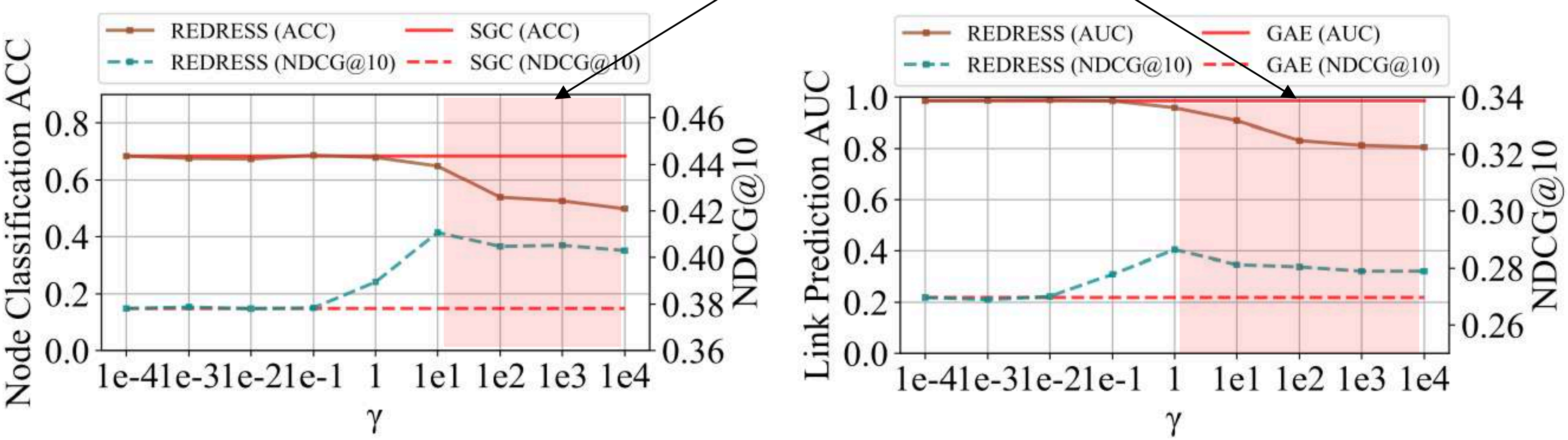


(b) Facebook

Experimental Results

RQ2: How will the individual fairness promotion hyperparameter γ affect the performance of REDRESS?

Relatively large: ACC and AUC will be affected by the strength of individual fairness promotion;

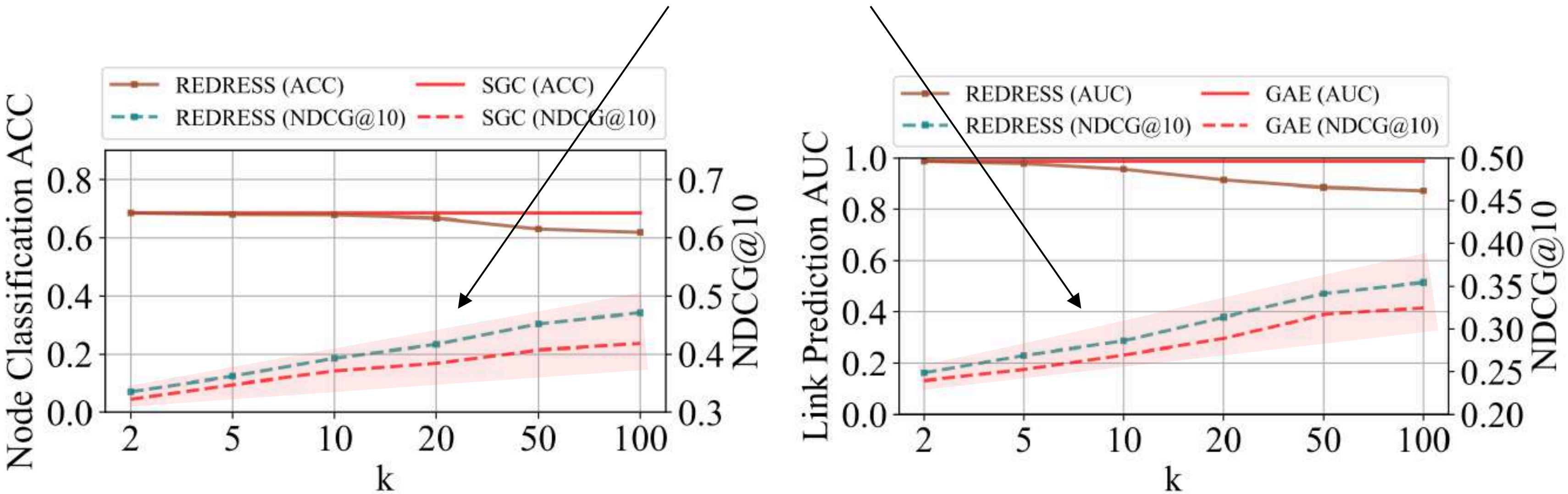


(a) ACM (b) Facebook

Experimental Results

RQ3: How will the choice of parameter *k* affect the performance of REDRESS?

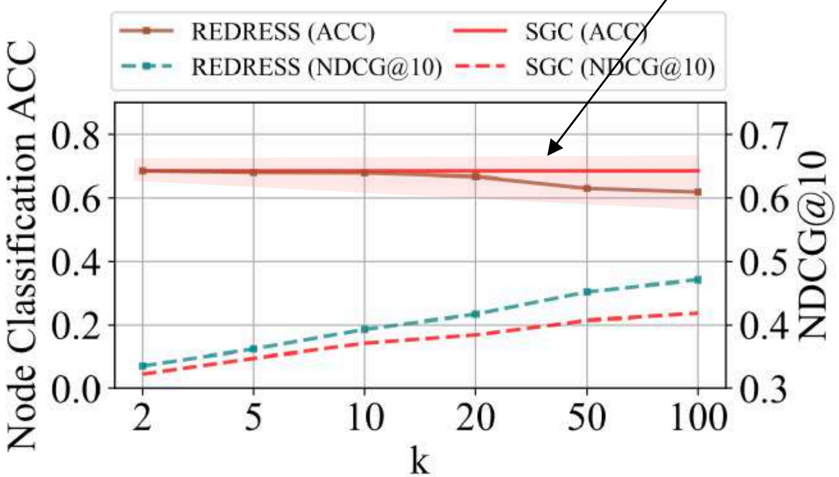
As *k* goes larger, REDRESS achieves larger improvement on NDCG;



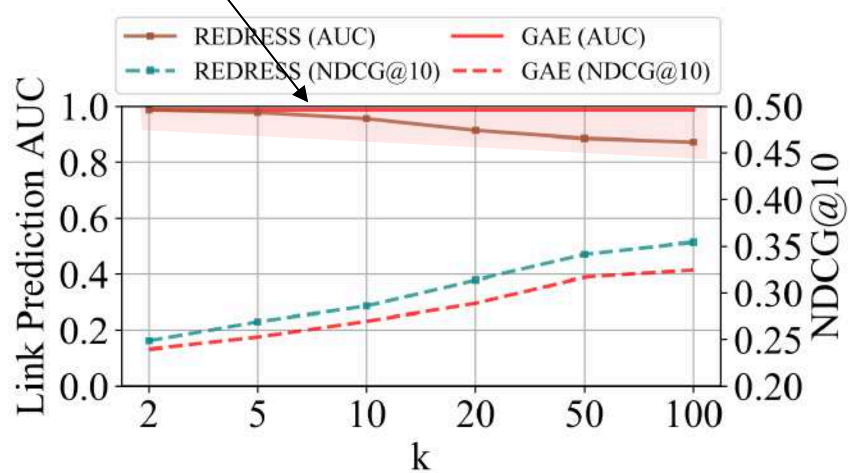
(a) ACM **(b) Facebook**

RQ3: How will the choice of parameter *k* affect the performance of REDRESS?

Model utility performance is barely influenced when *k* gets larger;



(a) ACM



(b) Facebook

Outline

- Background Introduction
- Previous Works
- Existing Problems & Challenges
- Our Solutions
- Experiments & Conclusion
- Future Works

Future Works

- Traditional graph mining algorithm debias;
- Generalization on different graphs (i.e., time-series graphs);
- Scalability on large graphs;

Reference

[Kang et al. 2020] Kang J, He J, Maciejewski R, et al. InFoRM: Individual Fairness on Graph Mining[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 379-389.

[Kipf et al. 2016] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[Kipf et al. 2016] Kipf T N, Welling M. Variational graph auto-encoders[J]. arXiv preprint arXiv:1611.07308, 2016.

[Wu et al. 2019] Wu F, Souza A, Zhang T, et al. Simplifying graph convolutional networks[C]//International conference on machine learning. PMLR, 2019: 6861-6871.

[Lahoti et al. 2019] Lahoti P, Gummadi K P, Weikum G. Operationalizing individual fairness with pairwise fair representations[J]. arXiv preprint arXiv:1907.01439, 2019.

[Tang et al. 2008] Tang J, Zhang J, Yao L, et al. Arnetminer: extraction and mining of academic social networks[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008: 990-998.

[Shchur et al. 2018] Shchur O, Mumme M, Bojchevski A, et al. Pitfalls of graph neural network evaluation[J]. arXiv preprint arXiv:1811.05868, 2018.

[Tang et al. 2009] Tang L, Liu H. Relational learning via latent social dimensions[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009: 817-826.

[Huang et al. 2017] Huang X, Li J, Hu X. Label informed attributed network embedding[C]//Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. 2017: 731-739.

[McAuley et al. 2012] McAuley J J, Leskovec J. Learning to discover social circles in ego networks[C]//NIPS. 2012, 2012: 548-56.

The End

Thanks for listening!