APPENDIX: DECEPTIVE FAIRNESS ATTACKS ON GRAPHS VIA META LEARNING

ORGANIZATION OF THE APPENDIX

The supplementary material contains the following information.

- Appendix A provides additional examples of graph learning models from the optimization perspective.
- Appendix B presents the pseudocode of FATE.
- Appendix C offers the detailed parameter settings regarding the reproducibility of this paper.
- Appendix D provides additional experimental results on using FairGNN (**?**) and evaluating under macro F1 score and AUC score.
- Appendix E provides additional experimental results on using InFoRM-GNN (**?**) and evaluating under macro F1 score and AUC score.
- Appendix F shows the transferability of using FATE to attack the statistical parity or individual fairness of the non-convolutional aggregation-based graph attention network with linear GCN as the surrogate model.
- Appendix G provides further discussions on (1) the relationship between fairness attacks and the impossibility theorem as well as Metattack (**?**), (2) an alternative perturbation set selection strategy via sampling, (3) the potential of FATE on attacking the fairness of a specific demographic group, and (4) justification of applying kernel density estimation on non-IID graph data.
- Appendix H presents more details of statistical parity and individual fairness.

Code can be found at the following repository:

`https://github.com/jiank2/FATE`.

A GRAPH LEARNING MODELS FROM THE OPTIMIZATION PERSPECTIVE

Here, we discuss four additional non-parameterized graph learning models from the optimization perspective, including PageRank, spectral clustering, matrix factorization-based completion and first-order LINE.

Model #1: PageRank. It is one of the most successful random walk based ranking algorithm to measure node importance. Mathematically, PageRank solves the linear system

$$\mathbf{r} = c\mathbf{P}\mathbf{r} + (1-c)\mathbf{e} \tag{1}$$

where c is the damping factor, $\mathbf{P}$ is the propagation matrix and $\mathbf{e}$ is the teleportation vector. In PageRank, the propagation matrix $\mathbf{P}$ is often defined as the row-normalized adjacency matrix of a graph $\mathcal{G}$ and the teleportation vector is a uniform distribution $\frac{1}{n}\mathbf{1}$ with $\mathbf{1}$ being a vector filled with 1. Equivalently, given a damping factor c and a teleportation vector $\mathbf{e}$, the PageRank vector $\mathbf{Y} = \mathbf{r}$ can be learned by minimizing the following loss function

$$\min_{\mathbf{r}} \quad c\mathbf{r}^T(\mathbf{I} - \mathbf{P})\mathbf{r} + (1-c)\|\mathbf{r} - \mathbf{e}\|_2^2 \tag{2}$$

where $c\left(\mathbf{r}^T\left(\mathbf{I} - \mathbf{P}\right)\mathbf{r}\right)$ is a smoothness term and $(1-c)\left\|\mathbf{r} - \mathbf{e}\right\|_2^2$ is a query-specific term. To attack the fairness of PageRank with FATE, the attacker could attack a surrogate PageRank with different choices of damping factor c and/or teleportation vector $\mathbf{e}$.

Model #2: Spectral clustering. It aims to identify clusters of nodes such that the intra-cluster connectivity are maximized while inter-cluster connectivity are minimized. To find k clusters of nodes, spectral clustering finds a soft cluster membership matrix $\mathbf{Y} = \mathbf{C}$ with orthonormal columns by minimizing the following loss function

$$\min_{\mathbf{C}} \quad \mathrm{Tr}\left(\mathbf{C}^T \mathbf{L} \mathbf{C}\right) \tag{3}$$

where $\mathbf{L}$ is the (normalized) graph Laplacian of the input graph $\mathcal{G}$. It is worth noting that the columns of learning result $\mathbf{C}$ is equivalent to the eigenvectors of $\mathbf{L}$ associated with smallest k eigenvalues. To attack the fairness of spectral clustering with FATE, the attacker might attack a surrogate spectral clustering with different number of clusters k.

Model #3: Matrix factorization-based completion. Suppose we have a bipartite graph $\mathcal{G}$ with n_1 users, n_2 items and m interactions between users and items. Matrix factorization-based completion aims to learn two low-rank matrices an $n_1 \times z$ matrix $\mathbf{U}$ and an $n_2 \times z$ matrix $\mathbf{V}$ such that the following loss function will be minimized

$$\min_{\mathbf{U},\mathbf{V}} \quad \|\, \mathrm{proj}_{\Omega}\left(\mathbf{R} - \mathbf{U}\mathbf{V}^T\right) \|_F^2 + \lambda_1 \|\mathbf{U}\|_F^2 \lambda_2 + \|\mathbf{V}\|_F^2 \tag{4}$$

where $\mathbf{A} = \begin{pmatrix} \mathbf{0}_{n_1} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0}_{n_2} \end{pmatrix}$ with $\mathbf{0}_{n_1}$ being an $n_1 \times n_1$ square matrix filled with 0, $\Omega = \{(i,j) | (i,j) \text{ is observed}\}$ is the set of observed interaction between any user i and any item j, $\mathrm{proj}_{\Omega}(\mathbf{Z})[i,j]$ equals to $\mathbf{Z}[i,j]$ if $(i,j) \in \Omega$ and 0 otherwise, λ_1 and λ_2 are two hyperparameters for regularization. To attack the fairness of matrix factorization-based completion with FATE, the attacker could attack a surrogate model with different number of latent factors z.

Model #4: First-order LINE. It is a skip-gram based node embedding model. The key idea of first-order LINE is to map each node into a h-dimensional space such that the dot product of the embeddings of any two connected nodes will be small. To achieve this goal, first-order LINE essentially optimizes the following loss function

$$\max_{\mathbf{H}} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{A}[i,j] \left(\log g\left(\mathbf{H}[j,:]\mathbf{H}[i,:]^T\right) + k \mathbb{E}_{j' \sim P_n}[\log g\left(-\mathbf{H}[j',:]\mathbf{H}[i,:]^T\right)]\right) \tag{5}$$

where $\mathbf{H}$ is the embedding matrix with $\mathbf{H}[i,:]$ being the h-dimensional embedding of node i, $g(x) = 1/(1+e^{-x})$ is the sigmoid function, k is the number of negative samples and P_n is the distribution for negative sampling such that the sampling probability for node i is proportional to its degree $\deg_i$. For a victim first-order LINE, the attacker could attack a surrogate LINE (1st) with different dimension h in the embedding space and/or a different number of negative samples g.

Remarks. Note that, for a non-parameterized graph learning model (e.g., PageRank, spectral clustering, matrix completion, first-order LINE), we have $\Theta = \{\mathbf{Y}\}$ which is the set of learning results. For example, we have $\Theta = \{\mathbf{r}\}$ for PageRank, $\Theta = \{\mathbf{C}\}$ for spectral clustering, $\Theta = \{\mathbf{U}, \mathbf{V}\}$ and $\Theta = \{\mathbf{H}\}$ for LINE (1st). For parameterized graph learning models (e.g., GCN), Θ refers to the set of learnable weights, e.g., $\Theta = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}\}$ for an L-layer GCN.

B PSEUDOCODE OF FATE

Algorithm 1 summarizes the detailed steps on fairness attack with FATE. To be specific, after initialization (line 1), we pre-train the surrogate graph learning model (lines 4 – 6) and get the pre-trained surrogate model $\Theta^{(T)}$ as well as learning results $\mathbf{Y}^{(T)}$ (line 7). After that, we compute the meta gradient of the bias function (lines 8 – 11) and perform either discretized attack or continuous attack based on the interest of attacker (i.e., discretized poisoning attack in lines 12 – 15 or continuous poisoning attack in lines 16 – 18).

C EXPERIMENTAL SETTINGS

In this section, we provide more detailed information about the experimental settings. These include the hardware and software specifications, dataset descriptions, evaluation metrics as well as detailed parameter settings. In all experiments, we evaluate our proposed FATE in the task of semi-supervised node classification to answer the following questions:

Algorithm 1: FATE

Given : an undirected graph $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$, the set of training nodes $\mathcal{V}_{\text{train}}$, fairness-related auxiliary information matrix $\mathbf{F}$, total budget B, budget in step i δ_i, the bias function b, number of pre-training epochs T;

Find : the poisoned graph $\widetilde{\mathcal{G}}$;

poisoned graph $\widetilde{\mathcal{G}} \leftarrow \mathcal{G}$, cumulative budget $\Delta \leftarrow 0$, step counter $i \leftarrow 0$;
while $\Delta < B$ **do**
  $\nabla_{\widetilde{\mathcal{G}}} b \leftarrow 0$;
  for $t = 1$ *to* T **do**
    update $\Theta_{\text{sur}}^{(t)}$ to $\Theta_{\text{sur}}^{(t+1)}$ with a gradient-based optimizer (e.g., Adam);
  end
  get $\mathbf{Y}^{(T)}$ and $\Theta_{\text{sur}}^{(T)}$;
  compute meta-gradient $\nabla_{\mathcal{G}} b \leftarrow \nabla_{\Theta_{\text{sur}}^{(T)}} b \left(\mathbf{Y}, \Theta_{\text{sur}}^{(T)}, \mathbf{F}\right) \cdot \nabla_{\mathcal{G}} \Theta_{\text{sur}}^{(T)}$;
  if *attack the adjacency matrix* **then**
    compute the derivative $\nabla_{\widetilde{\mathbf{A}}} b \leftarrow \nabla_{\widetilde{\mathbf{A}}} b + \left(\nabla_{\widetilde{\mathbf{A}}} b\right)^T - \operatorname{diag}\left(\nabla_{\widetilde{\mathbf{A}}} b\right)$;
  end
  if *discretized poisoning attack* **then**
    compute the poisoning preference matrix $\nabla_{\widetilde{\mathbf{A}}}$ by Eq. equation **??**;
    select the edges to poison in $\nabla_{\widetilde{\mathbf{A}}}$ with budget δ_i by Eq. equation **??**;
    update the corresponding entries in $\widetilde{\mathcal{G}}$;
  else
    update $\widetilde{\mathcal{G}}$ by Eq. equation **??** with budget δ_i;
  end
  $\Delta \leftarrow \Delta + \delta_i$;
  $i \leftarrow i + 1$;
end
return $\widetilde{\mathcal{G}}$;

Q1. How effective is FATE in exacerbating bias under different perturbation rates?

Q2. How effective is FATE in maintaining node classification accuracy for deceptiveness under different perturbation rates?

Q3. Can we characterize the properties of edges perturbed by FATE?

C.1 Hardware and Software Specifications

All codes are programmed in Python 3.8.13 and PyTorch 1.12.1. All experiments are performed on a Linux server with 2 Intel Xeon Gold 6240R CPUs and 4 Nvidia Tesla V100 SXM2 GPUs, each of which has 32 GB memory.

C.2 Dataset Descriptions

We use three widely-used benchmark datasets for fair graph learning: Pokec-z, Pokec-n and Bail. For each dataset, we use a fixed random seed to split the dataset into training, validation and test sets with the split ratio being 50%, 25%, and 25%, respectively. The statistics of the datasets, including the number of nodes (# Nodes), the number of edges (# Edges), the number of features (# Features), the sensitive attribute (Sensitive Attr.) and the label (Label), are summarized in Table 1.

- Pokec-z and Pokec-n are two datasets collected from the Slovakian social network *Pokec*, each of which represents a sub-network of a province. Each node in these datasets is a user belonging to two major regions of the corresponding provinces, and each edge is the friendship relationship between two users. The sensitive attribute is the user region, and the label is the working field of a user.

- Bail is a similarity graph of criminal defendants during 1990 – 2009. Each node is a defendant during this time period. Two nodes are connected if they share similar past criminal records and demographics. The sensitive attribute is the race of the defendant, and the label is whether the defendant is on bail or not.

Table 1: Statistics of the datasets.

Dataset	Pokec-z	Pokec-n	Bail
# Nodes	$7,659$	$6,185$	$18,876$
# Edges	$20,550$	$15,321$	$311,870$
# Features	276	265	17
Sensitive Attr.	Region	Region	Race
Label	Working field	Working field	Bail decision

C.3 Evaluation Metrics

In our experiments, we aim to evaluate how effective FATE is in (1) attacking the fairness and (2) maintaining the utility of node classification.

To evaluate the performance of FATE in attacking the group fairness, we evaluate the effectiveness using Δ_{SP}, which is defined as follows.

$$\Delta_{\text{SP}} = |P[\widehat{y} = 1 \mid s = 1] - P[\widehat{y} = 1 \mid s = 0]| \tag{6}$$

where s is the sensitive attribute value of a node and $\widehat{y}$ is the ground-truth and predicted class labels of a node. While to evaluate the performance of FATE in attacking the individual fairness, we evaluate the effectiveness using the InFoRM bias (Bias) measure (**?**), which is defined as follows.

$$\text{Bias} = \sum_{i \in \mathcal{V}_{\text{test}}} \sum_{j \in \mathcal{V}_{\text{test}}} \mathbf{S}[i,j] \, \|\mathbf{Y}[i,:] - \mathbf{Y}[j,:]\|_F^2 \tag{7}$$

where $\mathcal{V}_{\text{test}}$ is the set of test nodes and $\mathbf{S}$ is the oracle pairwise node similarity matrix. The intuition of Eq. equation 7 is to measure the squared difference between the learning results of two test nodes, weighted by their pairwise similarity.

To evaluate the performance of FATE in maintaining the utility, we use micro F1 score (Micro F1), macro F1 score (Macro F1) and AUC score.

C.4 Detailed Parameter Settings

Poisoning the input graph. During poisoning attacks, we set a fixed random seed to control the randomness. The random seed used for each dataset in attacking group/individual fairness are summarized in Table 2.

- **Surrogate model training.** We run all methods with a perturbation rate from 0.05 to 0.25 with a step size of 0.05. For FA-GNN (**?**), we follow its official implementation and use the same surrogate 2-layer GCN (**?**) with 16 hidden dimensions for poisoning attack.[1]. The surrogate GCN in FA-GNN is trained for 500 epochs with a learning rate $1e-2$, weight decay $5e-4$, and dropout rate 0.5. For FATE, we use a 2-layer linear GCN (**?**) with 16 hidden dimensions for poisoning attacks. And the surrogate linear GCN in FATE is trained for 500 epochs with a learning rate $1e-2$, weight decay $5e-4$, and dropout rate 0.5.
- **Graph topology manipulation.** For Random and DICE, we use the implementations provided in the deeprobust package with the default parameters to add the adversarial edges.[2]. For FA-GNN, we add adversarial edges that connect two nodes with different class labels and different sensitive attributes, which provides the most promising performance as shown in (**?**). For FATE, suppose we poison the input graph in p $(p > 1)$ attacking steps. Then the per-iteration attacking budget in Algorithm 1 is set as $\delta_1 = 1$ and $\delta_i = \frac{r|\mathcal{E}|-1}{p-1}$, $\forall i \in \{2, \ldots, p\}$, where r is the perturbation rate

[1] `https://github.com/mengcao327/attack-gnn-fairness`
[2] `https://deeprobust.readthedocs.io/`

Table 2: Parameter settings on the random seed for all baseline methods in poisoning attacks (Random Seed) and the number of steps for poisoning attacks in FATE (Attacking Steps).

Dataset	Fairness Definition	Attacking Steps	Random Seed
Pokec-n	Statistical parity	3	25
	Individual fairness	3	45
Pokec-z	Statistical parity	3	25
	Individual fairness	5	15
Bail	Statistical parity	3	25
	Individual fairness	3	5

and $|\mathcal{E}|$ is the number of edges. Detailed choices of p for each dataset in attacking group/individual fairness are summarized in Table 2.

Training the victim model. We use a fixed list of random seed ($[0, 1, 2, 42, 100]$) to train each victim model 5 times and report the mean and standard deviation. Regarding the victim models in group fairness attacks, we train a 2-layer GCN (**?**) for 400 epochs and a 2-layer FairGNN (**?**) for 2000 epochs to evaluate the efficacy of fairness attacks. The hidden dimension, learning rate, weight decay and dropout rate of GCN and FairGNN are set to 128, $1e-3$, $1e-5$ and 0.5, respectively. The regularization parameters in FairGNN, namely α and β, are set to 100 and 1 for all datasets, respectively. Regarding the victim models in individual fairness attacks, we train a 2-layer GCN (**?**) and 2-layer InFoRM-GNN (**??**) for 400 epochs. The hidden dimension, learning rate, weight decay and dropout rate of GCN and InFoRM-GNN are set to 128, $1e-3$, $1e-5$ and 0.5, respectively. The regularization parameter in InFoRM-GNN is set to 0.1 for all datasets.

D Additional Experimental Results: Attacking Statistical Parity on Graph Neural Networks

A – FATE with FairGNN as the victim model. Here, we study how robust FairGNN is in fairness attacks against statistical parity with linear GCN as the surrogate model. Note that FairGNN is a fairness-aware graph neural network that leverages adversarial learning to ensure statistical parity.

Main results. Similar to Section **??**, for FATE, we conduct fairness attacks via both edge flipping (FATE-flip) and edge addition (FATE-add). For all other baseline methods, edges are only added. From Table 3, we have the following key observations: (1) Even though the surrogate model is linear GCN without fairness consideration, FairGNN, which ensures statistical parity on graph neural networks, cannot mitigate the bias caused by fairness attacks and is vulnerable to fairness attack. (2) FATE-flip and FATE-add are effective and the most deceptive method in fairness attacks. (3) DICE-S, FATE-flip, and FATE-add are all capable of successful fairness attacks. But FATE-flip and FATE-add have better utility than DICE-S, making the fairness attacks more deceptive. Both Random and FA-GNN fail in some cases (indicated by the underlined Δ_{SP} in both tables). In short, even when the victim model is FairGNN (a fair graph neural network), our proposed FATE framework are effective in fairness attacks while being the most deceptive (i.e., highest micro F1 score).

Effect of the perturbation rate. From Table 3, we can find out that: (1) Δ_{SP} tends to increase when the perturbation rate increases, indicating the effectiveness of FATE-flip and FATE-add for attacking fairness. (2) There is no clear correlation between the perturbation rate and the micro F1 scores of FATE-flip and FATE-add, meaning that they are deceptive in maintaining the utility. As a consequence, FATE is effective and deceptive in attacking fairness of FairGNN across different perturbation rates.

B – Performance evaluation under different utility metrics. Here we provide additional evaluation results of utility using macro F1 score and AUC score. From Tables 4 and 5, we can see that macro F1 scores and AUC scores are less impacted by different perturbation rates. Thus, it provide additional evidence that FATE can achieve deceptive fairness attacks by achieving comparable or even better utility on the semi-supervised node classification.

Table 3: Attacking statistical parity on FairGNN under different perturbation rates (Ptb.). FATE poisons the graph via both edge flipping (FATE-flip) and edge addition (FATE-add) while all other baselines poison the graph via edge addition. Higher is better (↑) for micro F1 score (Micro F1) and Δ_{SP} (bias). Bold font indicates the most deceptive fairness attack, i.e., increasing Δ_{SP} and highest micro F1. Underlined cell indicates the failure of fairness attack, i.e., decreasing Δ_{SP} after fairness attack.

Dataset	Ptb.	Random		DICE-S		FA-GNN		FATE-flip		FATE-add	
		Micro F1 (↑)	Δ_{SP} (↑)	Micro F1 (↑)	Δ_{SP} (↑)	Micro F1 (↑)	Δ_{SP} (↑)	Micro F1 (↑)	Δ_{SP} (↑)	Micro F1 (↑)	Δ_{SP} (↑)
Pokec-n	0.00	68.2 ± 0.4	6.7 ± 2.0	68.2 ± 0.4	6.7 ± 2.0	68.2 ± 0.4	6.7 ± 2.0	68.2 ± 0.4	6.7 ± 2.0	68.2 ± 0.4	6.7 ± 2.0
	0.05	67.4 ± 0.8	8.2 ± 2.5	66.9 ± 0.9	7.4 ± 1.7	$\underline{66.7 \pm 1.2}$	$\underline{2.8 \pm 1.3}$	$\mathbf{68.4 \pm 0.2}$	$\mathbf{8.9 \pm 1.8}$	$\mathbf{68.4 \pm 0.2}$	$\mathbf{8.9 \pm 1.8}$
	0.10	67.5 ± 0.5	8.3 ± 1.5	67.6 ± 0.3	8.4 ± 1.2	$\underline{66.6 \pm 0.5}$	$\underline{5.9 \pm 1.3}$	$\mathbf{68.5 \pm 0.4}$	$\mathbf{9.5 \pm 1.4}$	$\mathbf{68.5 \pm 0.4}$	$\mathbf{9.5 \pm 1.4}$
	0.15	65.9 ± 0.6	10.4 ± 2.3	67.3 ± 0.3	9.9 ± 2.4	64.8 ± 1.6	9.0 ± 3.3	$\mathbf{68.5 \pm 0.8}$	$\mathbf{10.5 \pm 2.6}$	$\mathbf{68.5 \pm 0.8}$	$\mathbf{10.5 \pm 2.6}$
	0.20	65.4 ± 0.5	10.0 ± 1.5	66.5 ± 0.4	9.0 ± 2.3	65.2 ± 0.2	11.6 ± 2.6	$\mathbf{68.3 \pm 0.3}$	$\mathbf{10.7 \pm 2.3}$	$\mathbf{68.3 \pm 0.3}$	$\mathbf{10.7 \pm 2.3}$
	0.25	65.8 ± 1.1	7.5 ± 1.9	66.5 ± 0.8	9.7 ± 3.0	64.8 ± 0.8	14.2 ± 2.3	$\mathbf{68.5 \pm 0.3}$	$\mathbf{9.1 \pm 3.6}$	$\mathbf{68.5 \pm 0.3}$	$\mathbf{9.1 \pm 3.6}$
Pokec-z	0.00	68.7 ± 0.3	7.0 ± 0.9	68.7 ± 0.3	7.0 ± 0.9	68.7 ± 0.3	7.0 ± 0.9	68.7 ± 0.3	7.0 ± 0.9	68.7 ± 0.3	7.0 ± 0.9
	0.05	67.3 ± 0.6	8.7 ± 2.8	68.0 ± 0.7	9.4 ± 4.1	$\underline{67.1 \pm 1.0}$	$\underline{1.7 \pm 1.3}$	$\mathbf{68.7 \pm 0.4}$	$\mathbf{8.0 \pm 0.9}$	$\mathbf{68.7 \pm 0.4}$	$\mathbf{8.0 \pm 0.9}$
	0.10	67.1 ± 0.2	8.6 ± 2.7	68.1 ± 0.5	8.2 ± 5.0	$\underline{65.9 \pm 0.8}$	$\underline{6.8 \pm 1.7}$	$\mathbf{68.5 \pm 0.5}$	$\mathbf{9.0 \pm 1.8}$	$\mathbf{68.5 \pm 0.5}$	$\mathbf{9.0 \pm 1.8}$
	0.15	66.8 ± 0.8	8.9 ± 2.2	67.6 ± 0.6	9.6 ± 3.4	64.9 ± 0.9	10.0 ± 1.7	$\mathbf{68.7 \pm 0.5}$	$\mathbf{9.5 \pm 2.2}$	$\mathbf{68.7 \pm 0.5}$	$\mathbf{9.5 \pm 2.2}$
	0.20	66.8 ± 0.7	8.6 ± 3.0	67.4 ± 0.7	9.1 ± 4.9	64.6 ± 0.8	14.2 ± 3.1	$\mathbf{68.8 \pm 0.2}$	$\mathbf{10.4 \pm 1.6}$	$\mathbf{68.8 \pm 0.2}$	$\mathbf{10.4 \pm 1.6}$
	0.25	66.4 ± 0.4	7.9 ± 2.8	67.1 ± 0.6	8.7 ± 4.3	64.0 ± 1.1	14.0 ± 2.0	$\mathbf{68.5 \pm 0.3}$	$\mathbf{10.3 \pm 2.1}$	$\mathbf{68.5 \pm 0.3}$	$\mathbf{10.3 \pm 2.1}$
Bail	0.00	93.9 ± 0.1	8.4 ± 0.2	93.9 ± 0.1	8.4 ± 0.2	93.9 ± 0.1	8.4 ± 0.2	93.9 ± 0.1	8.4 ± 0.2	93.9 ± 0.1	8.4 ± 0.2
	0.05	$\underline{90.6 \pm 1.2}$	$\underline{8.3 \pm 0.2}$	90.5 ± 1.0	8.9 ± 0.5	89.1 ± 2.0	10.8 ± 1.1	$\mathbf{93.6 \pm 0.1}$	$\mathbf{9.2 \pm 0.2}$	$\mathbf{93.6 \pm 0.1}$	$\mathbf{9.1 \pm 0.2}$
	0.10	90.1 ± 2.0	8.5 ± 0.6	90.1 ± 1.0	8.6 ± 0.2	87.3 ± 2.2	12.2 ± 1.2	$\mathbf{93.4 \pm 0.1}$	$\mathbf{9.3 \pm 0.2}$	$\mathbf{93.4 \pm 0.1}$	$\mathbf{9.3 \pm 0.2}$
	0.15	$\underline{90.0 \pm 2.0}$	$\underline{8.1 \pm 0.5}$	90.6 ± 1.7	9.5 ± 0.6	87.8 ± 2.0	10.9 ± 2.1	$\mathbf{93.3 \pm 0.1}$	$\mathbf{9.2 \pm 0.3}$	$\mathbf{93.3 \pm 0.1}$	$\mathbf{9.2 \pm 0.3}$
	0.20	$\underline{89.2 \pm 2.4}$	$\underline{8.4 \pm 0.7}$	90.0 ± 1.7	9.9 ± 0.6	86.0 ± 2.7	11.7 ± 2.4	$\mathbf{93.1 \pm 0.2}$	$\mathbf{9.3 \pm 0.3}$	93.0 ± 0.1	9.4 ± 0.2
	0.25	$\underline{88.8 \pm 2.3}$	$\underline{8.2 \pm 0.7}$	89.9 ± 1.8	9.6 ± 0.5	87.0 ± 1.9	8.5 ± 2.6	$\mathbf{93.0 \pm 0.1}$	$\mathbf{9.2 \pm 0.4}$	$\mathbf{93.0 \pm 0.2}$	$\mathbf{9.3 \pm 0.3}$

Table 4: Macro F1 score and AUC score of attacking statistical parity on GCN under different perturbation rates (Ptb.). FATE poisons the graph via both edge flipping (FATE-flip) and edge addition (FATE-add) while all other baselines poison the graph via edge addition. Higher is better (↑) for macro F1 score (Macro F1) and AUC score (AUC). Bold font indicates the highest macro F1 score or AUC score.

Dataset	Ptb.	Random		DICE-S		FA-GNN		FATE-flip		FATE-add	
		Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)
Pokec-n	0.00	65.3 ± 0.3	69.9 ± 0.5	65.3 ± 0.3	69.9 ± 0.5	65.3 ± 0.3	69.9 ± 0.5	65.3 ± 0.3	69.9 ± 0.5	65.3 ± 0.3	69.9 ± 0.5
	0.05	65.7 ± 0.3	$\mathbf{70.4 \pm 0.4}$	65.4 ± 0.3	70.3 ± 0.3	64.9 ± 0.2	$\mathbf{70.4 \pm 0.2}$	$\mathbf{66.0 \pm 0.3}$	70.3 ± 0.6	$\mathbf{66.0 \pm 0.3}$	70.3 ± 0.6
	0.10	64.6 ± 0.4	69.6 ± 0.3	65.7 ± 0.2	70.2 ± 0.2	64.1 ± 0.3	70.0 ± 0.1	$\mathbf{66.1 \pm 0.6}$	$\mathbf{70.4 \pm 0.6}$	$\mathbf{66.1 \pm 0.6}$	$\mathbf{70.4 \pm 0.6}$
	0.15	65.1 ± 0.4	69.6 ± 0.1	64.9 ± 0.3	69.0 ± 0.3	64.3 ± 0.6	69.1 ± 0.5	$\mathbf{66.1 \pm 0.2}$	$\mathbf{70.6 \pm 0.6}$	$\mathbf{66.1 \pm 0.2}$	$\mathbf{70.6 \pm 0.6}$
	0.20	64.5 ± 0.5	69.1 ± 0.1	64.2 ± 0.3	68.7 ± 0.4	63.5 ± 0.2	68.0 ± 0.2	$\mathbf{66.4 \pm 0.3}$	$\mathbf{70.7 \pm 0.4}$	$\mathbf{66.4 \pm 0.3}$	$\mathbf{70.7 \pm 0.4}$
	0.25	64.5 ± 0.6	68.8 ± 0.1	63.7 ± 0.2	68.8 ± 0.2	65.0 ± 0.2	69.5 ± 0.3	$\mathbf{66.3 \pm 0.3}$	$\mathbf{70.6 \pm 0.6}$	$\mathbf{66.3 \pm 0.3}$	$\mathbf{70.6 \pm 0.6}$
Pokec-z	0.00	68.2 ± 0.4	75.1 ± 0.3	68.2 ± 0.4	75.1 ± 0.3	68.2 ± 0.4	75.1 ± 0.3	68.2 ± 0.4	75.1 ± 0.3	68.2 ± 0.4	75.1 ± 0.3
	0.05	68.5 ± 0.4	74.5 ± 0.4	$\mathbf{68.7 \pm 0.3}$	$\mathbf{75.4 \pm 0.4}$	67.9 ± 0.3	74.5 ± 0.2	68.6 ± 0.4	75.2 ± 0.4	68.6 ± 0.4	75.2 ± 0.4
	0.10	68.5 ± 0.3	74.8 ± 0.3	67.6 ± 0.2	74.5 ± 0.3	67.5 ± 0.5	73.8 ± 0.3	$\mathbf{68.6 \pm 0.6}$	$\mathbf{75.2 \pm 0.3}$	$\mathbf{68.6 \pm 0.6}$	$\mathbf{75.2 \pm 0.3}$
	0.15	67.8 ± 0.3	74.4 ± 0.3	67.6 ± 0.4	74.1 ± 0.4	66.1 ± 0.6	72.7 ± 0.2	$\mathbf{68.9 \pm 0.7}$	$\mathbf{75.3 \pm 0.2}$	$\mathbf{68.9 \pm 0.7}$	$\mathbf{75.3 \pm 0.2}$
	0.20	68.2 ± 0.4	74.5 ± 0.6	66.8 ± 0.5	73.6 ± 0.3	66.1 ± 0.2	71.9 ± 0.1	$\mathbf{68.4 \pm 0.5}$	$\mathbf{75.1 \pm 0.3}$	$\mathbf{68.4 \pm 0.5}$	$\mathbf{75.1 \pm 0.3}$
	0.25	68.0 ± 0.4	74.0 ± 0.4	67.1 ± 0.7	74.4 ± 0.3	65.3 ± 0.6	71.2 ± 0.3	$\mathbf{68.4 \pm 1.1}$	$\mathbf{74.4 \pm 1.4}$	$\mathbf{68.4 \pm 1.1}$	$\mathbf{74.4 \pm 1.4}$
Bail	0.00	92.3 ± 0.2	97.4 ± 0.1	92.3 ± 0.2	97.4 ± 0.1	92.3 ± 0.2	97.4 ± 0.1	92.3 ± 0.2	97.4 ± 0.1	92.3 ± 0.2	97.4 ± 0.1
	0.05	$\mathbf{92.0 \pm 0.2}$	95.3 ± 0.2	91.4 ± 0.3	95.1 ± 0.4	90.8 ± 0.1	94.4 ± 0.2	91.8 ± 0.1	$\mathbf{97.1 \pm 0.1}$	91.7 ± 0.1	$\mathbf{97.1 \pm 0.2}$
	0.10	91.4 ± 0.2	94.7 ± 0.3	91.4 ± 0.3	94.7 ± 0.4	89.5 ± 0.1	93.5 ± 0.1	$\mathbf{91.6 \pm 0.2}$	$\mathbf{96.9 \pm 0.1}$	$\mathbf{91.6 \pm 0.2}$	$\mathbf{96.9 \pm 0.1}$
	0.15	91.1 ± 0.2	94.2 ± 0.2	91.2 ± 0.2	94.5 ± 0.2	88.7 ± 0.3	92.5 ± 0.2	91.4 ± 0.2	$\mathbf{96.9 \pm 0.1}$	$\mathbf{91.5 \pm 0.1}$	$\mathbf{96.9 \pm 0.1}$
	0.20	90.7 ± 0.2	94.1 ± 0.1	90.9 ± 0.2	94.4 ± 0.3	88.4 ± 0.1	92.2 ± 0.1	91.3 ± 0.2	$\mathbf{96.8 \pm 0.1}$	$\mathbf{91.4 \pm 0.2}$	$\mathbf{96.8 \pm 0.1}$
	0.25	90.4 ± 0.2	93.4 ± 0.3	90.6 ± 0.3	94.3 ± 0.3	88.5 ± 0.2	92.0 ± 0.1	91.2 ± 0.1	$\mathbf{96.8 \pm 0.1}$	$\mathbf{91.3 \pm 0.2}$	$\mathbf{96.8 \pm 0.1}$

Table 5: Macro F1 score and AUC score of attacking statistical parity on FairGNN under different perturbation rates (Ptb.). FATE poisons the graph via both edge flipping (FATE-flip) and edge addition (FATE-add) while all other baselines poison the graph via edge addition. Higher is better (↑) for macro F1 score (Macro F1) and AUC score (AUC). Bold font indicates the highest macro F1 score or AUC score.

Dataset	Ptb.	Random		DICE-S		FA-GNN		FATE-flip		FATE-add	
		Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)
Pokec-n	0.00	65.6 ± 0.3	70.4 ± 0.5	65.6 ± 0.3	70.4 ± 0.5	65.6 ± 0.3	70.4 ± 0.5	65.6 ± 0.3	70.4 ± 0.5	65.6 ± 0.3	70.4 ± 0.5
	0.05	64.3 ± 0.6	68.3 ± 1.1	64.5 ± 0.4	69.5 ± 0.8	63.6 ± 0.7	68.2 ± 0.5	$\mathbf{65.8 \pm 0.5}$	$\mathbf{70.7 \pm 0.4}$	$\mathbf{65.8 \pm 0.5}$	$\mathbf{70.7 \pm 0.4}$
	0.10	63.8 ± 0.2	67.3 ± 1.1	64.3 ± 0.7	69.6 ± 0.4	63.9 ± 0.4	68.3 ± 0.2	$\mathbf{66.0 \pm 0.7}$	$\mathbf{70.8 \pm 0.5}$	$\mathbf{66.0 \pm 0.7}$	$\mathbf{70.8 \pm 0.5}$
	0.15	63.5 ± 0.2	67.8 ± 0.4	64.1 ± 0.7	68.5 ± 0.4	63.1 ± 0.6	67.2 ± 0.5	$\mathbf{65.8 \pm 1.0}$	$\mathbf{70.8 \pm 0.5}$	$\mathbf{65.8 \pm 1.0}$	$\mathbf{70.8 \pm 0.5}$
	0.20	63.1 ± 0.6	67.8 ± 1.1	62.4 ± 1.5	67.5 ± 1.1	62.3 ± 0.6	66.7 ± 0.9	$\mathbf{65.7 \pm 0.7}$	$\mathbf{70.4 \pm 0.5}$	$\mathbf{65.7 \pm 0.7}$	$\mathbf{70.4 \pm 0.5}$
	0.25	62.4 ± 0.3	66.8 ± 0.8	62.4 ± 1.6	67.2 ± 0.9	62.4 ± 1.4	67.6 ± 1.3	$\mathbf{65.1 \pm 1.2}$	$\mathbf{70.1 \pm 0.5}$	$\mathbf{65.1 \pm 1.2}$	$\mathbf{70.1 \pm 0.5}$
Pokec-z	0.00	68.4 ± 0.4	75.1 ± 0.3	68.4 ± 0.4	75.1 ± 0.3	68.4 ± 0.4	75.1 ± 0.3	68.4 ± 0.4	75.1 ± 0.3	68.4 ± 0.4	75.1 ± 0.3
	0.05	66.3 ± 0.9	73.5 ± 0.9	67.2 ± 0.7	73.9 ± 1.5	66.5 ± 1.4	72.6 ± 1.4	$\mathbf{68.4 \pm 0.4}$	$\mathbf{74.7 \pm 0.9}$	$\mathbf{68.4 \pm 0.4}$	$\mathbf{74.7 \pm 0.9}$
	0.10	66.0 ± 0.7	72.9 ± 1.1	67.1 ± 0.5	73.4 ± 0.2	65.2 ± 0.9	71.3 ± 1.7	$\mathbf{68.2 \pm 0.8}$	$\mathbf{75.3 \pm 0.8}$	$\mathbf{68.2 \pm 0.8}$	$\mathbf{75.3 \pm 0.8}$
	0.15	66.0 ± 0.8	71.8 ± 2.1	66.5 ± 0.9	73.4 ± 0.6	63.4 ± 1.5	70.0 ± 1.8	$\mathbf{68.3 \pm 0.5}$	$\mathbf{75.2 \pm 0.6}$	$\mathbf{68.3 \pm 0.5}$	$\mathbf{75.2 \pm 0.6}$
	0.20	65.6 ± 0.9	71.9 ± 1.4	66.4 ± 1.0	73.0 ± 0.8	63.7 ± 0.9	68.9 ± 1.6	$\mathbf{68.3 \pm 0.3}$	$\mathbf{75.5 \pm 0.3}$	$\mathbf{68.3 \pm 0.3}$	$\mathbf{75.5 \pm 0.3}$
	0.25	65.0 ± 0.7	71.2 ± 1.7	66.3 ± 1.0	73.3 ± 0.8	62.8 ± 1.8	69.4 ± 1.5	$\mathbf{68.0 \pm 0.5}$	$\mathbf{75.3 \pm 0.3}$	$\mathbf{68.0 \pm 0.5}$	$\mathbf{75.3 \pm 0.3}$
Bail	0.00	93.3 ± 0.2	97.4 ± 0.1	93.3 ± 0.2	97.4 ± 0.1	93.3 ± 0.2	97.4 ± 0.1	93.3 ± 0.2	97.4 ± 0.1	93.3 ± 0.2	97.4 ± 0.1
	0.05	89.5 ± 1.5	92.8 ± 1.8	89.5 ± 1.1	92.3 ± 1.7	87.8 ± 2.2	91.2 ± 1.8	$\mathbf{93.0 \pm 0.1}$	$\mathbf{97.3 \pm 0.1}$	$\mathbf{93.0 \pm 0.1}$	$\mathbf{97.3 \pm 0.1}$
	0.10	89.1 ± 2.2	92.7 ± 2.5	88.8 ± 1.3	92.3 ± 1.6	85.6 ± 2.7	90.5 ± 1.7	$\mathbf{92.7 \pm 0.1}$	$\mathbf{97.1 \pm 0.1}$	$\mathbf{92.7 \pm 0.1}$	$\mathbf{97.1 \pm 0.1}$
	0.15	88.8 ± 2.2	92.4 ± 2.5	89.6 ± 1.9	92.8 ± 2.2	86.1 ± 2.4	90.3 ± 2.2	$\mathbf{92.6 \pm 0.1}$	$\mathbf{97.0 \pm 0.1}$	$\mathbf{92.6 \pm 0.1}$	$\mathbf{97.0 \pm 0.1}$
	0.20	87.8 ± 2.8	91.6 ± 2.5	88.9 ± 1.8	92.2 ± 1.6	84.1 ± 3.0	89.0 ± 1.5	$\mathbf{92.5 \pm 0.2}$	$\mathbf{97.0 \pm 0.1}$	92.3 ± 0.1	$\mathbf{97.0 \pm 0.1}$
	0.25	87.5 ± 2.6	91.5 ± 2.6	88.7 ± 2.1	92.5 ± 2.3	85.1 ± 2.3	89.6 ± 1.3	$\mathbf{92.3 \pm 0.1}$	$\mathbf{97.0 \pm 0.1}$	$\mathbf{92.3 \pm 0.2}$	$\mathbf{97.0 \pm 0.1}$

E ADDITIONAL EXPERIMENTAL RESULTS: ATTACKING INDIVIDUAL FAIRNESS ON GRAPH NEURAL NETWORKS

A – FATE with InFoRM-GNN as the victim model. InFoRM-GNN is an individually fair graph neural network that ensures individual fairness through regularizing the individual bias measure defined in Section **??**. Here, we study how robust InFoRM-GNN is in fairness attacks against individual fairness with linear GCN as the surrogate model.

Main results. We attack individual fairness using FATE via both edge flipping (FATE-flip) and edge addition (FATE-add), whereas edges are only added for all other baseline methods. From Table 6, we can see that: (1) for Pokec-n and Pokec-z, FATE-flip and FATE-add are effective: they are the only methods that could consistently attack individual fairness across different perturbation rates; FATE-flip and FATE-add are deceptive by achieving comparable or higher micro F1 scores compared with the micro F1 score on the benign graph (when perturbation rate is 0.00). (2) For Bail, almost all methods fail the fairness attacks, except for FA-GNN with perturbation rates 0.20 and 0.25. A possible reason is that the adjacency matrix $\mathbf{A}$ of *Bail* is essentially a similarity graph, which causes pairwise node similarity matrix $\mathbf{S}$ being close to the adjacency matrix $\mathbf{A}$. Even though FATE and other baseline methods add adversarial edges to attack individual fairness, regularizing the individual bias defined by $\mathbf{S}$ (a) not only helps to ensure individual fairness (b) but also provide useful supervision signal in learning a representative node representation due to the closeness between $\mathbf{S}$ and $\mathbf{A}$. (3) Compared with the results in Table **??** where GCN is the victim model, InFoRM-GNN is more robust against fairness attacks against individual fairness due to smaller individual bias in Table 6.

Effect of the perturbation rate. From Table 6, we can see that FATE can always achieve comparable or even better micro F1 scores across different perturbation rates. In the meanwhile, the correlation between the perturbation rate and the individual bias is relatively weak. One possible reason is that the individual bias is computed using the pairwise node similarity matrix, which is not impacted by poisoning the adjacency matrix. Though poisoning the adjacency matrix could affect the learning results, the goal of achieving deceptive fairness attacks (i.e., the lower-level optimization problem in FATE) may not cause the learning results obtained by training on the benign graph to deviate much from the learning results obtained by training on the poisoned graph. Consequently, a higher perturbation rate may have less impact on the computation of individual bias.

Table 6: Attacking individual fairness on InFoRM-GNN under different perturbation rates (Ptb.). FATE poisons the graph via both edge flipping (FATE-flip) and edge addition (FATE-add) while all other baselines poison the graph via edge addition. Higher is better (↑) for micro F1 score (Micro F1) and InFoRM bias (Bias). Bold font indicates the most deceptive fairness attack, i.e., increasing bias and highest micro F1. Underlined cell indicates the failure of fairness attack, i.e., decreasing bias after fairness attack.

Dataset	Ptb.	Random		DICE-S		FA-GNN		FATE-flip		FATE-add	
		Micro F1 (↑)	Bias (↑)	Micro F1 (↑)	Bias (↑)	Micro F1 (↑)	Bias (↑)	Micro F1 (↑)	Bias (↑)	Micro F1 (↑)	Bias (↑)
Pokec-n	0.00	68.0 ± 0.4	0.5 ± 0.1	68.0 ± 0.4	0.5 ± 0.1	68.0 ± 0.4	0.5 ± 0.1	68.0 ± 0.4	0.5 ± 0.1	68.0 ± 0.4	0.5 ± 0.1
	0.05	67.3 ± 0.5	0.5 ± 0.0	68.0 ± 0.4	0.5 ± 0.1	68.3 ± 0.2	0.5 ± 0.0	$\mathbf{68.4 \pm 0.4}$	$\mathbf{0.6 \pm 0.1}$	68.3 ± 0.4	0.5 ± 0.1
	0.10	67.0 ± 0.2	0.5 ± 0.1	67.4 ± 0.4	0.5 ± 0.1	67.2 ± 0.2	0.4 ± 0.0	68.3 ± 0.6	0.5 ± 0.1	$\mathbf{68.4 \pm 0.5}$	$\mathbf{0.6 \pm 0.1}$
	0.15	66.7 ± 0.5	0.5 ± 0.1	67.7 ± 0.4	0.4 ± 0.1	66.1 ± 0.2	0.4 ± 0.0	$\mathbf{68.3 \pm 0.6}$	$\mathbf{0.6 \pm 0.1}$	68.1 ± 0.7	0.6 ± 0.1
	0.20	66.9 ± 0.3	0.4 ± 0.1	67.2 ± 0.2	0.5 ± 0.1	66.5 ± 0.2	0.4 ± 0.0	67.9 ± 0.8	0.5 ± 0.1	$\mathbf{68.1 \pm 0.7}$	$\mathbf{0.6 \pm 0.1}$
	0.25	66.6 ± 0.5	0.5 ± 0.0	66.7 ± 0.6	0.5 ± 0.1	65.1 ± 0.2	0.4 ± 0.0	$\mathbf{68.7 \pm 0.3}$	$\mathbf{0.6 \pm 0.0}$	68.5 ± 0.8	0.6 ± 0.1
Pokec-z	0.00	68.4 ± 0.5	0.5 ± 0.0	68.4 ± 0.5	0.5 ± 0.0	68.4 ± 0.5	0.5 ± 0.0	68.4 ± 0.5	0.5 ± 0.0	68.4 ± 0.5	0.5 ± 0.0
	0.05	68.9 ± 0.2	0.6 ± 0.1	68.9 ± 0.5	0.5 ± 0.1	68.1 ± 0.7	0.5 ± 0.1	68.7 ± 0.7	0.7 ± 0.1	$\mathbf{68.9 \pm 0.5}$	$\mathbf{0.6 \pm 0.0}$
	0.10	67.9 ± 0.2	0.6 ± 0.1	$\mathbf{69.0 \pm 0.1}$	$\mathbf{0.6 \pm 0.1}$	68.0 ± 0.6	0.5 ± 0.0	68.9 ± 0.6	0.6 ± 0.0	68.8 ± 0.6	0.6 ± 0.0
	0.15	67.6 ± 0.3	0.6 ± 0.1	68.2 ± 0.5	0.6 ± 0.1	66.8 ± 0.3	0.5 ± 0.1	$\mathbf{69.1 \pm 0.5}$	$\mathbf{0.6 \pm 0.0}$	69.0 ± 0.7	0.6 ± 0.1
	0.20	67.7 ± 0.5	0.6 ± 0.1	68.5 ± 0.2	0.5 ± 0.0	66.4 ± 0.6	0.4 ± 0.1	69.1 ± 0.2	0.6 ± 0.0	$\mathbf{69.3 \pm 0.3}$	$\mathbf{0.6 \pm 0.0}$
	0.25	66.8 ± 0.4	0.5 ± 0.1	68.5 ± 0.2	0.5 ± 0.0	65.3 ± 0.4	0.4 ± 0.0	68.9 ± 0.7	0.6 ± 0.0	$\mathbf{69.4 \pm 0.4}$	$\mathbf{0.6 \pm 0.0}$
Bail	0.00	92.8 ± 0.1	1.7 ± 0.1	92.8 ± 0.1	1.7 ± 0.1	92.8 ± 0.1	1.7 ± 0.1	92.8 ± 0.1	1.7 ± 0.1	92.8 ± 0.1	1.7 ± 0.1
	0.05	91.9 ± 0.1	0.4 ± 0.0	92.1 ± 0.1	1.7 ± 0.0	91.3 ± 0.1	1.5 ± 0.1	92.8 ± 0.3	1.7 ± 0.1	92.7 ± 0.1	1.6 ± 0.1
	0.10	91.7 ± 0.1	0.3 ± 0.0	92.0 ± 0.1	1.6 ± 0.1	90.4 ± 0.2	1.5 ± 0.1	92.8 ± 0.1	1.6 ± 0.0	92.8 ± 0.1	1.6 ± 0.0
	0.15	91.5 ± 0.1	0.3 ± 0.0	91.9 ± 0.1	1.6 ± 0.1	90.0 ± 0.1	1.7 ± 0.1	92.8 ± 0.0	1.6 ± 0.1	92.8 ± 0.1	1.6 ± 0.0
	0.20	91.5 ± 0.1	0.3 ± 0.0	91.8 ± 0.1	1.7 ± 0.0	$\mathbf{89.1 \pm 0.1}$	$\mathbf{1.7 \pm 0.1}$	92.8 ± 0.1	1.6 ± 0.0	92.7 ± 0.1	1.5 ± 0.1
	0.25	91.1 ± 0.2	0.3 ± 0.0	91.5 ± 0.1	1.6 ± 0.0	$\mathbf{88.9 \pm 0.1}$	$\mathbf{1.8 \pm 0.1}$	92.6 ± 0.1	1.6 ± 0.1	92.7 ± 0.0	1.6 ± 0.1

B – Performance evaluation under different utility metrics. Similar to Appendix D, we provide additional results on evaluating the utility of FATE in attacking individual fairness with macro F1 score and AUC score. From Tables 7 and 8, we can draw a conclusion that FATE can achieve comparable or even better macro F1 scores and AUC scores for both GCN and InFoRM-GNN across different perturbation rates. It further proves the ability of FATE on deceptive fairness attacks in the task of semi-supervised node classification.

Table 7: Macro F1 score and AUC score of attacking individual fairness on GCN under different perturbation rates (Ptb.). FATE poisons the graph via both edge flipping (FATE-flip) and edge addition (FATE-add) while all other baselines poison the graph via edge addition. Higher is better (↑) for macro F1 score (Macro F1) and AUC score (AUC). Bold font indicates the highest macro F1 score or AUC score.

Dataset	Ptb.	Random		DICE-S		FA-GNN		FATE-flip		FATE-add	
		Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)
Pokec-n	0.00	65.3 ± 0.3	69.9 ± 0.5	65.3 ± 0.3	69.9 ± 0.5	65.3 ± 0.3	69.9 ± 0.5	65.3 ± 0.3	69.9 ± 0.5	65.3 ± 0.3	69.9 ± 0.5
	0.05	65.2 ± 0.3	70.1 ± 0.2	65.7 ± 0.3	70.2 ± 0.2	65.6 ± 0.6	$\mathbf{71.1 \pm 0.2}$	$\mathbf{65.7 \pm 0.4}$	70.1 ± 0.6	65.5 ± 0.3	70.2 ± 0.8
	0.10	65.2 ± 0.3	69.6 ± 0.5	64.7 ± 0.5	69.9 ± 0.2	65.4 ± 0.6	70.2 ± 0.3	65.5 ± 0.3	70.2 ± 0.7	$\mathbf{65.8 \pm 0.5}$	$\mathbf{70.7 \pm 0.6}$
	0.15	65.4 ± 0.2	69.4 ± 0.3	64.9 ± 0.2	70.1 ± 0.4	64.6 ± 0.2	69.4 ± 0.1	$\mathbf{65.6 \pm 0.4}$	$\mathbf{70.0 \pm 0.5}$	65.4 ± 0.1	69.8 ± 0.7
	0.20	64.9 ± 0.2	69.6 ± 0.3	65.1 ± 0.4	70.2 ± 0.3	63.7 ± 0.5	69.0 ± 0.1	65.2 ± 0.3	69.7 ± 0.6	$\mathbf{65.6 \pm 0.6}$	$\mathbf{70.2 \pm 0.7}$
	0.25	64.7 ± 0.1	69.4 ± 0.2	64.1 ± 0.1	69.4 ± 0.2	63.3 ± 0.5	68.4 ± 0.3	65.4 ± 0.6	69.7 ± 0.7	$\mathbf{65.6 \pm 0.8}$	$\mathbf{69.8 \pm 0.8}$
Pokec-z	0.00	68.2 ± 0.4	75.1 ± 0.3	68.2 ± 0.4	75.1 ± 0.3	68.2 ± 0.4	75.1 ± 0.3	68.2 ± 0.4	75.1 ± 0.3	68.2 ± 0.4	75.1 ± 0.3
	0.05	$\mathbf{68.7 \pm 0.4}$	75.0 ± 0.4	$\mathbf{68.7 \pm 0.6}$	75.2 ± 0.4	68.0 ± 0.4	75.1 ± 0.5	68.5 ± 0.5	$\mathbf{75.4 \pm 0.2}$	68.5 ± 0.3	75.2 ± 0.4
	0.10	68.5 ± 0.1	75.1 ± 0.5	$\mathbf{68.9 \pm 0.2}$	75.3 ± 0.1	67.9 ± 0.6	74.4 ± 0.5	68.8 ± 0.5	75.5 ± 0.3	68.8 ± 0.4	$\mathbf{75.6 \pm 0.2}$
	0.15	67.5 ± 0.4	74.4 ± 0.3	67.9 ± 0.3	73.8 ± 0.1	66.8 ± 0.4	72.6 ± 0.2	68.4 ± 0.5	75.5 ± 0.4	$\mathbf{68.8 \pm 0.7}$	$\mathbf{75.6 \pm 0.3}$
	0.20	67.5 ± 0.4	74.7 ± 0.4	67.7 ± 0.3	74.7 ± 0.2	66.1 ± 0.1	71.8 ± 0.2	68.7 ± 0.5	75.5 ± 0.3	$\mathbf{69.0 \pm 0.4}$	$\mathbf{75.6 \pm 0.3}$
	0.25	67.2 ± 0.3	74.1 ± 0.3	68.0 ± 0.5	74.7 ± 0.2	64.8 ± 0.4	70.5 ± 0.4	68.9 ± 0.3	75.6 ± 0.2	$\mathbf{69.1 \pm 0.3}$	$\mathbf{75.7 \pm 0.3}$
Bail	0.00	92.3 ± 0.2	97.4 ± 0.1	92.3 ± 0.2	97.4 ± 0.1	92.3 ± 0.2	97.4 ± 0.1	92.3 ± 0.2	97.4 ± 0.1	92.3 ± 0.2	97.4 ± 0.1
	0.05	91.2 ± 0.3	94.8 ± 0.2	91.4 ± 0.2	95.0 ± 0.3	90.3 ± 0.2	94.1 ± 0.2	$\mathbf{92.3 \pm 0.4}$	$\mathbf{97.3 \pm 0.1}$	92.1 ± 0.3	$\mathbf{97.3 \pm 0.1}$
	0.10	90.6 ± 0.1	94.2 ± 0.3	91.3 ± 0.2	94.9 ± 0.4	89.1 ± 0.1	92.9 ± 0.3	$\mathbf{92.3 \pm 0.1}$	$\mathbf{97.3 \pm 0.4}$	92.2 ± 0.2	$\mathbf{97.3 \pm 0.1}$
	0.15	90.3 ± 0.1	94.1 ± 0.2	91.3 ± 0.3	94.7 ± 0.3	88.6 ± 0.2	92.4 ± 0.3	$\mathbf{92.4 \pm 0.1}$	$\mathbf{97.3 \pm 0.0}$	92.3 ± 0.2	$\mathbf{97.3 \pm 0.1}$
	0.20	90.2 ± 0.0	93.9 ± 0.1	90.9 ± 0.2	94.2 ± 0.2	87.9 ± 0.2	91.8 ± 0.2	$\mathbf{92.4 \pm 0.1}$	$\mathbf{97.3 \pm 0.0}$	$\mathbf{92.4 \pm 0.2}$	$\mathbf{97.3 \pm 0.1}$
	0.25	90.9 ± 0.1	93.5 ± 0.2	90.5 ± 0.1	94.1 ± 0.4	87.6 ± 0.1	91.6 ± 0.2	$\mathbf{92.2 \pm 0.2}$	97.2 ± 0.1	$\mathbf{92.2 \pm 0.2}$	$\mathbf{97.3 \pm 0.1}$

Table 8: Macro F1 score and AUC score of attacking individual fairness on InFoRM-GNN under different perturbation rates (Ptb.). FATE poisons the graph via both edge flipping (FATE-flip) and edge addition (FATE-add) while all other baselines poison the graph via edge addition. Higher is better (↑) for macro F1 score (Macro F1) and AUC score (AUC). Bold font indicates the highest macro F1 score or AUC score.

Dataset	Ptb.	Random		DICE-S		FA-GNN		FATE-flip		FATE-add	
		Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)	Macro F1 (↑)	AUC (↑)
Pokec-n	0.00	65.4 ± 0.4	70.5 ± 0.8	65.4 ± 0.4	70.5 ± 0.8	65.4 ± 0.4	70.5 ± 0.8	65.4 ± 0.4	70.5 ± 0.8	65.4 ± 0.4	70.5 ± 0.8
	0.05	65.1 ± 0.3	69.9 ± 0.2	65.7 ± 0.1	70.3 ± 0.1	65.6 ± 0.3	$\mathbf{70.8 \pm 0.1}$	$\mathbf{65.9 \pm 0.4}$	70.7 ± 0.8	65.8 ± 0.5	70.5 ± 0.9
	0.10	64.9 ± 0.2	69.6 ± 0.5	65.2 ± 0.4	70.2 ± 0.2	64.8 ± 0.4	69.8 ± 0.3	65.8 ± 0.5	70.3 ± 1.0	$\mathbf{66.0 \pm 0.4}$	$\mathbf{70.9 \pm 1.1}$
	0.15	64.8 ± 0.4	69.6 ± 0.4	65.1 ± 0.3	70.0 ± 0.5	64.4 ± 0.1	69.2 ± 0.3	65.7 ± 0.6	$\mathbf{70.3 \pm 0.7}$	$\mathbf{65.8 \pm 0.4}$	$\mathbf{70.3 \pm 0.9}$
	0.20	65.1 ± 0.2	69.5 ± 0.3	64.9 ± 0.5	69.9 ± 0.2	63.4 ± 0.4	69.0 ± 0.2	65.5 ± 0.8	70.2 ± 0.9	$\mathbf{65.6 \pm 0.6}$	$\mathbf{70.5 \pm 0.7}$
	0.25	64.6 ± 0.3	69.6 ± 0.2	64.5 ± 0.3	69.4 ± 0.2	63.6 ± 0.3	68.6 ± 0.2	$\mathbf{66.0 \pm 0.5}$	$\mathbf{70.8 \pm 0.3}$	65.9 ± 0.5	70.4 ± 0.8
Pokec-z	0.00	68.3 ± 0.4	75.2 ± 0.2	68.3 ± 0.4	75.2 ± 0.2	68.3 ± 0.4	75.2 ± 0.2	68.3 ± 0.4	75.2 ± 0.2	68.3 ± 0.4	75.2 ± 0.2
	0.05	68.6 ± 0.2	75.1 ± 0.3	$\mathbf{68.9 \pm 0.4}$	$\mathbf{75.5 \pm 0.2}$	67.8 ± 0.6	75.0 ± 0.3	68.6 ± 0.7	75.1 ± 0.6	68.7 ± 0.4	75.4 ± 0.3
	0.10	67.6 ± 0.2	74.3 ± 0.4	$\mathbf{68.9 \pm 0.2}$	75.3 ± 0.3	67.7 ± 0.6	73.9 ± 0.6	68.6 ± 0.6	$\mathbf{75.6 \pm 0.3}$	68.6 ± 0.6	75.5 ± 0.3
	0.15	67.2 ± 0.3	74.1 ± 0.4	67.9 ± 0.4	74.4 ± 0.3	66.7 ± 0.3	72.3 ± 0.1	$\mathbf{68.9 \pm 0.4}$	$\mathbf{75.4 \pm 0.4}$	$\mathbf{68.9 \pm 0.6}$	$\mathbf{75.4 \pm 0.4}$
	0.20	67.3 ± 0.6	74.4 ± 0.4	68.3 ± 0.1	75.1 ± 0.3	66.0 ± 0.5	71.7 ± 0.2	69.0 ± 0.2	$\mathbf{75.5 \pm 0.2}$	$\mathbf{69.2 \pm 0.4}$	75.4 ± 0.4
	0.25	66.3 ± 0.4	73.9 ± 0.4	68.2 ± 0.3	74.8 ± 0.1	65.0 ± 0.5	70.8 ± 0.2	68.8 ± 0.7	75.6 ± 0.3	$\mathbf{69.3 \pm 0.4}$	$\mathbf{75.8 \pm 0.1}$
Bail	0.00	91.9 ± 0.1	97.2 ± 0.0	91.9 ± 0.1	97.2 ± 0.0	91.9 ± 0.1	97.2 ± 0.0	91.9 ± 0.1	97.2 ± 0.0	91.9 ± 0.1	97.2 ± 0.0
	0.05	91.0 ± 0.1	94.2 ± 0.2	91.2 ± 0.1	95.0 ± 0.1	90.4 ± 0.1	94.2 ± 0.1	$\mathbf{92.0 \pm 0.0}$	$\mathbf{97.1 \pm 0.1}$	91.9 ± 0.2	97.0 ± 0.2
	0.10	90.7 ± 0.2	93.9 ± 0.3	91.1 ± 0.1	94.7 ± 0.3	89.4 ± 0.2	93.3 ± 0.1	$\mathbf{92.0 \pm 0.1}$	$\mathbf{97.0 \pm 0.0}$	91.9 ± 0.1	$\mathbf{97.0 \pm 0.0}$
	0.15	90.5 ± 0.1	93.8 ± 0.3	91.0 ± 0.2	94.5 ± 0.3	88.8 ± 0.2	92.4 ± 0.1	$\mathbf{92.0 \pm 0.1}$	$\mathbf{97.0 \pm 0.0}$	91.9 ± 0.2	$\mathbf{97.0 \pm 0.1}$
	0.20	90.5 ± 0.2	93.7 ± 0.2	90.8 ± 0.1	94.3 ± 0.2	87.8 ± 0.1	91.8 ± 0.1	$\mathbf{92.0 \pm 0.1}$	$\mathbf{96.9 \pm 0.0}$	91.9 ± 0.1	96.8 ± 0.1
	0.25	90.1 ± 0.2	93.4 ± 0.3	90.6 ± 0.1	94.0 ± 0.1	87.4 ± 0.1	91.4 ± 0.1	91.8 ± 0.1	96.8 ± 0.1	$\mathbf{91.9 \pm 0.1}$	$\mathbf{96.9 \pm 0.0}$

F TRANFERABILITY OF FAIRNESS ATTACKS BY FATE

For the evaluation results shown in Sections **??** and **??** as well as Appendices D and E, both the surrogate model (linear GCN) and the victim models (i.e., GCN, FairGNN, InFoRM-GNN) are convolutional aggregation-based graph neural networks. In this section, we aim to test the transferability of FATE by generating poisoned graphs on the convolutional aggregation-based surrogate model (i.e., linear GCN) and testing on graph attention network (GAT), which is a non-convolutional aggregation-based graph neural network (**?**).

More specifically, we train a graph attention network (GAT) with 8 attention heads for 400 epochs. The hidden dimension, learning rate, weight decay and dropout rate of GAT are set to 64, $1e-3$, $1e-5$ and 0.5, respectively.

The results on attacking statistical parity or individual fairness with GAT as the victim model are shown in Table 9. Even though the surrogate model used by the attacker is a convolutional aggregation-based linear GCN, from the table, it is clear that FATE can consistently succeed in (1) effective fairness attack by increasing Δ_{SP} and the individual bias (Bias) and (2) deceptive attack by offering comparable or even better micro F1 score (Micro F1) when the victim model is not a convolutional aggregation-based model. Thus, it shows that the adversarial edges flipped/added by FATE is able to transfer to graph neural networks with different type of aggregation function.

Table 9: Transferability of attacking statical parity and individual fairness with FATE on GAT under different perturbation rates (Ptb.). FATE poisons the graph via both edge flipping (FATE-flip) and edge addition (FATE-add). Higher is better (↑) for micro F1 score (Micro F1), Δ_{SP} (bias for statistical parity), and InFoRM bias (Bias, bias for individual fairness).

Attacking Statistical Parity

Dataset	Ptb.	Pokec-n		Pokec-z		Bail	
		Micro F1 (↑)	**Δ_{SP} (↑)**	**Micro F1 (↑)**	**Δ_{SP} (↑)**	**Micro F1 (↑)**	**Δ_{SP} (↑)**
FATE-flip	0.00	63.8 ± 5.3	4.0 ± 3.2	68.2 ± 0.5	8.6 ± 1.1	89.7 ± 4.2	7.5 ± 0.6
	0.05	63.9 ± 5.5	6.4 ± 5.1	68.3 ± 0.4	10.5 ± 1.3	90.1 ± 3.8	8.1 ± 0.6
	0.10	63.6 ± 5.3	7.9 ± 6.7	67.8 ± 0.4	11.2 ± 1.7	90.3 ± 3.2	8.5 ± 0.6
	0.15	63.7 ± 5.3	7.5 ± 6.1	68.2 ± 0.6	11.2 ± 1.5	90.2 ± 2.7	8.8 ± 0.3
	0.20	64.1 ± 5.6	7.7 ± 6.3	67.8 ± 0.6	11.1 ± 0.9	90.0 ± 2.7	8.7 ± 0.6
	0.25	63.6 ± 5.2	8.5 ± 7.0	68.0 ± 0.4	11.5 ± 1.2	89.9 ± 3.0	8.8 ± 0.5
FATE-add	0.00	63.8 ± 5.3	4.0 ± 3.2	68.2 ± 0.5	8.6 ± 1.1	89.7 ± 4.2	7.5 ± 0.6
	0.05	63.9 ± 5.5	6.4 ± 5.1	68.3 ± 0.4	10.5 ± 1.3	90.2 ± 3.7	8.1 ± 0.7
	0.10	63.6 ± 5.3	7.9 ± 6.7	67.8 ± 0.4	11.2 ± 1.7	90.3 ± 3.2	8.5 ± 0.6
	0.15	63.7 ± 5.3	7.5 ± 6.1	68.2 ± 0.6	11.2 ± 1.5	90.3 ± 2.6	8.8 ± 0.3
	0.20	64.1 ± 5.6	7.7 ± 6.3	67.8 ± 0.6	11.1 ± 0.9	90.1 ± 2.6	8.8 ± 0.5
	0.25	63.6 ± 5.2	8.5 ± 7.0	68.0 ± 0.4	11.5 ± 1.2	89.9 ± 2.9	8.8 ± 0.5

Attacking Individual Fairness

Dataset	Ptb.	Pokec-n		Pokec-z		Bail	
		Micro F1 (↑)	**Bias (↑)**	**Micro F1 (↑)**	**Bias (↑)**	**Micro F1 (↑)**	**Bias (↑)**
FATE-flip	0.00	63.8 ± 5.3	0.4 ± 0.2	68.2 ± 0.5	0.5 ± 0.1	89.7 ± 4.2	2.5 ± 1.2
	0.05	63.6 ± 5.3	0.5 ± 0.2	68.2 ± 0.8	0.6 ± 0.1	90.0 ± 4.2	2.7 ± 1.1
	0.10	63.7 ± 5.3	0.5 ± 0.2	67.8 ± 0.5	0.6 ± 0.1	90.0 ± 4.0	2.8 ± 1.3
	0.15	63.7 ± 5.4	0.5 ± 0.2	68.2 ± 0.5	0.6 ± 0.2	90.2 ± 3.6	2.8 ± 1.4
	0.20	63.5 ± 5.1	0.5 ± 0.2	68.5 ± 0.5	0.6 ± 0.2	90.2 ± 3.4	2.8 ± 1.2
	0.25	63.5 ± 5.1	0.5 ± 0.2	68.0 ± 0.6	0.6 ± 0.1	90.2 ± 3.1	2.7 ± 1.2
FATE-add	0.00	63.8 ± 5.3	0.4 ± 0.2	68.2 ± 0.5	0.5 ± 0.1	89.7 ± 4.2	2.5 ± 1.2
	0.05	63.9 ± 5.4	0.5 ± 0.2	68.2 ± 0.7	0.6 ± 0.1	90.0 ± 4.6	2.7 ± 1.4
	0.10	63.8 ± 5.4	0.5 ± 0.2	68.2 ± 0.5	0.6 ± 0.2	90.1 ± 4.0	2.8 ± 1.2
	0.15	63.8 ± 5.4	0.5 ± 0.2	68.3 ± 0.2	0.6 ± 0.2	90.1 ± 3.9	2.8 ± 1.2
	0.20	63.7 ± 5.3	0.5 ± 0.2	68.4 ± 0.3	0.6 ± 0.1	90.3 ± 3.2	2.8 ± 1.3
	0.25	63.7 ± 5.3	0.5 ± 0.2	68.4 ± 0.3	0.6 ± 0.1	90.2 ± 3.1	2.8 ± 1.2

G FURTHER DISCUSSIONS ABOUT FATE

A – Relationship between fairness attacks and the impossibility theorem of fairness. The impossibility theorems show that some fairness definitions may not be satisfied at the same time.[3] However, this may not always be regarded as fairness attacks. To our best knowledge, the impossibility theorems prove that two fairness definitions (e.g., statistical parity and predictive parity) cannot be fully satisfied at the same time, i.e., biases for two fairness definitions are both zero). However, there is no formal theoretical guarantees that ensuring one fairness definition will *always* amplify the bias of another fairness definition. Such formal guarantees might be nontrivial and beyond the scope of our paper. As we pointed out in the abstract, the main goal of this paper is to provide insights into the adversarial robustness of fair graph learning and can shed light for designing robust and fair graph learning in future studies.

B – Relationship between FATE and Metattack. FATE bears subtle differences with Metattack (**?**), which utilizes meta learning for adversarial attacks on utility. Note that Metattack aims to degrade the utility of a graph neural network by maximizing the task-specific utility loss (e.g., cross entropy for node classification) in the upper-level optimization problem. Different from Metattack, FATE aims to attack the fairness instead of utility by setting the upper-level optimization problem as maximizing a bias function rather than a task-specific utility loss.

C – Alternative edge selection strategy via sampling. Here we introduce an alternative perturbation set selection strategy that is different from the greedy selection described in Section **??**. The key idea is to view each edge in the graph as a Bernoulli random variable (**??**). And the general workflow is as follows. First, we follow Eq. **??** to get a poisoning preference matrix $\nabla_{\mathbf{A}}$. Then, we normalize $\nabla_{\mathbf{A}}$ to a probability matrix $\mathbf{P}_{\mathbf{A}}$. Finally, for the i-th attacking step, we can sample δ_i entries without replacement using $\mathbf{P}_{\mathbf{A}}$ as the set of edges to be manipulated (i.e., added/deleted/flipped).

D – The potential of FATE on attacking a specific demographic group in group fairness. To attack a specific group, there can be two possible strategies: (1) decreasing the acceptance rate of the corresponding group and (2) increasing the gap between the group to be attacked and another demographic group. For (1), following our strategy of modeling acceptance rate as the CDF of Gaussian KDE, we can set the bias function to be maximize as the negative of acceptance rate, i.e., $b(\mathbf{Y}, \Theta^*, \mathbf{F}) = -\mathrm{P}[\tilde{y} = 1 \mid s = a]$, where a is the sensitive attribute value denoting the demographic group to be attacked. For (2), suppose we want to attack the group with sensitive attribute value. We can also attack this demographic group by setting the bias function to be $b(\mathbf{Y}, \Theta^*, \mathbf{F}) = \mathrm{P}[\tilde{y} = 1 \mid s = 1] - \mathrm{P}[\tilde{y} = 1 \mid s = 0]$. In this way, we can increase the acceptance rate of demographic group ($s = 1$) while minimizing the acceptance rate of the group ($s = 0$).

E – The potential of FATE on attacking the best/worst accuracy group. To attack the best/worst accuracy group, the general idea is to set the bias function to be the loss of the best/worst group. It is worth noting that such attack is conceptually similar to adversarial attacks on the utility as shown in Metattack (**?**), but only focusing on a subgroup of nodes determined by the sensitive attribute rather than the validation set.

F – Justification of applying kernel density estimation on non-IID graph data. To date, it remains an open problem whether the learned node representations follow IID assumption on the low-dimensional manifold or not. Empirically from the experimental results, using KDE-based bias approximation effectively helps maximize the bias for fairness attacks. Meanwhile, relaxing the IID assumption is a common strategy in computing the distributional discrepancy of node representations. For example, MMD is a widely used distributional discrepancy measures, whose accurate approximation also requires IID assumption (**?**), and recent studies (**??**) show that we can also adapt it on non-IID data which shows promising empirical performance.

G – Possible defense strategies against deceptive fairness attacks. FATE demonstrate that it is possible to achieve deceptive fairness attacks on graph learning models by deliberately perturbing the input graph. Given its potential negative societal impacts, we discuss few possible defense strategies against such deceptive fairness attacks. To defend against deceptive fairness attacks for statistical parity, one possible strategy is to preprocess the input graph by either learning a bias-free graph (e.g., **?**) or sampling over the neighborhood (e.g., **???**) to control which node representations to aggregate during message passing. The reason for such possible design is that Figure **??** reveals the

[3] `https://machinesgonewrong.com/fairness/`

properties of injected edges that are likely to be incident to nodes in the minority class and/or protected group. Following similar principles, it is also possible to develop a selective or probabilistic message passing strategy to achieve the same goal during model optimization. To defend against deceptive fairness attacks for individual fairness, we can apply similar neighborhood sampling strategy or selective/probabilistic message passing strategy. Instead, for individual fairness, the neighborhood sampling or selective message passing would consider the class label rather than the sensitive attribute (i.e., sample edges that connect nodes in the minority class as shown in Figure **??**).

H – How does FATE maintain the performance for deceptiveness? We assume there is a divergence in optimizing the task-specific loss function $l\left(\mathcal{G}, \mathbf{Y}, \Theta, \theta\right)$ and optimizing the bias function $b\left(\mathbf{Y}, \Theta^{*}, \mathbf{F}\right)$. Thus, maximizing $b\left(\mathbf{Y}, \Theta^{*}, \mathbf{F}\right)$ may not affect $l\left(\mathcal{G}, \mathbf{Y}, \Theta, \theta\right)$ too much. Since we are minimizing the task-specific loss function in the inner loop (i.e., lower-level optimization), it helps to maintain the performance in the downstream task for deceptive fairness attacks. We think such assumption is reasonable for the following reason. In fair machine learning, a common strategy is to solve a regularized optimization problem, where the objective function to be minimized is often defined as $l\left(\mathcal{G}, \mathbf{Y}, \Theta, \theta\right) + \alpha b\left(\mathbf{Y}, \Theta^{*}, \mathbf{F}\right)$ with α being the regularization hyperparameter. If there is no divergence between the optimization of $l\left(\mathcal{G}, \mathbf{Y}, \Theta, \theta\right)$ and $b\left(\mathbf{Y}, \Theta^{*}, \mathbf{F}\right)$, it would be sufficient to optimize one of them to obtain fair and high-utility learning results, or it would be impossible to achieve a good trade-off between fairness and utility if they are completely conflicting with each other. All in all, we believe that optimizing the task-specific loss function $l\left(\mathcal{G}, \mathbf{Y}, \Theta, \theta\right)$ in the lower-level optimization problem could help maintain deceptiveness both intuitively and empirically as shown in Section **??**, Section **??**, Appendix D, and Appendix E.

H More Details on Fairness Definitions

We discuss more details about statistical parity and individual fairness here.

A – Statistical parity. Mathematically, statistical parity is equivalent to the statistical independence between the learning results (e.g., predicted labels of a classification algorithm) and the sensitive attribute. Consider a classification problem with $\tilde{y}$ being the predicted label, s being the sensitive attribute whose attribute value is in the set $\mathcal{S}$. Statistical parity is defined as follows.

$$\mathrm{P}\left[\tilde{y}=1\right] = \mathrm{P}\left[\tilde{y}=1 | s=a\right], \quad \forall a \in \mathcal{S} \tag{8}$$

B – Individual fairness. Other than group fairness, individual fairness studies fairness in a finer-grained individual level. It asks for similar individuals to be treated similarly **?**. Such principle is often formulated as a Lipschitz inequality

$$d_1\left(\mathbf{Y}\left[i,:\right], \mathbf{Y}\left[j,:\right]\right) \leq \epsilon d_2\left(i,j\right) \tag{9}$$

where $\mathbf{Y}$ is the learning results, ϵ is the Lipschitz constant, the left hand side $d_1\left(\mathbf{Y}\left[i,:\right], \mathbf{Y}\left[j,:\right]\right)$ measures the distance between the learning results $\mathbf{Y}\left[i,:\right]$ and $\mathbf{Y}\left[j,:\right]$ of data points i and j, respectively, and $d_2\left(i,j\right)$ measures the distance between the two data points. Given a graph $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$ with adjacency matrix $\mathbf{A}$ and node feature matrix $\mathbf{X}$, Kang et al. **?** further define d_1 as the squared Frobenius distance and assume the existence of an oracle pairwise node similarity matrix $\mathbf{S}$. Then, the overall individual bias of $\mathcal{G}$ is further defined as $\mathrm{Tr}\left(\mathbf{Y}^T\mathbf{L_S}\mathbf{Y}\right)$ where $\mathbf{L_S}$ is the graph Laplacian of similarity matrix $\mathbf{S}$.