Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information

Enyan Dai, Suhang Wang
The Pennsylvania State University
{emd5759,szw494}@psu.edu

ABSTRACT

Graph neural networks (GNNs) have shown great power in modeling graph structured data. However, similar to other machine learning models, GNNs may make predictions biased on protected sensitive attributes, e.g., skin color and gender. 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, the discrimination in GNNs can be magnified by graph structures and the message-passing mechanism. As a result, the applications of GNNs in sensitive domains such as crime rate prediction would be largely limited. Though extensive studies of fair classification have been conducted on i.i.d data, methods to address the problem of discrimination on non-i.i.d data are rather limited. Furthermore, the practical scenario of sparse annotations in sensitive attributes is rarely considered in existing works. Therefore, we study the novel and important problem of learning fair GNNs with limited sensitive attribute information. FairGNN is proposed to eliminate the bias of GNNs whilst maintaining high node classification accuracy by leveraging graph structures and limited sensitive information. Our theoretical analysis shows that FairGNN can ensure the fairness of GNNs under mild conditions given limited nodes with known sensitive attributes. Extensive experiments on real-world datasets also demonstrate the effectiveness of FairGNN in debiasing and keeping high accuracy.

KEYWORDS

Fairness; Graph Neural Networks; Node Classification

ACM Reference Format:
Enyan Dai, Suhang Wang. 2021. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441752

1 INTRODUCTION

Graph neural networks (GNNs) [5, 17, 24, 44] have achieved remarkable performance on various domains such as knowledge graph [16, 46], social media mining [17], nature language processing [24, 49], and recommendation system [2, 50]. Generally, message-passing process is adopted in GNNs [17, 24], 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 [3, 7, 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 [3, 10], 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 [9, 36]. For example, young people tend to build friendship with people of similar age on the social network [9]. 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 which 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 [32] and crime rate prediction [40]. Thus, it is important to investigate fair GNNs.

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 [30]. The lacking of sensitive information challenges many existing work on fair models [3, 7, 28, 29]. *Second*, though extensive efforts have been made to establish fair models by revising features [20, 21, 55], disentanglement [7, 29], adversarial debiasing [3, 11] and fairness constraints [51, 52], 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. Recently, [4, 36] aim to learn fair node representations from graphs. These methods merely deal with plain graphs without any node attributes, and focus on fair node representations instead of fair node classifications.

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441752

Therefore, in this paper, we study a novel problem of learning fair graph neural networks with limited sensitive information. In essence, we need to solve two challenges: (i) how to overcome the shortage of sensitive attributes for eliminating discrimination; and (ii) 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. A GNN sensitive attribute estimator is adopted in FairGNN to predict plenty of sensitive attributes with noise for fair classification. Inspired by existing works of fair classification on i.i.d data with adversarial learning [3, 11, 31, 54], 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. Our main contributions are:

- We study a novel problem of fair graph neutral networks learning with limited sensitive information;
- A new framework, FairGNN, is proposed to settle the shortage of sensitive attributes for adversarial debiasing and fairness constraint by estimating users' sensitive attributes;
- We conduct theoretical analysis showing fairness achieves at the global minimum even with estimated sensitive attributes;
- Extensive experiments on different datasets demonstrate the effectiveness of our methods in eliminating discrimination while keeping high accuracy of GNNs.

The rest of the paper is organized as follows. In Sec. 2, we review related work. In Sec. 3, we conduct preliminary analysis to understand the bias issue of GNNs. In Sec. 4, we give the details of FairGNN. In Sec. 5, we conduct experiments to show the effectiveness of FairGNN. In Sec. 6, we conclude with future work.

2 RELATED WORK

In this section, we will review related work including graph neural networks and fairness in machine 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 [16, 39, 42, 43, 49, 50, 56]. Generally, GNNs can be categorized into two categories, i.e., spectral-based [5, 8, 19, 24, 25] and spatial-based [6, 17, 44, 50]. Spectral-based GNNs define graph convolution based on spectral graph theory, which is first explored by Bruna et al. [5]. Since then, more spectral-based methods are developed for further improvements and extensions [8, 19, 24, 25]. Graph Convolutional Network (GCN) [24] is a particularly popular method which simplifies the convolutional operation on the graph. Spatial-based graph convolution directly updates the node representation by aggregating its neighborhoods' representations [13, 17, 33, 50]. Veličković et al. [44] introduce the self-attention into the aggregation of spatial graph convolution by assigning higher weights to the more important nodes in graph attention network (GAT). Various spatial methods are proposed to solve the scalability issue of GCN [6, 17]. For example, a neighbor sampling method to train GNN with nodes in mini-batch instead of the whole graph is developed in GraphSAGE [17]. Moreover, spatial-based methods have already been successfully deployed to deal with extremely large industrial datasets [50].

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, there is no existing work to address this problem. Thus, 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 [3, 10, 18, 20, 21, 28, 55]. 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 [20, 55], revising attributes of data [12, 21], generating non-discriminatory labeled data [38, 47, 48], and obtaining fair data representations [3, 7, 11, 28, 29, 53]. 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. [10, 22, 52, 54]. Finally, the post-processing approaches directly change the predictive labels to ensure fairness [18, 35]. Recently, several works explore the learning of fair graph embeddings for recommendation [4, 36]. Fairwalk [36] modifies the random walk procedure of node2vec [15] to obtain a more diverse network neighborhood representations. The sensitive attributes of all the nodes are required in the sampling procedure of FairWalk. Bose and Hamilton [4] propose to add discriminators to eliminate the sensitive information in the graph embeddings. Similar to Fairwalk, the training process of the discriminators is in need of 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 sensitive attributes in practice.

3 PRELIMINARIES ANALYSIS

In this section, we first conduct preliminary analysis on real-world datasets to show that GNNs could exhibit more serve bias due to the graph structure and the message-passing. Sequentially, We formally give the problem definition of fair node classification.

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

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
Group ratio	1.84	2.46	2.77
# of inter-group edges	39,804	31,515	4,401
# of intra-group edges	842,961	697,614	12,169

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 $\mathcal{S}$.

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 [41]: It 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 [1] 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 [2]. 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.

3.3 Preliminaries of Graph Neural Networks

Graph neural networks (GNNs) 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 $y_v = f(\mathbf{h}_v)$. Current GNNs are neighborhood aggregation approaches, which will update the representations of the nodes with the representations of the neighborhood nodes. The representations after k layers' aggregation would capture the structural information of the k-hop network neighborhoods. 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 a set of neighborhoods of v.

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

3.4 Fairness Evaluation Metrics

In this subsection, we will 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: $\mathbf{x} \rightarrow y$.

Definition 3.1. (Statistical Parity [10]). 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 3.2. (Equal Opportunity [18]). 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. According to [3, 29], we apply the following metrics to quantitatively evaluate 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 [52], SVM [52], and MLP [11] 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

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

GNNs, edges in graph can bring linking bias, i.e., the misrepresentation due to the connections of users [32]. It has been proven that the embeddings of nodes within the connected component will be closer after one aggregation in GCN [26, 45]. 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** [24]: A state-of-the-art spectral graph neural network.
- **GAT** [44]: A spatial graph neural network 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.*

3.6 Problem definition

Our preliminary analysis verifies that GNNs have severe bias issue. Thus, it is important to develop fair GNNs. Following existing work of fair models [3, 12, 29, 47], 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 fair GNN problem 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.,*

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

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 satisfying the fairness criteria such as statistical parity.

4 METHODOLOGY

In this section, we give the details of FairGNN. An illustration of the proposed framework is shown in Figure 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

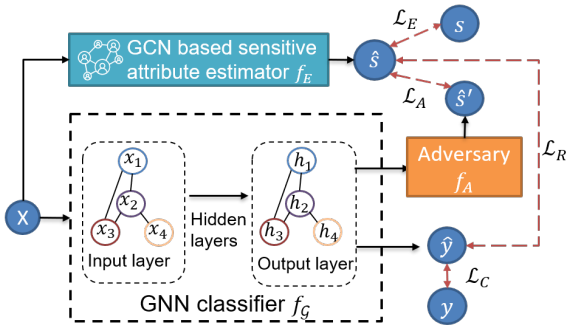


Figure 1: The overall framework of FairGNN.

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 theoretical proof.

4.1 The GNN Classifier $f_{\mathcal{G}}$

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 [24] and GAT [44]. Let $f_{\mathcal{G}}^{(k)}$ denote 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:

$$\mathbf{h}_v = f_{\mathcal{G}}^{(K)}(\mathbf{x}_v, \mathcal{N}_v^{(K)}), \tag{7}$$

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:

$$\hat{y}_v = \sigma(\mathbf{h}_v \cdot \mathbf{w}), \tag{8}$$

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

$$\min_{\theta_{\mathcal{G}}} \mathcal{L}_C = -\frac{1}{|\mathcal{V}_L|} \sum_{v \in \mathcal{V}_L} [y_v \log \hat{y}_v + (1 - y_v) \log (1 - \hat{y}_v)], \tag{9}$$

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.

4.2 Adversarial Debiasing with Estimator f_E

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 [3, 11, 27, 31]. 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 small size $\mathcal{V}_S$. 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 $f_E : \mathcal{G} \rightarrow \mathcal{S}$ 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

$$\min_{\theta_E} \mathcal{L}_E = -\frac{1}{|\mathcal{V}_S|} \sum_{v \in \mathcal{V}_S} [s_v \log \hat{s}_v + (1 - s_v) \log (1 - \hat{s}_v)], \quad (10)$$

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.

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

$$\begin{aligned} \min_{\theta_{\mathcal{G}}} \max_{\theta_A} \mathcal{L}_A = & \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\hat{s}=1)} [\log(f_A(\mathbf{h}))] \\ & + \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\hat{s}=0)} [\log(1 - f_A(\mathbf{h}))], \end{aligned} \quad (11)$$

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 4.1. *The global minimum of Eq.(11) 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 final layer representation learned by the K-layer GNN classifier $f_{\mathcal{G}}$.*

Proof. According to Proposition 1. in [14], 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.(11) could be reformulated as minimizing this function:

$$\begin{aligned} C^s = & \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\hat{s}=1)} \Big[\log \frac{p(\mathbf{h}|\hat{s}=1)}{p(\mathbf{h}|\hat{s}=1) + p(\mathbf{h}|\hat{s}=0)} \Big] \\ & + \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\hat{s}=0)} \Big[\log \frac{p(\mathbf{h}|\hat{s}=0)}{p(\mathbf{h}|\hat{s}=1) + p(\mathbf{h}|\hat{s}=0)} \Big] \\ & = -\log(4) + 2 \cdot JSD(p(\mathbf{h}|\hat{s}=1) || p(\mathbf{h}|\hat{s}=0). \end{aligned} \quad (12)$$

The Jensen-Shannon divergence between two distributions is nonnegative, 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 4.2. *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.(11) 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 4.1, 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

$$\sum_s p(\mathbf{h}|s)p(s|\hat{s} = 1) = \sum_s p(\mathbf{h}|s)p(s|\hat{s} = 0) \quad (13)$$

Reordering the terms in Eq.(13), we can get

$$\begin{aligned} \frac{p(\mathbf{h}|s=1)}{p(\mathbf{h}|s=0)} &= \frac{p(s=0|\hat{s}=1) - p(s=0|\hat{s}=0)}{p(s=1|\hat{s}=0) - p(s=1|\hat{s}=1)} \\ &= \frac{(1 - p(s=1|\hat{s}=1)) - (1 - p(s=1|\hat{s}=0))}{p(s=1|\hat{s}=0) - p(s=1|\hat{s}=1)} \\ &= 1 \end{aligned} \quad (14)$$

Eq.(14) 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.(11) 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.

4.3 Covariance Constraint

The instability of the training process of adversarial learning is well known [1]. In adversarial debiasing, failure to coverage may result in a classifier with discrimination. To alleviate this issue, we add a covariance constraint [51, 52] on the output of $f_{\mathcal{G}}$ to help the model achieve fairness. The covariance constraint has been explored in [51, 52] 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 $\hat{y}$ as

$$\mathcal{L}_R = |\text{Cov}(\hat{s}, \hat{y})| = |\mathbb{E}[(\hat{s} - \mathbb{E}(\hat{s}))(\hat{y} - \mathbb{E}(\hat{y}))]|, \quad (15)$$

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

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
1: Initialize f_E by optimizing Eq.(10) w.r.t θ_E
2: **repeat**
3: Obtain the estimated sensitive attributes with f_E
4: Optimize the GNN classifier parameters $\theta_{\mathcal{G}}$, the adversary parameters θ_A, and the estimator parameters θ_E by Eq.(17).
5: **until** convergence
6: **return** $f_{\mathcal{G}}$, f_A, and f_E

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 4.3. *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. $\hat{y} \perp s$, $\hat{y}$ 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, $p(\hat{y}|s, \hat{s}) = p(\hat{y}|s)$ could be derived. When $\hat{y} \perp s$, the distribution $p(\hat{y}, \hat{s})$ would be:

$$p(\hat{y}, \hat{s}) = \sum_s p(\hat{y}|s)p(\hat{s}, s) = \sum_s p(\hat{y})p(\hat{s}, s) = p(\hat{y})p(\hat{s}). \tag{16}$$

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 use the first assumption in Theorem 4.3, which is generally valid as discussed previously.

4.4 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 could be formulated as:

$$\min_{\theta_{\mathcal{G}}, \theta_E} \max_{\theta_A} \mathcal{L}_C + \mathcal{L}_E + \alpha \mathcal{L}_R - \beta \mathcal{L}_A, \tag{17}$$

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.

4.5 An Training Algorithm of FairGNN

The training algorithm of FairGNN is presented in Algorithm 1. Specially, we first pretrain f_E to ensure it meets the second assumption in Theorem 4.2. Sequentially, we optimize the whole model with Eq.(17) through the ADAM optimizer [23]. 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 [37].

5 EXPERIMENTS

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

- **RQ1** Can the proposed FairGNN reduce the bias of GNNs while maintaining high accuracy?
- **RQ2** How do the sensitive attribute estimator, adversarial loss, and covariance constraint affect FairGNN?
- **RQ3** Is FairGNN effective when different amount of sensitive attributes or labels are provided in the training set?

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

5.1 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** [11]: 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 [34] with the user features in the ALFR.
- **Debias** [54]: This is an in-processing fair classification method. It directly applies an discriminator on the estimated probability of classifier $\eta : \mathbf{x} \to \mathbb{R}$. 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** [4]: 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.

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 $\mathcal{V}_S$.

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.

5.2 Fair Classification on Graph

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. For all the models, we tune the hyperparameters on the training set via cross validation. For FairGCN, we set α to 100 and β to 1. For FairGAT, α is 2 and β is 0.1. More details about hyperparameter selection will be discussed in Sec 5.5. All the experiments

Table 3: The comparisons of our proposed methods with the baselines.

Dataset	Metrics	GCN	GAT	ALFR	ALFR-e	Debias	Debias-e	FCGE	FairGCN	FairGAT
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
	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
	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
	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
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
	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
	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
	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
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
	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
	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
	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

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 make the following observations:

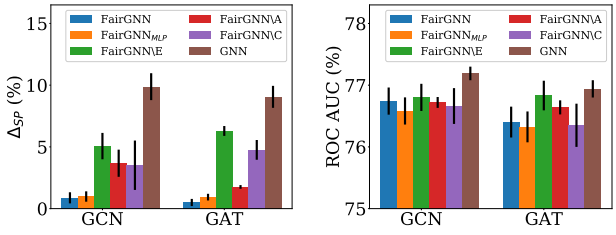
- Compared with GCN and GAT, the general fair classification methods and graph embeddings learning method show poor performance in classification even with the help of graph information, while FairGCN and FairGAT perform very close to the based GNNs. This suggests the necessity of investigating fair 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.

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

5.3 Ablation Study

To answer **RQ2**, we conduct ablation studies to understand the impacts of f_E, adversarial loss, and covariance constraint.

5.3.1 Impact of f_E. In our proposed framework, 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. Hyperparameters of these variants are determined by cross validation with gird search. Specifically, we vary α and β among $\{0.0001, 0.001, 0.1, 1\}$ and $\{1, 2, 5, 10, 20, 50, 100\}$, respectively. 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. 2(a) and (b), 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:



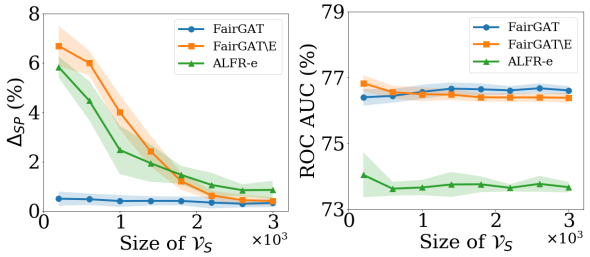
(a) Δ_{SP} **(b) ROC AUC**

Figure 2: Comparisons between FairGNN and its variants.

- 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; and
- 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 0.80. 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 marginal differences still indicate that too much noise in sensitive attributes may still slightly affect the fairness.

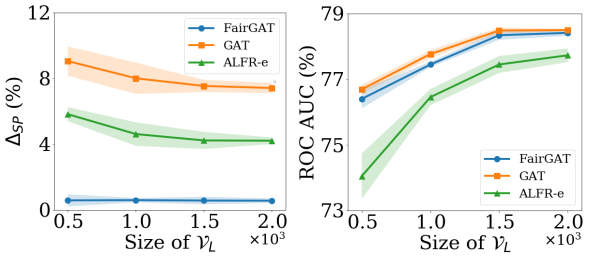
5.3.2 Impacts of the adversarial debiasing and covariance constraint. To demonstrate the effects of the adversarial loss and covariance constraint, we train two variants of FairGNN, i.e., FairGNN\A and FairGNN\C, where FairGNN\A means FairGNN without the adversarial loss, and FairGNN\C means FiarGNN without covariance constraint. Similarly, for each variant, we run the experiment 5 times on Pokec-z and the average performances are shown in Figure 2. From the figure, we observe:

- The Δ_{SP} scores for both FairGNN\C and FairGNN\A are much smaller than that of GNNs in Figure 2, 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

Figure 3: Impacts of the size of $\mathcal{V}_S$ to FairGAT.



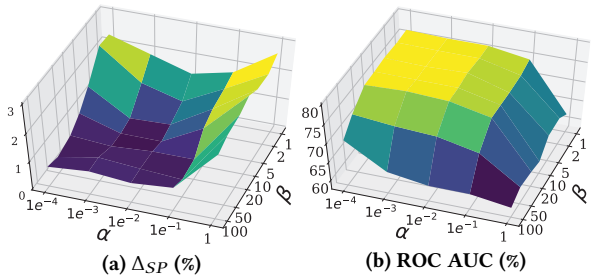
(a) Δ_{SP} (b) ROC AUC

Figure 4: Impacts of the size of $\mathcal{V}_L$ to FairGAT.

5.4 Impacts of Sizes of $\mathcal{V}_S$ and $\mathcal{V}_L$

To answer **RQ3**, we study the impacts of the sizes of $\mathcal{V}_S$ and $\mathcal{V}_L$ on FairGAT. We set $\alpha = 0.1$ and $\beta = 2$ based on cross validation. We vary $|\mathcal{V}_S|$ as $\{200, 600, 1000, 1400, 1800, 2200, 2600, 3000\}$. 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. 3. From the figure, we observe that: (i) 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; (ii) 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.

Similarly, we vary $|\mathcal{V}_L|$ as $\{500, 1000, 1500, 2000\}$ and each experiment is run for 5 times. The average results on Pokec-z are reported in Figure 4a. We only report the results on Pokec-z as we have simialr observations on other datasets. From the figure, we observe that: FairGAT consistently shows effectiveness in eliminating discrimination. The drop in classification performance is marginal. This demonstrates that our proposed method could achieve fairness while keep high accuracy in general scenarios which correspond to various sizes of $\mathcal{V}_S$ and $\mathcal{V}_L$.



(a) Δ_{SP} (%) (b) ROC AUC (%)

Figure 5: Parameter sensitivity analysis.

5.5 Parameter Sensitivity

There are two important hyperparameters in our proposed model, .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 Figure 5. From Figure 5 (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 Figure 5 (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 figures, 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.

6 CONCLUSION AND FUTURE WORK

In this paper, we study a novel problem of fair GNN learning with limited sensitive information. We empirically demonstrate that GNNs exhibit severe bias. We propose a novel and flexible framework FairGNN which is able to significantly alleviate the bias issue of GNNs meanwhile maintain high performance on node classification. 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. We theoretically show that FairGNN can reduce the bias. Experiment results on real-world datasets demonstrate the effectiveness of the proposed framework in terms of both fairness and classification performance. There are several interesting directions which need further investigation. First, we assume the provided sensitive attributes are clean. However, for some applications in social media, users might randomly input sensitive attributes such as gender due to privacy concern. Thus, we will extend FairGNN to deal with limited and inaccurate sensitive information. Second, 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.

7 ACKNOWLEDGEMENTS

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] Martin Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862* (2017).
[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
[3] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075* (2017).
[4] Avishek Joey Bose and William L Hamilton. 2019. Compositional fairness constraints for graph embeddings. *arXiv preprint arXiv:1905.10674* (2019).
[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).
[6] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks. In *SIGKDD*. 257–266.
[7] Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa A Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. 2019. Flexibly fair representation learning by disentanglement. *arXiv preprint arXiv:1906.02589* (2019).
[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*. 3844–3852.
[9] Yuxiao Dong, Omar Lizardo, and Nitesh V Chawla. 2016. 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).
[10] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *ITCS*. 214–226.
[11] Harrison Edwards and Amos Storkey. 2015. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897* (2015).
[12] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *SIGKDD*. 259–268.
[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212* (2017).
[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NeurIPS*. 2672–2680.
[15] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*. 855–864.
[16] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2017. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. *arXiv preprint arXiv:1706.05674* (2017).
[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.
[18] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *NeurIPS*. 3315–3323.
[19] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* (2015).
[20] Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In *ICCC*. IEEE, 1–6.
[21] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *KAIS* 33, 1 (2012), 1–33.
[22] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *ICDMW*. IEEE, 643–650.
[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[24] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[25] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. 2018. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing* 67, 1 (2018), 97–109.
[26] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*.
[27] Jiachun Liao, Chong Huang, Peter Kairouz, and Lalitha Sankar. 2019. Learning generative adversarial representations (GAP) under fairness and censoring constraints. *arXiv preprint arXiv:1910.00411* (2019).
[28] Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. 2019. On the fairness of disentangled representations. In *NeurIPS*. 14584–14597.
[29] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. 2015. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830* (2015).
[30] Mary Madden, Amanda Lenhart, Sandra Cortesi, Urs Gasser, Maeve Duggan, Aaron Smith, and Meredith Beaton. 2013. Teens, social media, and privacy. *Pew Research Center* 21, 1055 (2013), 2–86.
[31] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2018. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309* (2018).
[32] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* (2019).
[33] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *ICML*. 2014–2023.
[34] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*. 701–710.
[35] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. In *NeurIPS*. 5680–5689.
[36] Tahleen A Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding.. In *IJCAI*. 3289–3295.
[37] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *NeurIPS*. 2234–2242.
[38] Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. 2019. Fairness GAN: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development* 63, 4/5 (2019), 3–1.
[39] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. 2019. Node injection attacks on graphs via reinforcement learning. *WWW* (2019).
[40] Harini Suresh and John V Guttag. 2019. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002* (2019).
[41] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International scientific conference and international workshop present day trends of innovations*, Vol. 1.
[42] Xianfeng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. 2020. Transferring Robustness for Graph Neural Network Against Poisoning Attacks. In *WWW*. 600–608.
[43] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. 2020. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In *CIKM*. 1435–1444.
[44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[45] Hongwei Wang and Jure Leskovec. 2020. Unifying graph convolutional neural networks and label propagation. *arXiv preprint arXiv:2002.06755* (2020).
[46] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*. 349–357.
[47] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2018. Fairgan: Fairness-aware generative adversarial networks. In *Big Data*. IEEE, 570–575.
[48] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2019. FairGAN+: Achieving Fair Data Generation and Classification through Generative Adversarial Nets. In *Big Data*. IEEE, 1401–1406.
[49] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *AAAI*, Vol. 33. 7370–7377.
[50] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*. 974–983.
[51] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*. 1171–1180.
[52] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2015. Fairness constraints: Mechanisms for fair classification. *arXiv preprint arXiv:1507.05259* (2015).
[53] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *ICML*. 325–333.
[54] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *AIES*. 335–340.
[55] Lu Zhang, Yongkai Wu, and Xintao Wu. 2017. Achieving non-discrimination in data release. In *SIGKDD*. 1335–1344.
[56] Tianxiang Zhao, Xianfeng Tang, Xiang Zhang, and Suhang Wang. 2020. Semi-Supervised Graph-to-Graph Translation. In *CIKM*. 1863–1872.