Learning Fair Graph Neural Networks with Limited and Private Sensitive Attribute Information

Enyan Dai, Suhang Wang

Abstract—Graph neural networks (GNNs) have shown great power in modeling graph structured data. However, similar to other machine learning models, GNNs may make biased predictions w.r.t protected sensitive attributes, e.g., skin color and gender. This is because machine learning algorithms including GNNs are trained to reflect the distribution of the training data which often contains historical bias towards sensitive attributes. In addition, we empirically show that the discrimination in GNNs can be magnified by graph structures and the message-passing mechanism of GNNs. As a result, the applications of GNNs in high-stake domains such as crime rate prediction would be largely limited. Though extensive studies of fair classification have been conducted on independently and identically distributed (i.i.d) data, methods to address the problem of discrimination on non-i.i.d data are rather limited. Generally, learning fair models require abundant sensitive attributes to regularize the model. However, for many graphs such as social networks, users are reluctant to share sensitive attributes. Thus, only limited sensitive attributes are available for fair GNN training in practice. Moreover, directly collecting and applying the sensitive attributes in fair model training may cause privacy issues, because the sensitive information can be leaked in data breach or attacks on the trained model. Therefore, we study a novel and important problem of learning fair GNNs with limited and private sensitive attribute information. In an attempt to address these problems, FairGNN is proposed to eliminate the bias of GNNs whilst maintaining high node classification accuracy by leveraging graph structures and limited sensitive information. We further extend FairGNN to NT-FairGNN which can achieve both fairness and privacy on sensitive attributes by using limited and private sensitive attributes. Theoretical analysis and extensive experiments on real-world datasets demonstrate the effectiveness of FairGNN and NT-FairGNN in achieving fair and high-accurate classification.

Index Terms—Fairness; Graph Neural Networks; Privacy-Persevering

1 Introduction

Graph neural networks (GNNs) [1], [2], [3] have achieved remarkable performance on various domains such as knowledge graph [4], social media mining [5], and recommendation system [6], [7]. Generally, message-passing process is adopted in GNNs [2], [5], where information from neighbors is aggregated for every node in each layer. This process enriches node representations, and preserves both node feature characteristics and topological structures.

Despite the success in modeling graph data, GNNs trained on graphs may inherit the societal bias in data, which limits the adoption of GNNs in many real-world applications. *First*, extensive studies [8], [9], [10] have revealed that historical data may include patterns of previous discrimination and societal bias. Machine learning models trained on such data can inherit the bias on sensitive attributes such as ages, genders, skin color, and regions [8], [9], which implies that GNNs could also exhibit the bias. *Second*, the topology of graphs and the message-passing of GNNs could magnify the bias. Generally, in graphs such as social networks, nodes of similar sensitive attributes are more likely to connect to each other than nodes of different sensitive attributes [11], [12]. For example, young people tend to build friendship with people of similar age on the social network [11]. This makes the aggregation of neighbors' features in GNN have similar representations for nodes of similar sensitive information while different representations for nodes of different sensitive features, leading to severe bias in decision making, i.e., the predictions are highly correlated with the sensitive attributes of the nodes. Our preliminary experiments in Sec. 3.5 indicate that GNNs have a larger bias due to the adoption of graph structure than models that only use node attributes, which verifies our intuition. The bias would largely limit the wide adoption of GNNs in domains such as ranking of job applicants [13] and crime rate prediction [14]. Thus, it is important to investigate fair graph neural networks.

However, developing fair GNNs is a non-trivial task. *First*, to achieve fairness, we need to obtain abundant nodes with known sensitive attributes so that we can either revise the data or regularize the model; whereas people are unwilling to share their sensitive information in the real-world, and resulting in inadequate nodes with sensitive attributes known for fair model learning. For example, only 14% teen users public their complete profiles on Facebook [15]. The lacking of sensitive information challenges many existing work on fair models [9], [10], [16], [17]. *Second*, directly collecting the users' sensitive attributes and applying them for fair machine learning models may lead to a privacy issue. The collected sensitive attributes have a risk of data breach during the storage. For instance, the personal data of more than half a billion Facebook users was leaked online for free in a hacker forum in 2021[1]. The user sensitive attributes can also be leaked from the trained models which utilize the ground-truth sensitive attributes, because various attack methods [18], [19] can infer the training dataset infor-

• *Enyan Dai and Suhang Wang are with the College of Information Sciences and Technology at The Pennsylvania State University, University Park, PA, 16802, USA.*
E-mail: {emd5795,szw494}@psu.edu

1. https://www.bbc.com/news/technology-56745734

mation from the trained models. One of the most promising solutions is to obtain *private sensitive attributes* with Differential Privacy (DP) [20] which gives strong privacy guarantees. Though some efforts [21], [22] have been conducted for fairness on private demographic data, works that consider limited and private sensitive attributes are rather limited. *Third*, though extensive efforts have been made to establish fair models such as by revising features [23], disentanglement [10], adversarial debiasing [9] and fairness constraints [24], [25], they are overwhelmingly dedicated to independently and identically distributed (i.i.d) data, which cannot be directly applied on graph data for the absence of simultaneous consideration of the bias from node attributes and graph structures.

Therefore, in this paper, we study a novel problem of learning fair graph neural networks with limited private sensitive information. In essence, we need to solve three challenges: (i) how to overcome the shortage of sensitive attributes for eliminating discrimination; (ii) how to protect the privacy of the users on sensitive attributes; and (iii) how to ensure the fairness of the GNN classifier. In an attempt to address these challenges, we propose a novel framework named as **FairGNN** for fair node classification with limited sensitive attributes. We further extend the FairGNN to achieve fairness with limited private sensitive attributes to protect the privacy of users, obtaining a framework named **NT-FairGNN** . To alleviate the shortage of sensitive attributes, FairGNN and NT-FairGNN adopt a GNN sensitive attribute estimator to predict sensitive attributes for fair classification. Inspired by existing works of fair classification on i.i.d data with adversarial learning [9], [26], [27], we deploy an adversary to ensure the GNN classifier make predictions independent with the estimated sensitive attributes. To further stabilize the training process and performance in fairness, we introduce a fairness constraint to make the predictions invariant with the estimated sensitive attributes. Since the sensitive attribute estimator is trained on limited and private sensitive attributes, the predicted sensitive attributes are noisy. We theoretically show that our framework can learn fair classifier with the noisy estimated sensitive attributes. Our main contributions are:

- We study novel problems of fair graph neutral networks learning with limited/private sensitive information;
- We propose a new framework, FairGNN, which can learn fair and high accurate classifier on graphs with limited sensitive attributes. We further extend FairGNN to give fair prediction whilst protecting the privacy of user sensitive attributes; and
- We theoretically show that the proposed framework can achieve fairness with estimated noisy sensitive attributes;
- Extensive experiments on different datasets demonstrate the effectiveness of our methods in eliminating discrimination while keeping high accuracy of GNNs.

2 Related Work

In this section, we review related work including graph neural networks, fairness in machine learning and differential privacy in deep learning.

2.1 Graph Neural Networks

Graph neural networks (GNNs), which generalize neural networks for graph structured data, have shown great success for various applications [4], [5], [6]. Generally, GNNs can be categorized into two categories, i.e., spectral-based [1], [2], [28], [29] and spatial-based [3], [5]. Spectral-based GNNs define graph convolution based on spectral graph theory, which is first explored by Bruna et al. [1]. Since then, more spectral-based methods are developed for further improvements and extensions [2], [28], [29]. Graph Convolutional Network (GCN) [2] is a particularly popular method which simplifies the graph convolution by first-order approximation. For spatial-based graph convolution, it directly updates the node representation by aggregating the representations of its neighbors [5], [30]. For instance, spatial graph convolution that incorporates the attention mechanism is applied in GAT [3] to facilitate the information aggregation. GraphSAGE [5] adopts a neighbor sampling method in the training of GNN to solve the scalability issue of GCN [5]. Graph Isomorphism Network (GIN) [31] is proposed to learn more powerful representations of the graph structures. Recently, to incorporate long range neighbors in the message-passing of GNNs, deep graph neural networks [32], [33], [34], [35] has been investigated. For example, initial residual and identity mapping are used in [32] to extend the GCN to overcome the over-smoothing issue of deep GNNs. In [35], DropEdge is applied as data augmenter and a message passing reducer to reduces the convergence speed of over-smoothing.

The essential idea of GNNs is to propagate the information of nodes through the graph to get better representations. However, people tend to build relationships with those sharing the same sensitive attributes. Then, representations in GNNs are nearly propagated within the subgroup, which highly increases the risk of discrimination towards sensitive attributes. Despite the risk of discrimination in GNNs, the work on addressing this issue is rather limited. Therefore, we study the novel problem of learning fair GNNs to eliminate the potential discrimination.

2.2 Fairness in Machine Learning

Many works have been conducted to deal with the bias in the training data to achieve fairness in machine learning [8], [9], [16], [23], [36]. Based on which stage of the machine learning training process is revised, algorithms could be split into three categories: the pre-processing approaches, the in-processing approaches, and the post-processing approaches. The pre-processing approaches are applied before training machine learning models. They could reduce the bias by modifying the training data through correcting labels [23], revising attributes of data [37], generating non-discriminatory labeled data [38], [39], [40], and obtaining fair data representations [9], [10], [16], [41]. The in-processing approaches are designed to revise the training of the state-of-the-art models. Typically the machine learning models are trained with additional regularization terms or a new objective function [8], [24], [26], [42]. Finally, the post-processing approaches directly change the predictive labels to ensure fairness [36], [43]. The majority of the aforementioned approaches are for i.i.d data while fair models on

graphs are rather limited. Recently, several works explore the learning of fair graph embeddings for recommendation [12], [44]. Fairwalk [12] modifies the random walk procedure of node2vec [45] to obtain a more diverse network neighborhood representations. The sensitive attributes of all the nodes are required in the sampling procedure of FairWalk. Bose et al. [44] propose to add discriminators to eliminate the sensitive information in the graph embeddings. Similar to Fairwalk, the training process of discriminators also needs the sensitive attributes of all the nodes.

Our work is inherently different from existing works: (i) we focus on learning fair GNNs for node classification instead of fair graph embeddings; (ii) we address the problem that only a limited number of nodes are provided with privacy-preserving sensitive attributes in practice.

2.3 Differential Privacy in Deep Learning

Differential Privacy (DP) [20], [46] is a popular approach to provide privacy guarantee of training data. The key idea of DP is to add noise to datasets or model training to prevent the individual information leakage. Based on the assumption about the curator, DP can be categorized into centralized DP [46] and local DP [20]. Centralized DP assumes a trusted curator is available to apply calibrated noise to produce DP; while local DP assumes the curator is untrusted and perturbs users' data locally. Recently, various differential-privacy preserving deep learning methods [47], [48], [49], [50], [51] are investigated to protect the training data privacy. For instance, NoisySGD [48] add noises to the gradients during model training so the trained model parameters will not leak training data with certain guarantee. To preserve the privacy of users sensitive information, DP is also employed in fair machine learning models [21], [22], [52], [53] which requires sensitive information of users for discrimination elimination. For example, in [22], sensitive attributes are perturbed to obtain private sensitive attributes to meet local DP before a non-discriminatory learner is adapted to work with privatized protected attributes [22].

However, the aforementioned works are dedicated to i.i.d data, which may not be directly applicable to fair graph neural networks. Moreover, current fair models mostly requires plenty of private sensitive attributes, which is not realistic for real-world graphs. Therefore, we propose a novel framework NT-FariGNN to achieve fairness in GNN while maintaining the privacy of users' sensitive information.

3 Preliminaries Analysis

In this section, we conduct preliminary analysis on real-world datasets to show that GNNs could exhibit more severe bias due to the graph structure and message-passing.

3.1 Notations

We use $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ to denote an attributed graph, where $\mathcal{V} = \{v_1, ..., v_N\}$ is the set of N nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ is the set of node features. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix of the graph $\mathcal{G}$, where $\mathbf{A}_{ij} = 1$ if nodes v_i and v_j are connected; otherwise, $\mathbf{A}_{ij} = 0$. In the semi-supervised setting, part of nodes $v \in \mathcal{V}_L$ are provided with labels $y_v \in \mathcal{Y}$, where $\mathcal{V}_L \subseteq \mathcal{V}$ denotes nodes with labels, and $\mathcal{Y}$ is the set of labels. Sensitive attributes of training nodes are required to achieve fairness of machine learning algorithms. In our setting, only a small set of nodes $\mathcal{V}_S \subset \mathcal{V}$ are provided with the sensitive attribute $s \in \{0, 1\}$. The set of provided sensitive attributes is denoted by $\mathcal{S}$. Note that $\mathcal{V}_L$ and $\mathcal{V}_S$ can be totally different.

TABLE 1: The statistics of datasets.

Dataset	Pokec-z	Pokec-n	NBA
# of nodes	67,797	66,569	403
# of node attributes	59	59	39
# of edges	882,765	729,129	16,570
Size of $\mathcal{V}_L$	500	500	100
Size of $\mathcal{V}_S$	200	200	50
# of inter-group edges	39,804	31,515	4,401
# of intra-group edges	842,961	697,614	12,169

3.2 Datasets

For the purpose of this study, we collect and sample datasets from Pokec and NBA. The details are described as below.

Pokec [54]: Pokec is the most popular social network in Slovakia, which is very similar to Facebook and Twitter. This dataset contains anonymized data of the whole social network in 2012. User profiles of Pokec contain gender, age, hobbies, interest, education, working field and etc. The original Pokec dataset contains millions of users. Based on the provinces that users belong to, we sampled two datasets named as: **Pokec-z** and **Pokec-n**. Both Pokec-z and Pokec-n consist of users belonging to two major regions of the corresponding provinces. We treat the region as the sensitive attribute. The classification task is to predict the working field of the users.

NBA: This is extended from a Kaggle dataset [2] containing around 400 NBA basketball players. The performance statistics of players in the 2016-2017 season and other various information e.g., nationality, age, and salary are provided. To obtain the graph that links the NBA players together, we collect the relationships of the NBA basketball players on Twitter with its official crawling API [3]. We binarize the nationality to two categories, i.e., U.S. players and oversea players, which is used as sensitive attribute. The classification task is to predict whether the salary of the player is over median.

For all the datasets, we eliminate nodes without any links with others. We randomly sample labels and sensitive attributes separately to get $\mathcal{V}_L$ and $\mathcal{V}_S$. We randomly sample 25% and 50% of nodes containing both sensitive attributes and labels in Pokec-z, Pokec-n and NBA as validation sets and test sets. Note that the validation sets and test sets have no overlap with $\mathcal{V}_L$ and $\mathcal{V}_S$. The key statistics of the datasets are given in Table 1. Apart from the basic statistics, we also report the ratio of the majority and minority group and the number of edges linking the same group and different groups. It is evident from the table that: (i) skew exists in sensitive attributes; (ii) most of relationships are between users who share the same sensitive attribute.

2. https://www.kaggle.com/noahgift/social-power-nba
3. https://developer.twitter.com/en

TABLE 2: Results of models w/ and w/o utilizing graph.

Dataset	Metrics	MLP	MLP-e	GCN	GAT
Pokec-z	ACC (%)	65.3 ±0.5	68.6 ±0.3	70.2 ±0.1	70.4 ±0.1
	AUC (%)	71.3 ±0.3	74.8 ±0.3	77.2 ±0.1	76.7 ±0.1
	Δ_{SP} (%)	3.8 ±1.3	6.9 ±1.0	9.9 ±1.1	9.1 ±0.9
	Δ_{EO} (%)	2.2 ±0.7	4.0 ±1.5	9.1 ±0.6	8.4 ±0.6
Pokec-n	ACC (%)	63.1 ±0.4	66.3 ±0.6	70.5 ±0.2	70.3 ±0.1
	AUC (%)	68.2 ±0.3	72.4 ±0.6	75.1 ±0.2	75.1 ±0.2
	Δ_{SP} (%)	3.3 ±0.6	8.7 ±1.0	9.6 ±0.9	9.4 ±0.7
	Δ_{EO} (%)	7.1 ±0.9	9.9 ±0.6	12.8 ±1.3	12.0 ±1.5
NBA	ACC (%)	63.6 ±0.9	66.1 ±1.1	71.2 ±0.5	71.9 ±1.1
	AUC (%)	73.5 ±0.3	74.4 ±1.2	78.3 ±0.3	78.2 ±0.6
	Δ_{SP} (%)	6.0 ±1.5	10.9 ±1.9	7.9 ±1.3	10.2 ±2.5
	Δ_{EO} (%)	6.1 ±1.8	8.8 ±3.0	17.8 ±2.6	15.9 ±4.0

3.3 Preliminaries of Graph Neural Networks

Graph neural networks utilize the node attributes and edges to learn a representation $\mathbf{h}_v$ of the node $v \in \mathcal{V}$. The goal of learning representation in node classification is to predict the node v's label as $y_v = f(\mathbf{h}_v)$. Generally, GNNs adopt neighborhood aggregation approaches, which update the representations of a node v with the representations of v's neighborhood nodes. The representations of v after k layers' aggregation could capture the structural information of the k-hop subgraph centered at v. The updating process of the k-th layer in GNN could be formulated as:

$$\begin{aligned}\mathbf{a}_v^{(k)} &= \text{AGGREGATE}^{(k-1)}(\{\mathbf{h}_u^{(k-1)} : u \in \mathcal{N}(v)\}),\\ \mathbf{h}_v^{(k)} &= \text{COMBINE}^{(k)}(\mathbf{h}_v^{(k-1)}, \mathbf{a}^{(k)}),\end{aligned} \tag{1}$$

where $\mathbf{h}_v^{(k)}$ is the representation vector of the node $v \in \mathcal{V}$ at k-th layer and $\mathcal{N}(v)$ is the set of neighbors of v.

3.4 Fairness Evaluation Metrics

In this subsection, we present two definitions of fairness for the binary label $y \in \{0, 1\}$ and the sensitive attribute $s \in \{0, 1\}$. $\hat{y} \in \{0, 1\}$ denotes the prediction of the classifier η: $\mathbf{x} \to y$.

Definition 1. *(Statistical Parity [8]). Statistical parity requires the predictions to be independent with the sensitive attribute s, i.e., $\hat{y} \perp s$. It could be formally written as:*

$$P(\hat{y}|s = 0) = P(\hat{y}|s = 1). \tag{2}$$

Definition 2. *(Equal Opportunity [36]). Equal opportunity requires the probability of an instance in a positive class being assigned to a positive outcome should be equal for both subgroup members. The property of equal opportunity is defined as:*

$$P(\hat{y} = 1|y = 1, s = 0) = P(\hat{y} = 1|y = 1, s = 1). \tag{3}$$

The equal opportunity expects the classifier to give equal true positive rates across the subgroups.

Following [9], [17], we adopt the following metrics to measure statistical parity and equal opportunity:

$$\Delta_{SP} = |P(\hat{y} = 1|s = 0) - P(\hat{y} = 1|s = 1)|, \tag{4}$$

$$\Delta_{EO} = |P(\hat{y} = 1|y = 1, s = 0) - P(\hat{y} = 1|y = 1, s = 1)|, \tag{5}$$

where the probabilities are evaluated on the test set.

3.5 Discrimination in Graph Neural Networks

Various machine learning algorithms such as logistic regression [24], SVM [24], and MLP [41] have been reported to have discrimination. The features of the instances may contain proxy variables of the sensitive attribute. It could result in biased predictions. For GNNs, edges in graph can bring linking bias, i.e., the misrepresentation due to the connections of users [13]. It has been proven that the embeddings of nodes within the connected component will be closer after one aggregation in GCN [55], [56]. Since most of edges are intra-group as Table 1 shows, embeddings of nodes sharing the same sensitive attribute will be closer after k-layer information aggregation. As a result, representations of the nodes may exhibit bias. Intuitively, similar discrimination also exists in other GNNs that aggregate information of neighborhoods.

To empirically demonstrate the existence of discrimination in GNNs, we make comparisons between the following models:

- **MLP**: A multi-layer perception model trained on $\mathcal{V}_L$.
- **MLP-e**: A MLP model utilizes graph structure by adding embeddings learned by deepwalk to the features.
- **GCN** [2]: A representative spectral graph neural network.
- **GAT** [3]: A spatial GNN which utilizes attention to assign higher weights to more important edges.

For each model, we run the experiment 5 times. The classification results and discrimination scores on the test set are reported in Table 2. From the table, we observe that (i) both performance of GCN and GAT are much better than MLP, which is as expected because GCN and GAT adopt both node attributes and the graph structure for classification; (ii) Compared with MLP, models utilizing graph structure, i.e., GCN and GAT, perform significantly worse in terms of fairness, which verifies that *bias exists in GNNs and the graph structure could further aggravate the discrimination.*

4 Problem Definition

Our preliminary analysis verifies that GNNs have severe bias issue. Thus, it is important to develop fair GNNs. In this section, we first present the formal problem definition of learning fair GNN with limited sensitive attributes. Next, we give the problem definition of training fair GNN with limited and private sensitive attributes to simultaneously give fair node predictions and protect the privacy of users' sensitive attributes.

4.1 Fair GNN with Limited Sensitive Attributes

Following existing work of fair models [9], [17], [37], [38], we focus on the binary class and binary sensitive attribute setting, i.e., both y and s can either be 0 or 1. We leave the extension to multi-class and multi-sensitive attribute setting as a future work. With the notations given in Section 3.1, the problem of learning fair GNN with limited sensitive attributes is formally defined as:

Problem 1. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, small labeled node set $\mathcal{V}_L \in \mathcal{V}$ with the corresponding labels in $\mathcal{Y}$, and a small set of nodes $\mathcal{V}_S \in \mathcal{V}$ with corresponding sensitive attributes in $\mathcal{S}$, learn a fair GNN for fair node classification, i.e.,*

$$f(\mathcal{G}, \mathcal{Y}, \mathcal{S}) \to \hat{\mathcal{Y}} \tag{6}$$

where f is the function we aim to learn and $\hat{\mathcal{Y}}$ is the set of predicted labels for unlabeled nodes. $\hat{\mathcal{Y}}$ should maintain high accuracy whilst satisfy fairness criteria such as statistical parity.

4.2 Privacy Protection in Fair GNN

As it is presented in Problem 1, a limited number of sensitive attributes will be utilized to achieve fairness in GNNs. However, if we directly apply the ground-truth sensitive attributes for debiasing, the trained model will have a risk of sensitive information leakage from the attacks. This privacy concern may result in the loss of users and even cause some legal issues. Thus, it is also crucial to protect the privacy of users on sensitive attributes in fair GNN. One of the most promising directions is to apply local differential privacy to obtain private sensitive attributes that provide strong privacy guarantees. Therefore, we first introduce the definitions of local differential privacy and private sensitive attributes in this subsection. We then formalize the problem of learning fair GNN with limited and private sensitive attributes to protect users' privacy.

Local differential privacy (LDP) [20] guarantees that the data aggregator does not know for certain the protected attributes of any data point by injecting noises before it leaves the user's device. To preserve users' privacy in fair GNN, we can obtain private sensitive attributes with local differential privacy. As a result, the fair GNN built on the private sensitive attributes is differentially private. Formally, the local differential privacy is defined as follows:

Definition 3 (ϵ-Local Differential Privacy [20])**.** *Given $\epsilon > 0$, a randomized mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy, if for all possible pairs of users' private data s_i and s_j, the following equation is met:*

$$\forall a \in Range(\mathcal{M}) : \frac{P(\mathcal{M}(s_i) = a)}{P(\mathcal{M}(s_j) = a)} \leq e^{\epsilon}. \tag{7}$$

where $Range(\mathcal{M})$ denotes every possible output of $\mathcal{M}$.

The parameter ϵ is the privacy budget to tune the utility and privacy of the model. A larger ϵ will lead to stronger privacy guarantee, but weaker utility. To preserve the privacy of the sensitive information, we can obtain the private sensitive attributes that follows local differential privacy according to the following Lemma:

Lemma 1. *To achieve ϵ-local differential privacy on the binary sensitive attribute, we can randomly flip the sensitive attributes with a probability of $\rho = \frac{1}{\exp(\epsilon)+1}$.*

For detailed proof, please refer to Lemma 3 in [21]. According to Lemma 1, the differentially private sensitive attributes can be obtained by injecting a certain level of flipping noises to the sensitive attributes. In the real-world applications, platforms can ask the users whether they are willing to share their sensitive attributes with privacy protection. If the answer is yes, the collection code will automatically flip the sensitive attributes with a probability of ρ in users' local devices and upload the private sensitive attributes to the platforms. If the answer is no, the sensitive attributes will not be collected. Note that many users could be not willing to share their sensitive attribute information, we may still collect a limited number of private sensitive attributes. This data collection process have been shown on the left part of Figure 2. With the definitions of LDP and private sensitive attributes, the problem of learning fair GNN with private sensitive attributes can be formulated as:

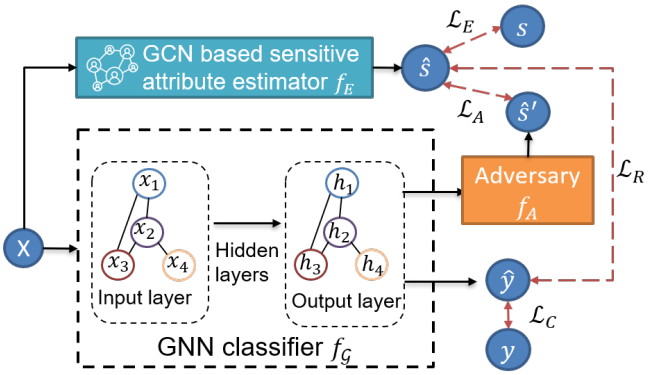


Fig. 1: The overall framework of FairGNN.

Problem 2. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, small labeled node set $\mathcal{V}_L \in \mathcal{V}$ with labels $\mathcal{Y}$, a part of nodes $\mathcal{V}_S \in \mathcal{V}$ with private sensitive attributes $\mathcal{S}_p$, and the randomlly flipping probability ρ in $\mathcal{S}_p$, learn a fair GNN for fair node classification, i.e.,*

$$f(\mathcal{G}, \mathcal{Y}, \mathcal{S}_p, \rho) \rightarrow \hat{\mathcal{Y}} \tag{8}$$

where f is the function we aim to learn and $\hat{\mathcal{Y}}$ is the set of predicted labels for unlabeled nodes. $\mathcal{S}_p$ meets ϵ-local differential privacy by randomly flipping their values with a probability of ρ. And $\hat{\mathcal{Y}}$ should maintain high accuracy whilst satisfying the fairness criteria such as statistical parity.

5 FairGNN for Limited Sensitive Attributes

In this section, we give the details of the proposed FairGNN for learning fair GNN with limited sensitive attributes (Problem 1). An illustration of the proposed framework is shown in Fig. 1, which is composed of a GNN classifier $f_{\mathcal{G}}$, a GCN based sensitive attribute estimator f_E and an adversary f_A. The classifier $f_{\mathcal{G}}$ takes $\mathcal{G}$ as input for node classification. The sensitive attribute estimator f_E is to predict the sensitive attributes for nodes whose sensitive attributes are unknown, which paves us a way to adopt adversarial learning to learn fair node representations and to regularize the predictions of $f_{\mathcal{G}}$. Specifically, the adversary f_A aims to predict the known or estimated sensitive attributes by f_E from the node representation learned by $f_{\mathcal{G}}$; while $f_{\mathcal{G}}$ aims to learn fair node representations that can fool the adversary f_A to make wrong predictions. We theoretically prove that under mild conditions, such minmax game can guarantee that learned representations are fair. In addition to make the representations fair, we directly add a regularizer on the predictions of $f_{\mathcal{G}}$ to guarantee that $f_{\mathcal{G}}$ gives fair predictions. Next, we introduce each component in detail along with the theoretical proof.

5.1 The GNN Classifier

The GNN classifier $f_{\mathcal{G}}$ takes $\mathcal{G}$ as input and predicts node labels. The proposed framework FairGNN is flexible. Any GNNs that follow the structure of Eq.(1) can be used such as

GCN [2] and GAT [3]. Let $f_{\mathcal{G}}^{(k)}$ denotes the operation of aggregating and combining the information of node v and its k-hop neighborhoods through k layers' iterations in GNN classifier $f_{\mathcal{G}}$. For a GNN with K layers, the representation of node v of the final layer could be written as:

$$\mathbf{h}_v = f_{\mathcal{G}}^{(K)}(\mathbf{x}_v, \mathcal{N}_v^{(K)}), \tag{9}$$

where $\mathcal{N}_v^{(K)}$ represents the K-hop neighborhoods of v. To get the $\hat{y}_v$, i.e., the prediction of node v, a linear classification layer is applied to $\mathbf{h}_v$ as:

$$\hat{y}_v = \sigma(\mathbf{h}_v \cdot \mathbf{w}), \tag{10}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weights of the linear classification layer and σ is the sigmoid function. The loss function for training $f_{\mathcal{G}}$ is given as

$$\min_{\theta_{\mathcal{G}}} \mathcal{L}_C = -\frac{1}{|\mathcal{V}_L|} \sum_{v \in \mathcal{V}_L} [y_v \log \hat{y}_v + (1 - y_v) \log (1 - \hat{y}_v)], \tag{11}$$

where $|\mathcal{V}_L|$ denotes the size of $\mathcal{V}_L$, $\theta_{f_{\mathcal{G}}}$ represents the parameters of $f_{\mathcal{G}}$ and y_v is the groundtruth label of node v.

5.2 Sensitive Attribute Estimation

The GNN classifier $f_{\mathcal{G}}$ can make biased predictions because the learned representations of $f_{\mathcal{G}}$ exhibit bias due to the node features, graph structure and aggregation mechanism of GNN. One way to make $f_{\mathcal{G}}$ fair is to eliminate the bias of the final layer representations $\mathbf{h}_v$. Recently, adversarial debiasing has been proven to be effective in alleviating the bias of representations [9], [27], [41], [57]. In the general process of adversarial debiasing, an adversary is used to predict sensitive attributes from the representations of the classifier; while the classifier is trained to learn representations to make the adversary unable to predict the sensitive attributes while keep high accuracy in the classification task. Such process requires *abundant data samples with known sensitive attributes* so that we can judge if the adversary can make accurate predictions or not.

However, in practice people are reluctant to share their sensitive attributes, which leads to a small size of $\mathcal{V}_S$. Lacking of data with labeled sensitive attributes would result in poor improvement in fairness even with adversarial debiasing. Though we have limited nodes with sensitive attributes, i.e., small $\mathcal{V}_S$, generally, nodes with similar sensitive attributes are more likely connected to each other, which makes it possible to accurately predict the sensitive attributes for nodes in $\mathcal{V} - \mathcal{V}_S$ using the graph $\mathcal{G}$ and $\mathcal{V}_S$. Thus, we deploy a graph convolutional network $f_E : \mathcal{G} \to \mathcal{S}$ to estimate the sensitive attribute of node whose sensitive attribute is unavailable. The large amount of estimated sensitive attributes would greatly benefit the adversarial debiasing. Note that it is important to use two separate GNNs for node label prediction and sensitive attribute prediction because we aim to learn fair representations $\mathbf{h}_v$ for $f_{\mathcal{G}}$, i.e., $\mathbf{h}_v$ does not contain the sensitive information. The objective function of training f_E is

$$\min_{\theta_E} \mathcal{L}_E = -\frac{1}{|\mathcal{V}_S|} \sum_{v \in \mathcal{V}_S} [s_v \log \hat{s}_v + (1 - s_v) \log (1 - \hat{s}_v)], \tag{12}$$

where $\hat{s}_v$ is the predicted sensitive attribute of node $v \in \mathcal{V}_S$ by f_E and θ_E is the set of parameters of f_E.

5.3 Adversarial Debiasing with Estimator f_E

With f_E, we could get the estimation of the sensitive attributes $\hat{\mathcal{S}}_u$ of the nodes $u \in (\mathcal{V} - \mathcal{V}_S)$. We use $\hat{\mathcal{S}}$ to denote the set of sensitive attributes by combining $\mathcal{S}$ and $\hat{\mathcal{S}}_u$, i.e., $\hat{\mathcal{S}} = \mathcal{S} \cup \hat{\mathcal{S}}_u$. During the training process, for each node $v \in \mathcal{V}$, the adversary f_A tries to predict v's sensitive attribute $\hat{s}_v$ given the representation $\mathbf{h}_v$ as $f_A(\mathbf{h}_v)$; while $f_{\mathcal{G}}$ aims to learn node representation $\mathbf{h}_v$ that makes the adversary f_A unable to distinguish which sensitive group the node v belong to. This min max game can be written as

$$\begin{aligned} \min_{\theta_{\mathcal{G}}} \max_{\theta_A} \mathcal{L}_A = {} & \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\hat{s}=1)}[\log(f_A(\mathbf{h}))] \\ & + \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\hat{s}=0)}[\log(1 - f_A(\mathbf{h}))], \end{aligned} \tag{13}$$

where $\mathbf{h} \sim p(\mathbf{h}|\hat{s} = 1)$ means sampling a node with sensitive attribute as 1 from $\mathcal{G}$. θ_A is the parameters of f_A.

Theoretical Analysis. Since the size of $\mathcal{V}_S$ is small, the estimation of sensitive attributes will introduce nonnegligible noise. The noise of the sensitive attributes may influence the adversarial debiasing. Thus, we conduct theoretical analysis to show that sensitive attributes containing noise could help to achieve statistical parity under mild conditions. Next, we give the details of the proof.

Proposition 1. *The global minimum of Eq.(13) is achieved if and only if $p(\mathbf{h}|\hat{s} = 1) = p(\mathbf{h}|\hat{s} = 0)$, where $\hat{s} \in \hat{\mathcal{S}}$ and $\mathbf{h}$ is the final layer representation learned by the K-layer GNN classifier $f_{\mathcal{G}}$.*

Proof. According to Proposition 1. in [58], the optimal adversary is $f_A^*(\mathbf{h}) = \frac{p(\mathbf{h}|\hat{s}=1)}{p(\mathbf{h}|\hat{s}=1)+p(\mathbf{h}|\hat{s}=0)}$. Then the min max game in Eq.(13) could be reformulated as minimizing this function:

$$\begin{aligned} C^s = {} & \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\hat{s}=1)} \big[\log \frac{p(\mathbf{h}|\hat{s}=1)}{p(\mathbf{h}|\hat{s}=1) + p(\mathbf{h}|\hat{s}=0)} \big] \\ & + \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\hat{s}=0)} \big[\log \frac{p(\mathbf{h}|\hat{s}=0)}{p(\mathbf{h}|\hat{s}=1) + p(\mathbf{h}|\hat{s}=0)} \big] \\ = {} & -\log(4) + 2 \cdot JSD(p(\mathbf{h}|\hat{s}=1) || p(\mathbf{h}|\hat{s}=0). \end{aligned} \tag{14}$$

The Jensen-Shannon divergence between two distributions is non-negative, and become zero if the two distributions are equal. Thus, only if $p(\mathbf{h}|\hat{s} = 1) = p(\mathbf{h}|\hat{s} = 0)$, the objective function C^s will reach the minimum, which completes our proof. □

Theorem 1. *Let $\hat{y}$ denote the prediction of $f_{\mathcal{G}}$. Suppose:*

1) *The estimated sensitive attribute $\hat{s}$ and $\mathbf{h}$ are independent conditioned on true sensitive attribute s, i.e., $p(\hat{s}, \mathbf{h}|s) = p(\hat{s}|s)p(\mathbf{h}|s)$;*
2) *$p(s = 1|\hat{s} = 1) \neq p(s = 1|\hat{s} = 0)$.*

If Eq.(13) reaches the global minimum, the GNN classifier $f_{\mathcal{G}}$ will achieve statistical parity, i.e., $p(\hat{y}|s = 0) = p(\hat{y}|s = 1)$.

Proof. Under the assumption that $p(\hat{s}, \mathbf{h}|s) = p(\hat{s}|s)p(\mathbf{h}|s)$, we could obtain $p(\mathbf{h}|s, \hat{s}) = p(\mathbf{h}|s)$. From Proposition 1, we have $p(\mathbf{h}|\hat{s} = 1) = p(\mathbf{h}|\hat{s} = 0)$ when the algorithm converges, which is equivalent to $\sum_s p(\mathbf{h}, s|\hat{s} = 1) = \sum_s p(\mathbf{h}, s|\hat{s} = 0)$. Together with $p(\mathbf{h}|s, \hat{s}) = p(\mathbf{h}|s)$, we arrive at

$$\sum_s p(\mathbf{h}|s)p(s|\hat{s} = 1) = \sum_s p(\mathbf{h}|s)p(s|\hat{s} = 0) \tag{15}$$

Reordering the terms in Eq.(15), we can get

$$\begin{aligned}\frac{p(\mathbf{h}|s=1)}{p(\mathbf{h}|s=0)} &= \frac{p(s=0|\hat{s}=1)-p(s=0|\hat{s}=0)}{p(s=1|\hat{s}=0)-p(s=1|\hat{s}=1)} \\ &= \frac{(1-p(s=1|\hat{s}=1))-(1-p(s=1|\hat{s}=0))}{p(s=1|\hat{s}=0)-p(s=1|\hat{s}=1)} \\ &= 1\end{aligned} \tag{16}$$

Eq.(16) shows that at the global minimum $p(\mathbf{h}|s=1) = p(\mathbf{h}|s=1)$ under the assumption $p(s=1|\hat{s}=1) \neq p(s=1|\hat{s}=0)$. Since $\hat{y} = \sigma(\mathbf{h}\cdot\mathbf{w})$, we could get $p(\hat{y}|s=1) = p(\hat{y}|s=0)$. Thus, the statistical parity is achieved when Eq.(13) converges. □

In our proof, two assumptions are made. For the first assumption, since we use f_E to predict the sensitive attributes $\hat{s}$ and $f_{\mathcal{G}}$ to get the latent representation $\mathbf{h}$, and f_E and $f_{\mathcal{G}}$ doesn't share any parameters, it is generally true that $\hat{s}$ is independent with the representation $\mathbf{h}$, i.e., $p(\hat{s},\mathbf{h}|s) = p(\hat{s}|s)p(\mathbf{h}|s)$. As for the second assumption, it will be satisfied when we have a reasonable estimator f_E, i.e., f_E doesn't give random predictions.

5.4 Covariance Constraint

The instability of the training process of adversarial learning is well known [59]. In adversarial debiasing, failure to coverage may result in a classifier with discrimination. To alleviate this issue, we add a covariance constraint [24], [25] on the output of $f_{\mathcal{G}}$ to help the model achieve fairness. The covariance constraint has been explored in [24], [25] by minimizing the absolute covariance between users' sensitive attributes and the signed distance from the users' features to the decision boundary for fair linear classifiers. In our problem, only a small portion of users' sensitive attributes are known and the decision boundary of GNN is hard to obtain. Thus, we propose to minimize the absolute covariance between the noisy sensitive attribute $\hat{s} \in \hat{\mathcal{S}}$ and prediction $\hat{y}$

$$\mathcal{L}_R = |\mathrm{Cov}(\hat{s},\hat{y})| = |\mathbb{E}[(\hat{s}-\mathbb{E}(\hat{s}))(\hat{y}-\mathbb{E}(\hat{y}))]|, \tag{17}$$

where $|\cdot|$ indicates the absolute value.

Theoretical Analysis. Since $\mathcal{L}_R$ is the absolute value of covariance between $\hat{y}$ and $\hat{s}$, $\mathcal{L}_R = 0$, i.e., the global minimum of $\mathcal{L}_R$, is the prerequisite that $\hat{y}$ and $\hat{s}$ are independent. Thus, we will show that $\mathcal{L}_R = 0$ is the prerequisite of the statistical parity under mild assumption with the following theorem.

Theorem 2. *Suppose that $p(\hat{s},\mathbf{h}|s) = p(\hat{s}|s)p(\mathbf{h}|s)$, when $f_{\mathcal{G}}$ satisfies statistical parity, i.e. $\hat{y}\perp s$, we have $\hat{y}$ is independent with $\hat{s}$ and $\mathcal{L}_R = 0$.*

Proof. Through $p(\hat{s},\mathbf{h}|s) = p(\hat{s}|s)p(\mathbf{h}|s)$, we could get $p(\mathbf{h}|s,\hat{s}) = p(\mathbf{h}|s)$. Then, $p(\hat{y}|s,\hat{s}) = p(\hat{y}|s)$ as $\hat{y} = \sigma(\mathbf{h}\cdot\mathbf{w})$. When $\hat{y}\perp s$, the distribution $p(\hat{y},\hat{s})$ would be:

$$p(\hat{y},\hat{s}) = \sum_s p(\hat{y}|s)p(\hat{s},s) = \sum_s p(\hat{y})p(\hat{s},s) = p(\hat{y})p(\hat{s}). \tag{18}$$

Thus, $\hat{y}$ is independent with $\hat{s}$ when the statistical parity is achieved. Then, we can get $\mathcal{L}_R = |\mathrm{Cov}(\hat{s},\hat{y})| = |\mathbb{E}(\hat{s},\hat{y}) - \mathbb{E}(\hat{s})\mathbb{E}(\hat{y})| = 0$. □

In the proof, we assume that $p(\hat{s},\mathbf{h}|s) = p(\hat{s}|s)p(\mathbf{h}|s)$, which is generally valid as discussed previously.

Algorithm 1 Training Algorithm of FairGNN.

Input: $\mathcal{G} = (\mathcal{V},\mathcal{E},\mathbf{X})$, $\mathcal{Y}$, $\mathcal{S}$, α and β.
Output: $f_{\mathcal{G}}$, f_A, and f_E

```
Initialize f_E by optimizing Eq.(12) w.r.t θ_E
repeat
    Obtain the estimated sensitive attributes with f_E
    Optimize the GNN classifier parameters θ_G, the adversary parameters θ_A, and the estimator parameters θ_E by Eq.(19).
until convergence
return f_G, f_A, and f_E
```

5.5 Final Objective Function of FairGNN

We now have $f_{\mathcal{G}}$ for label prediction, f_E for sensitive attribute estimation, f_A with adversarial debiasing to force the node representations learned by $f_{\mathcal{G}}$ are fair, and covariance constraint to further ensure that the prediction of $f_{\mathcal{G}}$ is fair. Combining all these together, the final objective function of FairGNN could be written as:

$$\min_{\theta_{\mathcal{G}},\theta_E}\max_{\theta_A} \mathcal{L}_C + \mathcal{L}_E + \alpha\mathcal{L}_R - \beta\mathcal{L}_A, \tag{19}$$

where $\theta_{\mathcal{G}}$, θ_E, and θ_A are the parameters of classifier, estimator, and adversary, respectively. α and β are scalars to control the contributions of the covariance constraint and adversarial debiasing, respectively.

5.6 Training Algorithm of FairGNN

The training algorithm of FairGNN is presented in Algorithm 1. Specifically, we first pretrain f_E to ensure it meets the second assumption in Theorem 1. Sequentially, we optimize the whole model with Eq.(19) through the ADAM optimizer [60]. In the training process, we replace the hard labels in $\mathcal{L}_A$ with soft labels, i.e., the probability produced by f_E, to stabilize the adversarial learning [61].

6 Proposed NT-FairGNN for Limited and Private Sensitive Attributes

In this section, we introduce the details of NT-FairGNN which can achieve both fairness and privacy by using limited and private sensitive attributes (Problem 2). The illustration of our proposed NT-FairGNN is shown in Fig. 2. During the data collection phase, local differential privacy mechanism, i.e., randomly flipping, is applied to the sensitive attributes on users' own devices. Platforms will only be able to access the private sensitive attributes $\mathcal{S}_P$ and the attributed graph $\mathcal{G}$ to train a fair model. However, the noise in private sensitive may largely affects the sensitive attribute estimation in FairGNN. And according to Theorem 1, a reasonable sensitive attribute estimator is required to help debias in GNN. Thus, to address the challenge of limited and private sensitive attributes, NT-FairGNN extends the FairGNN by adapting a corrected loss $\mathcal{L}_E^P$ to the GCN-based sensitive attribute estimator f_E. Similar to FairGNN, the estimated sensitive attributes are utilized in both adversarial debiasing and covariance constraint to obtain fair node classification. Next, we will first introduce how to learn a

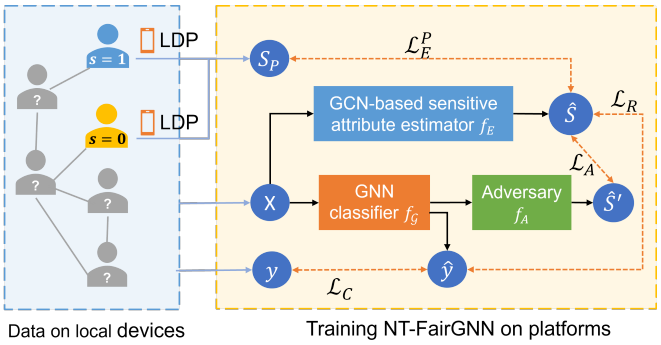


Fig. 2: The illustration of NT-FairGNN.

sensitive attribute estimator robust to the noise in the private sensitive attributes. Then, the overall objective function for discrimination elimination and the training algorithm of NT-FairGNN are presented.

6.1 Sensitive Attribute Estimator in NT-FairGNN

For NT-FairGNN, only private sensitive attributes $\mathcal{S}_P$ are available for training. As discussed in Lemma 1, private sensitive attributes are obtained by flipping the real sensitive attributes with a probability of ρ to preserve privacy. Directly optimizing the sensitive attribute estimator f_E with Eq.(12) is equivalent to learn an estimator for noisy private sensitive attributes $p(\tilde{s}|\mathbf{x})$, which will lead to poor sensitive attribute estimation. Following [62], we correct the loss function to reduce the effects of noise in provided labels of sensitive attributes. Since the sensitive attributes are randomly flipped in the LDP process, the obtained $\tilde{s}$ is only related with the original value s, i.e., $p(\tilde{s}|s,\mathbf{x}) = p(\tilde{s}|s)$. The cross entropy loss for a node v with private sensitive attribute $\tilde{s}$ can be rewritten as:

$$\begin{aligned} l(p(\tilde{s}|\mathbf{x})) &= -\log p(\tilde{s}|\mathbf{x}) \\ &= -\log \sum_j p(\tilde{s}|s=j)p(s=j|\mathbf{x}). \end{aligned} \tag{20}$$

From Eq. 20, we can find that if f_E aims to estimate $p(s|\mathbf{x})$, a prediction correction based on the $p(\tilde{s}|s)$ is required before the cross entropy loss. $p(\tilde{s}|s)$ can be represented by the noise transition matrix $\mathbf{T}$, where $T_{ij} = p(\tilde{s}=i|s=j)$. Since ρ, i.e., the probability of flipping sensitive attribute $s \in \{0,1\}$, is known for the platforms, the noise transition matrix $\mathbf{T}$ can be written as:

$$\mathbf{T} = \begin{bmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{bmatrix}. \tag{21}$$

With the correction based on Eq. 20 and the noise transition matrix $\mathbf{T}$, the objective function of optimizing the sensitive attribute estimator f_E with limited private sensitive attributes $\mathcal{S}_P$ can be formally stated as:

$$\min_{\theta_E} \mathcal{L}_E^P = \frac{1}{|\mathcal{S}_P|} \sum_{\tilde{s}_v \in \mathcal{S}_P} [-\log \sum_j T_{\tilde{s}_v j} f_E(\hat{s}=j|\mathbf{x})], \tag{22}$$

where θ_E denotes the learnable parameters of the sensitive attribute estimator f_E, and $f_E(\hat{s}=j|\mathbf{x})$ represents the predicted probability from f_E that the instance $\mathbf{x}$ belongs to the sensitive group i.

Theoretical Analysis. The following Theorem shows that optimizing the cross entropy loss between the corrected predictions and the noisy sensitive attributes is equivalent to optimizing the loss between the estimations and clean sensitive attribute distribution.

Algorithm 2 Training Algorithm of NT-FairGNN.

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, $\mathcal{Y}$, $\mathcal{S}_P$, α, β and ρ.
Output: $f_\mathcal{G}$, f_A, and f_E
1: Get noise transition matrix $\mathbf{T}$ with Eq.(21)
2: Initialize f_E by optimizing Eq.(22) w.r.t θ_E
3: **repeat**
4: Obtain the estimated sensitive attributes with f_E
5: Optimize the GNN classifier parameters $\theta_\mathcal{G}$, the adversary parameters θ_A, and the estimator parameters θ_E by Eq.(24).
6: **until** convergence
7: **return** $f_\mathcal{G}$, f_A, and f_E

Theorem 3. *Let $\tilde{s} \in \{0,1\}$ denotes the private sensitive attribute, and $f_E(\hat{s}=i|\mathbf{x})$ represents the predicted probability from f_E that the $\mathbf{x}$ belongs to sensitive group i. When $\mathbf{T}$ is non-singular,*

$$\begin{aligned} &\arg\min_{f_E} \mathbb{E}_{(\mathbf{x},\tilde{s})\sim p(\mathbf{x},\tilde{s})}[-\log \sum_i T_{\tilde{s}i} f_E(\hat{s}=i|\mathbf{x})] \\ &= \arg\min_{f_E} \mathbb{E}_{(\mathbf{x},s)\sim p(\mathbf{x},s)}[-\log f_E(\hat{s}=s|\mathbf{x})], \end{aligned} \tag{23}$$

where $p(\mathbf{x},\tilde{s})$ denotes the joint distribution of the input attribute $\mathbf{x}$ and private sensitive attribute $\tilde{s}$ and $p(\mathbf{x},s)$ denotes the joint distribution of the input attribute $\mathbf{x}$ and real sensitive attribute s.

Proof. For a detailed proof, please refer to Theorem 2 in [62]. We omit the details here. □

6.2 Overall Objective Function of NT-FairGNN

Similar to FairGNN, adversarial debiasing and covariance constraint are applied to ensure the fairness of NT-FairGNN. More specifically, they are based on sensitive attributes $\hat{\mathcal{S}}$ that combine estimated sensitive attributes $\hat{\mathcal{S}}_u$ of nodes $u \in (\mathcal{V} - \mathcal{V}_S)$ and private sensitive attributes $\mathcal{S}_P$. The assumptions in Theorem 1 and 2 are generally true with the sensitive attributes $\hat{\mathcal{S}} = \mathcal{S}_P \cup \hat{\mathcal{S}}_u$ in NT-FairGNN. The flipping noise added to $\mathcal{S}_P$ is irrelevant to $\mathbf{x}$ and learned representation $\mathbf{h}$, and f_E and $f_\mathcal{G}$ do not share any parameters. Thus, the assumption that $p(\hat{s},\mathbf{h}|s) = p(\hat{s}|s)p(\mathbf{h}|s)$ is generally true. Since the loss correction is applied in Eq.(23) to migrate the negative effects of noise in private sensitive attributes, the second assumption in Theorem 1 can also be easily met. Combining objective functions for sensitive attribute estimation with limited and private sensitive attributes, adversarial debiasing and covariance constraint, the overall objective function can be formally written as:

$$\min_{\theta_\mathcal{G},\theta_E} \max_{\theta_A} \mathcal{L}_C + \mathcal{L}_E^P + \alpha\mathcal{L}_R - \beta\mathcal{L}_A, \tag{24}$$

where $\theta_\mathcal{G}$, θ_E, and θ_A are the parameters of classifier, estimator, and adversary, respectively. α and β are scalars to control the contributions of the covariance constraint and adversarial debiasing.

TABLE 3: The comparisons of our proposed methods with the baselines using limited sensitive attributes.

Dataset	Metrics	GCN	GAT	ALFR	ALFR-e	Debias	Debias-e	FCGE	FairGCN	FairGAT	NT-FairGNN
Pokec-z	ACC (%)	70.2 ±0.1	70.4 ±0.1	65.4 ±0.3	68.0 ±0.6	65.2 ±0.7	67.5 ±0.7	65.9 ±0.2	**70.0 ±0.3**	**70.1 ±0.1**	70.0 ±0.1
	AUC (%)	77.2 ±0.1	76.7 ±0.1	71.3 ±0.3	74.0 ±0.7	71.4 ±0.6	74.2 ±0.7	71.0 ±0.2	**76.7 ±0.2**	**76.5 ±0.2**	76.7 ±0.3
	Δ_{SP} (%)	9.9 ±1.1	9.1 ±0.9	2.8 ±0.5	5.8 ±0.4	1.9 ±0.6	4.7 ±1.0	3.1 ±0.5	**0.9 ±0.5**	**0.5 ±0.3**	1.0 ±0.4
	Δ_{EO} (%)	9.1 ±0.6	8.4 ±0.6	1.1 ±0.4	2.8 ±0.8	1.9 ±0.4	3.0 ±1.4	1.7 ±0.6	**1.7 ±0.2**	**0.8 ±0.3**	1.6 ±0.2
Pokec-n	ACC (%)	70.5 ±0.2	70.3 ±0.1	63.1 ±0.6	66.2 ±0.5	62.6 ±0.9	65.6 ±0.8	64.8 ±0.5	**70.1 ±0.2**	**70.0 ±0.2**	70.1 ±0.2
	AUC (%)	75.1 ±0.2	75.1 ±0.2	67.7 ±0.5	71.9 ±0.3	67.9 ±0.7	71.7 ±0.7	69.5 ±0.4	**74.9 ±0.4**	**74.9 ±0.4**	74.9 ±0.4
	Δ_{SP} (%)	9.6 ±0.9	9.4 ±0.7	3.05 ±0.5	4.1 ±0.5	2.4 ±0.7	3.6 ±0.2	4.1 ±0.8	**0.8 ±0.2**	**0.6 ±0.3**	0.8 ±0.2
	Δ_{EO} (%)	12.8 ±1.3	12.0 ±1.5	3.9 ±0.6	4.6 ±1.6	2.6 ±1.0	4.4 ±1.2	5.5 ±0.9	**1.1 ±0.5**	**0.8 ±0.2**	1.1 ±0.3
NBA	ACC (%)	71.2 ±0.5	71.9 ±1.1	64.3 ±1.3	66.0 ±0.4	63.1 ±1.1	65.6 ±2.4	66.0 ±1.5	**71.1 ±1.0**	**71.5 ±0.8**	71.1 ±1.0
	AUC (%)	78.3 ±0.3	78.2 ±0.6	71.5 ±0.3	72.9 ±1.0	71.3 ±0.7	72.9 ±1.2	73.6 ±1.5	**77.0 ±0.3**	**77.5 ±0.7**	77.0 ±0.3
	Δ_{SP} (%)	7.9 ±1.3	10.2 ±2.5	2.3 ±0.9	4.7 ±1.8	2.5 ±1.5	5.3 ±0.9	2.9 ±1.0	**1.0 ±0.5**	**0.7 ±0.5**	1.0 ±0.5
	Δ_{EO} (%)	17.8 ±2.6	15.9 ±4.0	3.2 ±1.5	4.7 ±1.7	3.1 ±1.9	3.1 ±1.3	3.0 ±1.2	**1.2 ±0.4**	**0.7 ±0.3**	1.2 ±0.4

6.3 Training Algorithm of NT-FairGNN

The training algorithm of NT-FairGNN is presented in Algorithm 2. Specifically, f_E is firstly pretrained with Eq.(23) to ensure it meets the second assumption in Theorem 1. Then, we optimize the whole model with Eq.(24) through the ADAM optimizer [60].

7 Experiments

In this section, we conduct experiments to show the effectiveness of the proposed models for fair node classification. In particular, we aim to answer the following questions:

- **RQ1** Can our proposed FairGNN reduce the bias of GNNs while maintaining high accuracy given limited/private sensitive attributes?
- **RQ2** Can our proposed NT-FairGNN on private sensitive attributes ensure the privacy and fairness of GNN and give accurate node classification?
- **RQ3** How do the sensitive attribute estimator, adversarial loss, and covariance constraint affect our frameworks?
- **RQ4** Are FairGNN and NT-FairGNN effective when different numbers of sensitive attributes are provided?

We use the same datasets introduced in Sec. 3.2 for all the experiments. Next, we will begin by introducing compared methods.

7.1 Experimental Settings

7.1.1 Compared Methods

We compare our proposed framework with GCN, GAT, and the following representative and state-of-the-art methods for fair classification and fair graph embedding learning:

- **ALFR** [41]: This is a pre-processing method. A discriminator is applied to remove the sensitive information in the representations produced by a MLP-based autoencoder. Then, linear classifier is trained on the debiased representations.
- **ALFR-e**: To utilize the graph structure information, ALFR-e concatenates the graph embeddings learned by deepwalk [63] with the user features in the ALFR.
- **Debias** [26]: This is an in-processing fair classification method. It directly applies a discriminator on the estimated probability of classifier $\eta : \mathbf{x} \rightarrow \mathbb{R}$. It would make the probability distribution $p(\eta(\mathbf{x})|s = 0)$ closer to $p(\eta(\mathbf{x})|s = 1)$.
- **Debias-e**: Similar to the ALFR-e, we also add the deepwalk embeddings to the features used in Debias.
- **FCGE** [44]: FCGE is proposed to learn fair node embeddings in graph without node features through edge prediction. The sensitive information in the embeddings is filtered by discriminators.
- **NTFC** [21]: This method is proposed to learn a fair classifier with private sensitive attributes on i.i.d data. A constraint of fairness based on the private sensitive attributes is applied to achieve fairness and protect the users' privacy.

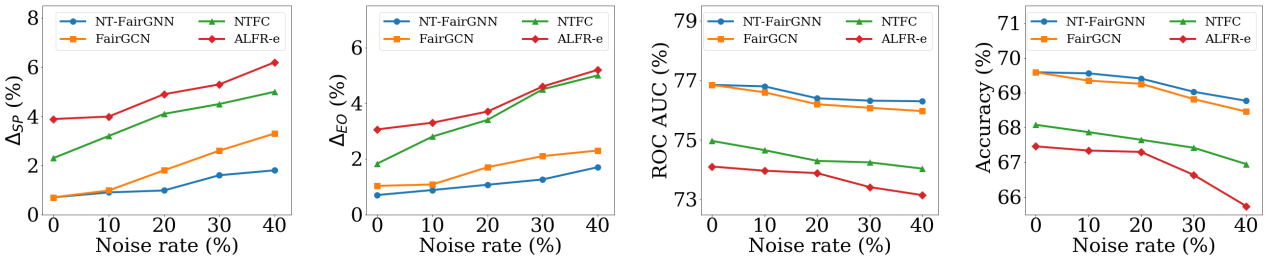
ALFR and ALFR-e are trained with features of all the users $\mathcal{V}$, labels of $\mathcal{V}_L$, and the sensitive attributes of $\mathcal{V}_S$ for fair classification. Debis and Debias-e require the sensitive attributes of labeled nodes, which is on contrary with our setting that $\mathcal{V}_L$ could have no overlap with $\mathcal{V}_S$. Thus, we use the estimated labels of $\mathcal{V}_S$, features of $\mathcal{V}_L$, and labels of $\mathcal{V}_L$ to train Debias and Debias-e. FCGE utilizes $\mathcal{G}$, labels of $\mathcal{V}_L$, and sensitive attributes of $\mathcal{V}_S$. Note that NTFC is proposed for private sensitive attributes, thus it is only compared with NT-FairGNN.

7.1.2 Implementation Details

For FairGNN, we deploy a one hidden layer GCN for f_E. The hidden dimension is set as 128. We use a linear classifier for f_A. To verify that our framework is useful for various GNNs, we adopt both GCN and GAT as the backbone of the FairGNN classifier $f_{\mathcal{G}}$, which are named as **FairGCN** and **FairGAT**. In FairGCN, the GCN classifier contains one hidden layer with dimension 128. The GAT classifier in FairGAT also contains two layers in total. We set the number of heads as 1. The dimensions of the GAT classifiers' hidden layer for Pokec-z, Pokec-n and NBA are 64, 64 and 32, respectively. For NT-FairGNN, we apply GCN as the backbone of sensitive attribute estimation and classifier. The hidden dimiension is set as 128. For hyperparameter selection, we vary α and β among $\{0.0001, 0.001, 0.1, 1\}$ and $\{1, 2, 5, 10, 20, 50, 100\}$, respectively based on the performance on validation set.

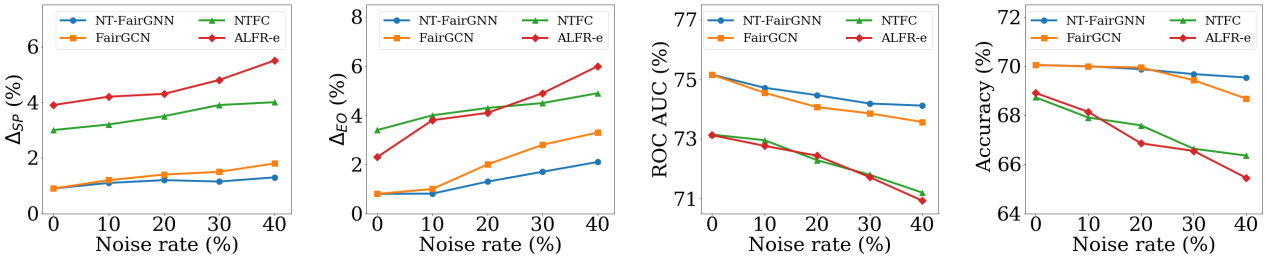
7.2 Performance of FairGNN

To answer **RQ1**, we evaluate our proposed FairGNN in terms of fairness and classification performance. Δ_{SP} and Δ_{EO} are used to show the discrimination level, which are introduced in Section 3.4. The smaller Δ_{SP} and Δ_{EO} are, the more fair the classifier is. Accuracy (ACC) and ROC AUC score are used to evaluate the classification performance. The size of $\mathcal{V}_L$ and $\mathcal{V}_S$ are set as 500 and 200, respectively. Apart from FairGCN and FairGAT, we also compare the results of NT-FairGNN given limited sensitive attributes with the baselines. For all the models, we tune the hyperparameters on the training set via cross validation



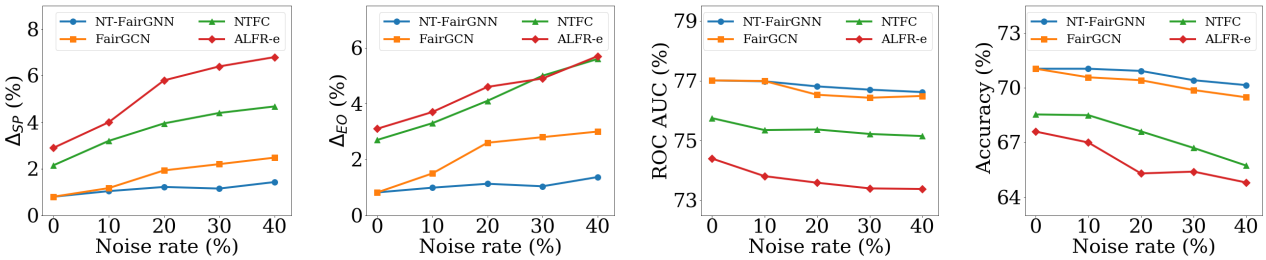
(a) Δ_{SP} (b) Δ_{SP} (c) ROC AUC (d) Accuracy

Fig. 3: Performance of NT-FairGNN on Pokec-z with sensitive attributes containing different levels of noises.



(a) Δ_{SP} (b) Δ_{SP} (c) ROC AUC (d) Accuracy

Fig. 4: Performance of NT-FairGNN on Pokec-n with sensitive attributes containing different levels of noises.



(a) Δ_{SP} (b) Δ_{SP} (c) ROC AUC (d) Accuracy

Fig. 5: Performance of NT-FairGNN on NBA with sensitive attributes containing different levels of noises.

following the description in Sec. 7.1.2. More details about hyperparameter selection will be discussed in Sec. 7.6. All the experiments are conducted 5 times. The mean and standard deviations for all the models on the three datasets are reported in Table 3. From the table, we observe:

- Compared with GCN and GAT, the general fair classification methods and graph embeddings learning method show poor performance in classification even with the help of graph information, while FairGCN and FairGAT perform very close to the based GNNs. This suggests the necessity of investigating fair classification algorithms on GNNs for accurate predictions;
- Under the condition of limited sensitive information, baselines show obvious bias and the ones utilizing graph information are even worse. On the contrary, our proposed models obtain Δ_{SP} and Δ_{EO} that are close to 0, which indicates that the discrimination is basically eliminated; and
- FairGAT is slightly better than FairGCN in Fairness. This is reasonable because the learnable edge coefficients in GAT could be helpful to reduce the weights of the edges that bring bias.
- NT-FairGNN shows similar results to FairGCN. This is because NT-FairGNN will be equivalent to FairGNN when the noise rate ρ is 0.

These observations demonstrate the effectiveness of our proposed frameworks in making fair and accurate predictions.

7.3 Performance of NT-FairGNN

To answer **RQ2**, we compare NT-FairGNN with the baselines on limited private sensitive attributes for different levels of privacy. More specifically, we vary the noise rate ρ as $\{0\%, 10\%, 20\%, 30\%, 40\%\}$ to obtain private sensitive attributes that generated for different levels of privacy protection. The size of labeled nodes $\mathcal{V}_L$ is set as 500, 500, and 100 for Pokec-z, Pokec-n, and NBA. Since the privacy protection will encourage more users share their sensitive attributes for fair node classification, we increase the sizes of $\mathcal{V}_S$ to 1000, 1000, and 100 in Pokec-z, Pokec-n and NBA respectively. The most effective baselines in Table 3 are compared with NT-FairGNN. We also compare our NT-FairGNN with NTC which is designed for private sensitive attributes. Each experiments are conducted 5 times. The results are shown in Fig. 3-5. From the figures, we have the following observations:

- When the noise rate of limited private sensitive attributes is small, both NT-FairGNN and FairGNN perform significantly better than the baselines in terms of fairness. This is because plenty of estimated sensitive attributes can be obtained in FairGNN and NT-FairGNN to address the problem of lacking sensitive attributes.
- As the noise rate increases in the private sensitive attributes, the performance of baselines and FairGCN in fairness and classification will drop, which is as expected. Though NT-FairGNN also drops, it still can obtain Δ_{SP}

and Δ_{EO} smaller than 2%. The classification performance of NT-FairGNN is also more stable. This shows the effectiveness of NT-FairGNN in learning an accurate fair graph neural network with private sensitive attributes.

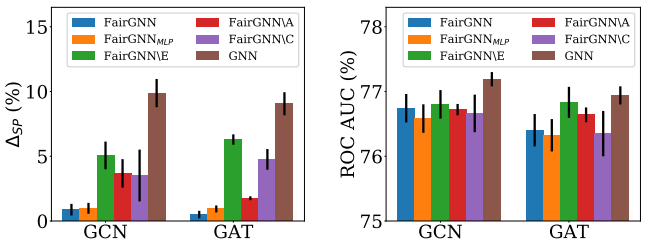
7.4 Ablation Study

To answer **RQ3**, we conduct ablation studies to understand the impacts of f_E, adversarial loss, and covariance constraint on the proposed frameworks.

7.4.1 Ablation Study on FairGNN

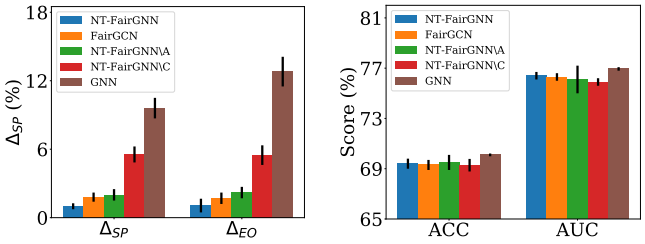
In our proposed FairGNN, a GCN estimator is deployed to predict sensitive attributes for adversarial debiasing. To show the importance of the GCN estimator, we analyze it from two aspects. Firstly, to demonstrate the effectiveness of the noisy sensitive attributes, we eliminate the estimator and only use the provided sensitive attributes $\mathcal{S}$ to get a variant denoted as FairGNN\E. Secondly, to investigate how a weaker estimator would influence the fair classification, we train a variant FairGNN$_{MLP}$ by using MLP as the estimator. To demonstrate the effects of the adversarial loss and covariance constraint, we train two variants of FairGNN, i.e., FairGNN\A and FairGNN\C, where FairGNN\A means FairGNN without the adversarial loss, and FairGNN\C means FiarGNN without covariance constraint. Hyperparameters of these variants are determined by cross validation with gird search which is mentioned in Sec. 7.1.2. For each variant, the experiments are conducted 5 times. The average performance of fairness in terms of Δ_{SP} and node classificaiton in terms of AUC on Pockec-z are presented in Fig. 6, respectively. We only show the results on Pockec-z as we have similar observations on the other datasets. From the figures, we make the following observations:

- The Δ_{SP} score of FairGNN\E is much larger than that of FairGNN. which is because the provided sensitive attributes are inadequate. This shows that f_E plays an important role in FairGNN;
- The performance of sensitive attribute prediction in terms of AUC for MLP estimator is 0.69, which is much lower than that of GCN estimator, which is 0.8. Though FairGNN$_{MLP}$ adopts a much weaker estimator than FairGNN, the performance in terms of fairness is slightly worse than FairGNN. This aligns with our theoretical analysis that f_E doesn't need to be very accurate. However, the differences indicate that too much noise in sensitive attributes may still affect the fairness;
- The Δ_{SP} scores for both FairGNN\C and FairGNN\A are much smaller than that of GNNs in Figure 6, which shows that both covariance constraint and adversarial debiasing can improve fairness; and
- The Δ_{SP} scores for both FairGNN\C and FairGNN\A are much larger than that of FairGNN, which implies that using both covariance constraint and adversarial debiasing can achieve better fairness. This is because they regularize the GNN from two different perspectives, i.e., adversarial debiasing regularizes on the node representations while covariance cosntraint is directly on the predictions for fair classification.



(a) Δ_{SP} (b) ROC AUC

Fig. 6: Comparisons between FairGNN and its variants.



(a) Δ_{SP} (b) ROC AUC

Fig. 7: Comparisons between NT-FairGNN and its variants on Pokec-z with noisy sensitive attributes.

7.4.2 Ablation Study on NT-FairGNN

To reduce the negative effects of the noise in private sensitive attributes, the loss correction is applied to the sensitive attribute estimator in NT-FairGNN. To demonstrates the effectiveness of the loss correction, we compare NT-FairGNN with FairGCN which directly treats private sensitive attributes as the ground truth. We also train two variants of NT-FairGNN, i.e., NT-FairGNN\A and NT-FairGNN\C, to show the effectiveness of adversarial debiasing and covariance constraint in eliminating discrimination in GNN. The hyperparameter selection is based on the process described in Sec. 7.1.2. We only report the results on Pokec-z because similar observations are obtained in other datasets. The sizes of $\mathcal{V}_L$ and $\mathcal{V}_S$ in Pokec-z are set as 500 and 1000, respectively. The noise rate of private sensitive attributes is set as 20%.The average results of 5 runs are shown in Fig. 7. From the figure, we have the following observations:

- With private sensitive attributes which contains random flipping noise for privacy protection, NT-FairGNN performs better than FairGNN in fairness. This is because FairGNN directly applies the private sensitive attributes to train the sensitive attribute estimator, which may result in a poor estimator that cannot meet the requirement of debiasing. On the contrary, loss correction is applied in NT-FairGNN to learn a useful sensitive attribute estimator with private sensitive attributes. Thus, NT-FairGNN can achieve better fairness with private sensitive attributes;
- The discrimination scores of NT-FairGNN\C and NT-FairNN\A are significantly larger than that of NT-FairGNN, which implies that both convariance constraint and adversarial debiasing are effective in eliminating the discrimination with the estimated sensitive attributes. This also suggests that combining the convariance constraint and adversarial debiasing together can lead to a more fair graph neural network.

7.5 Impacts of Sizes of Sensitive Attributes

To answer **RQ4**, we study the impacts of the sizes of $\mathcal{V}_S$ on FairGNN and NT-FairGNN in this section.

7.5.1 Impacts of $|\mathcal{V}_S|$ to FairGNN

We select the GAT as the backbone of FairGNN. The hyperparameters are selected as the description in Sec.7.1.2. We vary $|\mathcal{V}_S|$ as $\{200, 500, 1500, 2000, 2500\}$. Each experiment is conducted 5 times and the average results on Pokec-z with comparison to FairGAT\E and ALFR-e are shown in Fig. 8. From the figure, we observe that:

- Generally, both FairGAT\E and ALFR-e have high discrimination scores when $|\mathcal{V}_S|$ is small. They need plenty of data with sensitive attributes to become effective. FairGAT could get very low Δ_{SP} even when $|\mathcal{V}_S|$ is as small as 200. This implies that FairGAT is insensitive to the size of data with sensitive attributes, which is because we have f_E to estimate the sensitive attributes. Though extremely small $|\mathcal{V}_S|$ would lead to a weak f_E, we still have similar Δ_{SP} score as that when $\mathcal{V}_S$ is large. This verifies our theoretical analysis that we can achieve good fairness with a reasonable f_E;
- FairGAT\E and ALFR-e decrease slightly in classification performance with the increasing of the size of $\mathcal{V}_S$, which is because more data with sensitive attribute would lead to a stricter regularization. In the contrary, FairGAT keeps high classification performance and even perform slightly better with more sensitive attributes. This is because the size of sensitive attributes $\hat{\mathcal{S}}$ used for training FairGAT are fixed to the size of $\mathcal{V}$, and less noise in the estimation of the sensitive attributes is helpful to better learn representations for classification.

7.5.2 Impacts of $|\mathcal{V}_S|$ to NT-FairGNN

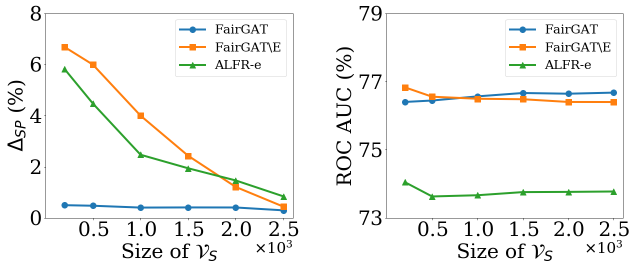
Similarly, we alter the size of private sensitive attributes $|\mathcal{V}_S|$ as $\{200, 500, 1000, 1500, 2000, 2500\}$. The noise rate of private sensitive attribute is set as 20%. Since we have similar observations on other datasets, we only show the comparisons between NT-FairGNN, FairGCN and NTFC on Pokec-z. The average results of 5 runs are reported in Figure 9. We can observe that:

- When the size of private sensitive attributes is small, NT-FairGNN outperforms NTFC by a large margin in fairness. This is because NT-FairGNN learns a robust f_E with limited private sensitive attributes to address the challenge of lacking sensitive attributes;
- With the increasing of $|\mathcal{V}_S|$, the gap between NT-FairGNN and FairGCN decreases. This is because with the increase of $|\mathcal{V}|_S$, we will have plenty of private sensitive attributes to regularize the GNN. And according to Theorem 1, discrimination in GNNs can be eliminated with adequate noisy/private sensitive attributes.

7.6 Parameter Sensitivity Analysis

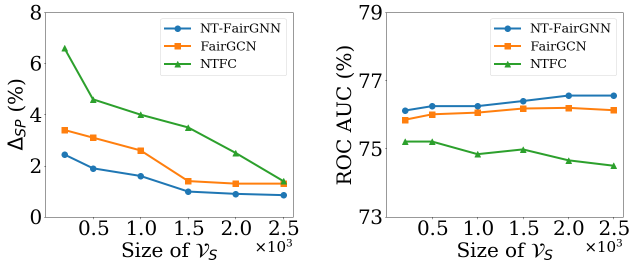
7.6.1 Parameter Sensitivity of FairGNN

There are two important hyperparameters in our proposed frameworks, .i.e., α controlling the influence of the adversary to the GNN classifier, while β controlling the contribution of the covariance constraint to ensure fairness. To investigate the parameter sensitivity and find the ranges that achieve high accuracy with low discrimination score, we train FairGAT models on Pokec-z with various hyperparameters. More specifically, we alter the values of α and β among $\{0.0001, 0.001, 0.01, 0.1, 1\}$ and $\{1, 2, 5, 10, 20, 50, 100\}$. The results are presented in Fig. 10. From Fig. 10 (b), we can find that when $\alpha \leq 0.01$ and $\beta \leq 20$ the classification performance is almost unaffected. Once α and β are too large, the classifier's performance will decay rapidly. The impacts of the hyperparameters to the discrimination score are presented in Fig. 10 (a). When we increase the value of α, Δ_{SP} will firstly decrease as expected. Then, it would increase when the value of α is too large. Because it would be difficult to optimize the GNN classifier to the global minimum when the contribution of the adversary is extremely high. As for β, the discrimination score would consistently reduce when we increase its value. Combining the two Fig.s, we could determine that when $\alpha \in [0.001, 0.01]$ and $\beta \in [5, 20]$, the GNN classifier achieves fairness and maintains high node classification accuracy.



(a) Δ_{SP} (b) ROC AUC

Fig. 8: Impacts of the size of $\mathcal{V}_S$ to FairGNN.

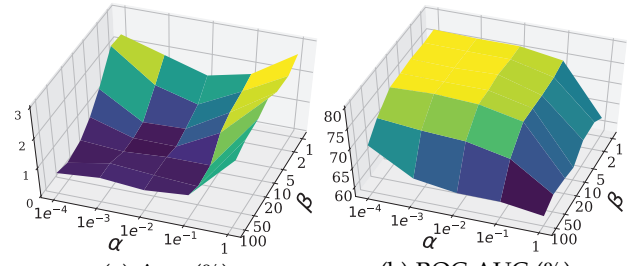


(a) Fairness Scores (b) Classification Scores

Fig. 9: Impacts of the size of $\mathcal{V}_S$ to NT-FairGNN.

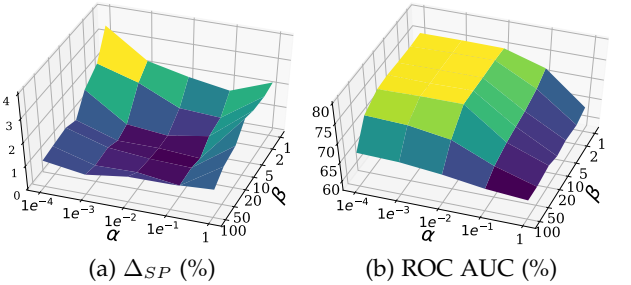
7.6.2 Parameter Sensitivity of NT-FairGNN

Similar to the parameter sensitivity analysis of FairGNN, we train NT-FairGNN on Pokec-z with various hyperparameters to find the ranges that achieve high accuracy with low discrimination score. The values of α and β are varied as $\{0.0001, 0.001, 0.01, 0.1, 1\}$ and $\{1, 2, 5, 10, 20, 50, 100\}$. The noise rates ρ are set as 0.2 for both datasets. The results are presented in Fig. 11. We can observe that (i) With the increasing of α and β, the performance of NT-FairGNN in fairness will firstly increase, while keeping high classification accuracy. But if the α and β is too large, the performance of NT-FairGNN will drop in both fairness and classification. (ii) Similar to FairGNN, NT-FairGNN achieves fairness and maintains high node classification accuracy when $\alpha \in [0.001, 0.01]$ and $\beta \in [5, 20]$, which eases the hyperparameter tuning.



(a) Δ_{SP} (%) (b) ROC AUC (%)

Fig. 10: Parameter sensitivity analysis of FairGNN on Pokec-z with clean sensitive attributes.



(a) Δ_{SP} (%) (b) ROC AUC (%)

Fig. 11: Parameter sensitivity analysis of NT-FairGNN on Pokec-n with noisy sensitive attributes.

8 Conclusion And Future Work

In this paper, we study a novel problem of fair GNN learning with limited and private sensitive information. We empirically demonstrate that GNNs exhibit severe bias. We propose a novel and flexible framework FairGNN for fair node classification with limited sensitive attributes. FairGNN adopts a sensitive attribute estimator to alleviate the issue of lacking sensitive attribute information. With the estimated sensitive attributes, FairGNN designs adversarial debiasing and covariance constraint to regularize the GNN to have fair node representations and predictions, respectively. Furthermore, NT-GNN is proposed to extend FairGNN to handle limited and private sensitive attributes to simultaneously achieve fairness and protect the privacy of users. Theoretical analysis proves that the proposed FairGNN and NT-FairGNN can eliminate the discrimination in GNNs under the corresponding settings of sensitive attributes. Experiment results on real-world datasets demonstrate the effectiveness of the proposed frameworks in terms of both fairness and classification performance.

There are several interesting directions which need further investigation. First, the experiments show that the edges are possible to bring bias. Thus, we will also explore methods which add/delete links in graphs to improve the fairness and classification performance of FairGNN. Second, the privacy budget is uniform to all users by setting a fixed sensitive attribute flipping noise rate in NT-FairGNN. However, the users' sensitivity to the privacy can be various. We may need to add different levels of flipping noises to set customized privacy budgets for users. Thus, we will extend NT-FairGNN to deal with limited and private sensitive attributes that contain various levels of noises.

Acknowledgments

This material is based upon work supported by, or in part by, the National Science Foundation (NSF) under grant IIS-1909702, IIS-1955851, and the Global Research Outreach program of Samsung Advanced Institute of Technology under grant #225003. The findings and conclusions in this paper do not necessarily reflect the view of the funding agency.

References

[1] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[3] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[4] T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto, "Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach," *arXiv preprint arXiv:1706.05674*, 2017.

[5] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1024–1034.

[6] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *SIGKDD*, 2018, pp. 974–983.

[7] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[8] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *ITCS*, 2012, pp. 214–226.

[9] A. Beutel, J. Chen, Z. Zhao, and E. H. Chi, "Data decisions and theoretical implications when adversarially learning fair representations," *arXiv preprint arXiv:1707.00075*, 2017.

[10] E. Creager, D. Madras, J.-H. Jacobsen, M. A. Weis, K. Swersky, T. Pitassi, and R. Zemel, "Flexibly fair representation learning by disentanglement," *arXiv preprint arXiv:1906.02589*, 2019.

[11] Y. Dong, O. Lizardo, and N. V. Chawla, "Do the young live in a" smaller world" than the old? age-specific degrees of separation in a large-scale mobile communication network," *arXiv preprint arXiv:1606.07556*, 2016.

[12] T. A. Rahman, B. Surma, M. Backes, and Y. Zhang, "Fairwalk: Towards fair graph embedding." in *IJCAI*, 2019, pp. 3289–3295.

[13] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[14] H. Suresh and J. V. Guttag, "A framework for understanding unintended consequences of machine learning," *arXiv preprint arXiv:1901.10002*, 2019.

[15] M. Madden, A. Lenhart, S. Cortesi, U. Gasser, M. Duggan, A. Smith, and M. Beaton, "Teens, social media, and privacy," *Pew Research Center*, vol. 21, no. 1055, pp. 2–86, 2013.

[16] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem, "On the fairness of disentangled representations," in *NeurIPS*, 2019, pp. 14 584–14 597.

[17] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," *arXiv preprint arXiv:1511.00830*, 2015.

[18] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[19] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[20] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[21] A. L. Lamy, Z. Zhong, A. K. Menon, and N. Verma, "Noise-tolerant fair classification," *arXiv preprint arXiv:1901.10837*, 2019.

[22] H. Mozannar, M. Ohannessian, and N. Srebro, "Fair learning with private demographic data," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7066–7075.

[23] L. Zhang, Y. Wu, and X. Wu, "Achieving non-discrimination in data release," in *SIGKDD*, 2017, pp. 1335–1344.

[24] M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," *arXiv preprint arXiv:1507.05259*, 2015.

[25] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *WWW*, 2017, pp. 1171–1180.
[26] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *AIES*, 2018, pp. 335–340.
[27] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning adversarially fair and transferable representations," *arXiv preprint arXiv:1802.06309*, 2018.
[28] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016, pp. 3844–3852.
[29] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 97–109, 2018.
[30] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *ICML*, 2016, pp. 2014–2023.
[31] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.
[32] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1725–1735.
[33] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," *arXiv preprint arXiv:1810.05997*, 2018.
[34] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9267–9276.
[35] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," *arXiv preprint arXiv:1907.10903*, 2019.
[36] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *NeurIPS*, 2016, pp. 3315–3323.
[37] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *SIGKDD*, 2015, pp. 259–268.
[38] D. Xu, S. Yuan, L. Zhang, and X. Wu, "Fairgan: Fairness-aware generative adversarial networks," in *Big Data*. IEEE, 2018, pp. 570–575.
[39] ——, "Fairgan+: Achieving fair data generation and classification through generative adversarial nets," in *Big Data*. IEEE, 2019, pp. 1401–1406.
[40] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney, "Fairness gan: Generating datasets with fairness properties using a generative adversarial network," *IBM Journal of Research and Development*, vol. 63, no. 4/5, pp. 3–1, 2019.
[41] H. Edwards and A. Storkey, "Censoring representations with an adversary," *arXiv preprint arXiv:1511.05897*, 2015.
[42] T. Kamishima, S. Akaho, and J. Sakuma, "Fairness-aware learning through regularization approach," in *ICDMW*. IEEE, 2011, pp. 643–650.
[43] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," in *NeurIPS*, 2017, pp. 5680–5689.
[44] A. J. Bose and W. L. Hamilton, "Compositional fairness constraints for graph embeddings," *arXiv preprint arXiv:1905.10674*, 2019.
[45] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *SIGKDD*, 2016, pp. 855–864.
[46] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.
[47] S. Sajadmanesh and D. Gatica-Perez, "Locally private graph neural networks," *arXiv preprint arXiv:2006.05535*, 2020.
[48] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.
[49] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5827–5842, 2019.
[50] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016.
[51] Y. Zhu, X. Yu, M. Chandraker, and Y.-X. Wang, "Private-knn: Practical differential privacy for computer vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 854–11 862.
[52] D. Xu, S. Yuan, and X. Wu, "Achieving differential privacy and fairness in logistic regression," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 594–599.
[53] M. Jagielski, M. Kearns, J. Mao, A. Oprea, A. Roth, S. Sharifi-Malvajerdi, and J. Ullman, "Differentially private fair learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3000–3008.
[54] L. Takac and M. Zabovsky, "Data analysis in public social networks," in *International scientific conference and international workshop present day trends of innovations*, vol. 1, no. 6, 2012.
[55] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *AAAI*, 2018.
[56] H. Wang and J. Leskovec, "Unifying graph convolutional neural networks and label propagation," *arXiv preprint arXiv:2002.06755*, 2020.
[57] J. Liao, C. Huang, P. Kairouz, and L. Sankar, "Learning generative adversarial representations (gap) under fairness and censoring constraints," *arXiv preprint arXiv:1910.00411*, 2019.
[58] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014, pp. 2672–2680.
[59] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.
[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[61] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NeurIPS*, 2016, pp. 2234–2242.
[62] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1944–1952.
[63] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *SIGKDD*, 2014, pp. 701–710.

PLACE PHOTO HERE

Enyan Dai received the B.S. degree in mechanical engineering from University of Science and Technology of China, and M.S. degree in AI from KU Leuven. He is currently working towards his Ph.D. degree at The Pennsylvania State University under the supervision of Professor Suhang Wang. His research interests are: data mining, graph neural networks, and trustworthy AI. He has published innovative works in top conference proceedings such as WSDM, KDD, CIKM, and ICWSM.He also worked as a research intern at IBM in 2021.

PLACE PHOTO HERE

Suhang Wang is an assistant professor of the College of Information Sciences and Technology, The Pennsylvania State University. He received his Ph.D. in Computer Science from Arizona State University in 2018, M.S. in Electrical Engineering from University of Michigan - Ann Arbor in 2013, and B.S. in Electrical and Computer Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012. His research interests are in graph mining, data mining and machine learning. He is an associate editor for several journals and serves as regular journal reviewers and numerous conference program committees. He has published innovative works in highly ranked journals and top conference proceedings such as IEEE TKDE, ACM TIST, KDD, WWW, AAAI, IJCAI, CIKM, SDM, WSDM, ICDM and CVPR, which have received extensive coverage in the media.