Learning a fair and privacy-preserving graph neural network from private and limited sensitive attributes

Xuemin Wang [1], Tianlong Gu [2], xuguang bao [2], and Liang Chang [2]

[1]Guangxi Key Laboratory of Trusted Software
[2]Affiliation not available

October 31, 2023

Abstract

Learning a fair and privacy-preserving graph neural network from private and limited sensitive attributes

Learning a fair and privacy-preserving graph neural network from private and limited sensitive attributes

Xuemin Wang, Tianlong Gu, *Senior Member, IEEE,* Xuguang Bao, and Liang Chang

***Abstract*—Graph neural networks (GNNs) have been applied in various high-stake scenarios to make decisions. However, the successful adoption of GNNs may raise ethical issues such as fairness and privacy. In most recent studies, these issues are addressed separately, disregarding their potential trade-off. In this paper, we propose a novel framework called FPGNN (Fair and Privacy-preserving GNN) on limited sensitive attributes. Specifically, FPGNN promotes individual fairness by minimizing the difference between two ranking lists derived from the input and output spaces using differentiable ranking metrics. To defend against the attribute inference attack in the downstream tasks, FPGNN purges the information about sensitive attributes from the released graph embedding using adversarial training. Besides, FPGNN consists of a utility maximization module to preserve competitive accuracy. Furthermore, we consider a situation in which the collected sensitive attributes are protected by local differential privacy (LDP). In this situation, the attacker employs methods that can learn from the noisy label, to perform the attribute inference attacks. To defend against this kind of attack, we extend FPGNN to PL-FPGNN (Private and Limited sensitive attributes-Fair and Privacy-preserving GNN). Experimental results on three benchmark datasets demonstrate that our methods achieve a good balance among fairness, utility, and privacy.**

***Index Terms*—Individual fairness, attribute inference attack, graph neural network, trustworthy.**

I. Introduction

GRAPH-STRUCTURED data are ubiquitous in the real world, such as social networks [1], knowledge graphs [2], and trading networks [3]. To better understand such data, various graph mining algorithms have been proposed. Among these algorithms, graph neural networks (GNNs) [4] have demonstrated remarkable performance. Besides, they are increasingly adopted in high-stake scenarios such as credit scoring [5], fack review detection [6], and medical diagnosis [7]. Although GNNs have excelled in accomplishing corresponding tasks in these scenarios, adopting GNNs directly could empirically result in fairness issues and privacy issues. Specifically, GNNs may inherit societal bias from the graph data [8]. For example, GNNs may give unfair credit scores to low-income people. Besides, recent studies have shown that GNNs are vulnerable to attribute inference attacks [9]. For example, the attacker leveraged Facebook data such as user linkage, gender, and other attributes to infer users' sensitive attributes such as sexual orientation. Therefore, lacking either fairness consideration or privacy considerations may cause unanticipated harm to humans and society.

A wide spectrum of fair GNNs has been developed to mitigate the bias of GNNs. Existing fairness notations mainly consist of group fairness and individual fairness [10]. Group fairness aims to provide equal outcome rates for people in different demographic subgroups (e.g., age, gender, and race). Since group fairness focuses on the bias for a specific group, it only eliminates limited forms of bias. However, the bias for graph data exists in various shapes and formats as the graph data is heterogenous and comprises various data modalities such as node features and edges. It is necessary to scrutinize the bias at a much finer level of granularity. To address this challenge, individual fairness is proposed to consider atomic components of graphs such as nodes. It requires similar nodes to receive similar outcomes. The formulation based on the Lipschitz condition [11] requires the distances of any node pairs in the output space should be smaller or equal to the corresponding distance in the input spaces. However, it is difficult to calibrate the difference between the two individuals. To tackle this challenge, ranking-based individual fairness is proposed [12]. This fairness focuses on a ranking list that consists of other nodes in descending order according to their similarity to the chosen node. It first derives two ranking lists for each node from input space and output space and then ensures the two ranking lists are consistent. However, this individual fairness-aware GNN also encounters privacy issues. An existing method called LPF-IFGNN [13], aims to promote ranking-based individual fairness while protecting node privacy. Since LPF-IFGNN assumes that the trust third party is absent, it needs to perturb the nodes and labels using local differential privacy and publish the noisy data to the server. This way can provide strict privacy protection, while fairness and utility promotion is limited. To better promote individual fairness, we consider a scenario involving a trusted third party for learning graph representation. It is noted that vector representation contains significant individual information and the downstream task requires a classifier to accomplish tasks such as node classification. In this scenario, the attacker can train a classifier to predict the sensitive attributes using the released graph embedding and the collected sensitive attributes. Hence, the sensitive factor is necessary to be purged from the released graph embeddings for the downstream tasks.

In this paper, we introduce a novel problem of promoting ranking-based individual fairness, defending against attribute inference attacks, and preserving competitive utility perfor-

This work was supported in part by the National Natural Science Foundation of China under Grant U22A2099, Grant 62006057, and Grant 61966009. (Corresponding authors: Tianlong Gu and Xuguang Bao).

Xuemin Wang, Xuguang Bao, and Liang Chang are with the Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China (xueminwangbetter@163.com, bbaaooxx@163.com, changl@guet.edu.cn).

Tianlong Gu is with the College of Cyber Security. Jinan University, Guangzhou 510632, China (gutianlong@jnu.edu.cn)

mance. To address this problem, we face three challenges: **(1) Efficiently promoting ranking-based individual fairness.** For fairness promotion, the ranking-based individual fairness can be seen as the ranking accuracy of the prediction matrices, which can be measured by NDCG@K. However, existing methods such as REDRESS [12] and LPF-IFGNN [13] optimize a surrender loss rather than optimizing the ranking metrics. In this way, the approximation quality of surrogate loss is controlled by the number of samples. Besides, sometimes, the surrogate loss is loosely related to the target loss [14], which results in the inefficiency of promoting ranking-based individual fairness. Hence we need to better measure the changes of similarity and position information in the loss function and promote individual fairness more efficiently. **(2) Mitigating the leakage of sensitive attributes from the graph embedding.** In this paper, we consider a more realistic scenario where the sensitive attributes are private and limited. This consideration arises from the fact that, despite the presence of a trusted third party, private users may not disclose their sensitive attributes Furthermore, the limited sensitive attributes may be perturbed by local differential privacy (LDP). To explore the privacy issues in this situation, we propose a novel attribute inference attack approach for the private and limited sensitive attributes. Specifically, the attacker accesses the privacy budget and trains a GNN model using forward correction loss [15], which can infer the sensitive attributes with high accuracy. Thus, we need to defend against these kinds of attribute inference attacks on private and limited sensitive attributes. **(3) Balancing fairness, privacy, and utility.** Except for fairness promotion and privacy protection, competitive utility is also essential to be preserved. The superior performance of GNNs on various downstream tasks benefits from the end-to-end learning methods. Therefore we need to incorporate fairness and privacy requirements into the training process while preserving competitive utility performance.

To tackle these challenges, we propose a novel GNN training algorithm called FPGNN (Fair and Privacy-preserving Graph Neural Network). FPGNN comprises three modules: individual fairness promotion module, privacy-preserving module, and utility maximization module. The individual fairness promotion module formulates the loss function by employing differentiable ranking metrics to measure the difference between two ranking lists from the input and the output spaces. The privacy-preserving module employs a sensitive attribute estimator to provide accurate sensitive attributes. Using these sensitive attributes, it employs adversarial training to improve the privacy performance of released graph embedding. The utility module aims to preserve competitive accuracy for downstream tasks. Furthermore, we consider a situation in which limited sensitive attributes are protected by LDP. To explore privacy issues in this situation, we propose a novel attribute inference attack. To defend against the novel inference attack, we further propose PL-FPGNN (Private and Limited sensitive attributes-Fair and Privacy-preserving Graph Neural Network). Specifically, we train a sensitive attribute estimator using forward correction loss to provide accurate and clean sensitive attributes. Based on these sensitive attributes, adversarial training can purge the individual information about sensitive attributes from the graph embeddings more efficiently. Experiments on three real datasets demonstrate the effectiveness of the proposed model in balancing fairness promotion, privacy protection, and utility maximization.

The contributions of our work are summarized as follows: (1) We propose a novel method that leverages differentiable ranking metrics to effectively promote ranking-based individual fairness. (2) We introduce an adversarial training method on private and limited sensitive attributes to defend against attribute inference attacks. (3) We propose a novel framework FPGNN, which can promote individual fairness and mitigate the individual privacy leakage issues of private users while preserving high accuracy on limited sensitive attributes. (4) We propose a novel attribute inference attack approach for the limited sensitive attributes protected by LDP. To defend against this attack, we further propose the framework PL-FPGNN.

The rest of this paper is organized as the following. Section 2 summarizes the related work of fair GNNs, privacy-preserving GNNs, and fair and privacy-preserving GNNs. Section 3 introduces the background knowledge required for our study, and presents the definitions of our problems. Section 4 describes the details of the proposed FPGNN framework and training algorithm. Section 5 shows the details of the proposed PL-FPGNN framework and training algorithm. Section 6 provides details of the experiments and a discussion of the experiment results. Section 7 concludes the paper and presents future research directions for fairness and privacy in GNNs.

II. RELATED WORK

A. Fair GNNs

Many works have been conducted to deal with the bias in graph representation learning. The fairness notation can be categorized into group fairness, counterfactual fairness, and individual fairness [10]. *Group fairness* ensures equal outcome statistics such as true positives across different groups. Adversarial learning is a popular strategy for learning a graph embedding align with the group fairness. Bose et al. [16] employed a discriminator to predict the sensitive attributes using graph embedding. The generator continually generates the graph embeddings until the embeddings are indistinguishable w.r.t. sensitive attributes. Dai et al. [17] achieved group fairness using adversarial training for situations where sensitive attributes are limited and perturbed using differential privacy techniques. Wang et al. [18] generated fair views of the graph by identifying and masking sensitive-correlated features and clamped weights of the encoder to give up the sensitive-related features, which can mitigate the biased caused by feature propagation. *Counterfactual fairness* ensures the prediction for each individual and its counterfactuals are the same. Agarwal et al. [19] introduced a novel objective function to achieve fair and stable representations both and developed a layer-wise weight normalization to promote fairness and the stability of the graph representation. Ma et al. [20] mitigated graph unfairness by generating counterfactuals and minimizing the discrepancy between the representations that are learned from the original graph and the counterfactual of each node to achieve counterfactual fairness. *Individual fairness* requires

treating similar people similarly. Kang et al. [11] introduced node pair distance-based fairness, which requires satisfying the Lipschitz condition for node-wise distances calculated in both input and output spaces. Song et al. [21] focused on addressing discrimination arising from group disparities in optimizing individual fairness. Specifically, this discrimination is caused by the variation in scalers used in the Lipschitz condition for different groups. Dong et al. [13] proposed node-ranking-based fairness, where they computed similarity ranking lists for each node and enforced consistent rankings between the input and output spaces.

In summary, individual fairness focuses on atomic components of a graph such as nodes rather than a specific protected group. It is a finer fairness guarantee, which can mitigate more forms of bias. However, existing methods for promoting ranking-based fairness employ a surrogate loss measuring the difference between two rankings. Since the instability of surrogate loss, we propose a novel fairness promotion method that benefits from the differentiable ranking metrics [14].

B. Privacy-preserving GNNs

Recent studies indicate that graph neural network is vulnerable to privacy attack which aims to extract sensitive information that users aren't intended to publish, including membership inference attacks, attribute inference attacks, property inference attacks, and model extraction attack [8]. In this paper, we focus on attribute inference attacks. To avoid extracting node-level information about sensitive attributes, Liao et al. [22] proposed a minimax game between desired GNN encoder and the worst-case attacker. Li et al. [23] considered another inference attack that infers users' sensitive attributes from the node representation and proposed a graph adversarial training network that removes the sensitive factors from the learned node representation using disentangling and purging mechanisms. To defend against inference attacks, Jiang et al. [24] employed secure aggregation to achieve privacy-preserving federated GNNs. Differential privacy can be divided into common differential privacy and local differential privacy. The former adds noise such as Gaussian noise into attributes value or gradients which are in the form of continuous numbers. The latter perturbs the attributes in discrete forms on each user's terminals with a certain probability [25]. Hu et al. [9] disentangled the non-sensitive attributes into sensitive latent representation and non-sensitive latent representation under the orthogonal constraint. They only publish the non-sensitive latent representation rather than the non-sensitive attributes to defend against the attribute inference attack.

In summary, existing attribute inference attacks assume that attackers can access all clean sensitive attributes, without considering the situation that only limited sensitive attributes can be accessed. Furthermore, the limited sensitive attributes may be perturbed by differential privacy. To explore the privacy issues in this situation, we propose a novel attribute inference attack. Existing privacy-preserving methods cannot defend against this kind of attack efficiently, as they only have access to the noisy sensitive attributes. Hence, we present the adversarial training methods to defend against this kind of attack.

C. Fair and privacy-preserving GNNs

Although much progress has been made to meet both privacy and fairness requirements simultaneously in machine learning, the exploration of these issues in graph mining algorithms is fairly recent. Dai et al. [26] focused on group fairness and considered the privacy scenario that the sensitive attributes are limited and perturbed by LDP. In this scenario, they adopted a sensitive attribute estimator to provide sufficient sensitive attributes for adversarial training. Dai et al's work can defend against attribute inference attacks to some extent but cannot promote individual fairness. Zheng et al. [27] enforced the fairness constraint on the graph generative model to treat protected groups and unprotected groups equally in the generated graph. However, the generative model needs personal information to improve performance, which can be leveraged by the attacker to infer sensitive attributes. Zhang et al. [28] explored the relationship between edge privacy and individual fairness based on Lipschitz-based individual fairness. Their work considers Lipschitz-based individual fairness which possesses more limitations compared to the latest ranking-based individual fairness. Wang et al. [13] protected node privacy and promote ranking-based individual fairness on the perturbed node data. Specifically, they protected node privacy using LDP and aggregated the K-hop neighbor's attributes to average injected noise. However, their main privacy protection goals are different from ours. Besides, they promoted individual fairness using a surrogate loss which may loosely be related to the true loss.

Therefore, our work complements established research by considering the promotion of ranking-based individual fairness and defense of attribute inference attacks on limited and private sensitive attributes, simultaneously. Specifically, we propose novel fairness promotion and privacy-preserving methods, overcoming the limitation of existing methods. Furthermore, we achieve a fair and privacy-preserving graph neural network based on our proposed fair and privacy-preserving methods.

III. PROBLEM DEFINITION

In this section, we first present the notation used in this paper. Then, we introduce the preliminaries of GNNs and ranking-based individual fairness. Finally, we provide the problem formulation of fair and privacy-preserving GNNs on limited sensitive attributes and further formulate the problem on the private and limited sensitive attributes.

A. Notations

In this paper, bold uppercase characters (e.g., $\mathbf{S}$), bold lowercase characters (e.g., $\mathbf{s}$), and lowercase characters (e.g., s) denote matrices, vectors, and scalars, respectively. Let $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$ be an input graph, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the set of N nodes, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix of the graph and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the attribute matrix. Here n denotes the node number and d denotes the dimension of features. $\mathbf{Y} \in \{0,1\}^{n \times c}$ and $\widehat{\mathbf{Y}} \in \mathbb{R}^{n \times c}$ are the label matrix and prediction matrix for the node classification task, where c refers to the number of classes. The set $\mathcal{V}_L \subseteq \mathcal{V}$ denotes the

nodes with the label and the set $\mathcal{V}_S \subseteq \mathcal{V}$ refers to the nodes with the sensitive attributes. The similarity matrices from the input and output spaces are denoted as $\mathbf{S}_{\mathcal{G}}$ and $\mathbf{S}_{\widehat{\mathbf{Y}}}$, describing the pairwise similarity for individuals. Concretely, the input space refers to the node attributes, and the output space refers to the predictions of nodes.

B. Preliminaries of GNNs

To capture both attribute and structure information, GNNs learn a representation for each node using several layers. Each layer gets the output of the previous layer for each node and outputs an aggregation of adjacent neighbors' vectors by a non-linear transformation. Formally, given an input graph $(\mathcal{V}, \mathbf{A}, \mathbf{X})$, a GNN is built to learn a representation $\mathbf{h}_v$ of node $v \in \mathcal{V}$. The aggregation steps between the l-th and $(l+1)$-th layer are formulated as follows:

$$\mathbf{h}_{\mathcal{N}(v)}^{(l+1)} = \mathrm{AGGREGATE}^{(l)}\left(\left\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v)\right\}\right) \tag{1}$$

$$\mathbf{h}_v^{(l+1)} = \mathrm{ComBINE}^{(l)}\left(\mathbf{h}_v^{(l)}, \mathbf{h}_{\mathcal{N}(v)}^{(l+1)}\right) \tag{2}$$

where $\mathcal{N}(v)$ denotes the neighbors of node v, AGGREGATE $^{(l)}$ (.) denotes an aggregation function of layer l, and the combination function of layer l is denoted as COMBINE $^{(l)}$. The initial embeddings of u and v (i.e., $\mathbf{h}_u^{(0)}$ and $\mathbf{h}_v^{(0)}$) are the feature vector of $\mathbf{x_u}$ and $\mathbf{x_v}$.

C. Ranking-based Individual fairness

Ranking-based individual fairness is formulated as "*for each instance u_i, the two ranking lists of other instances (based on their distances to u_i) in the input space and outcome space should be as similar as possible*". Specifically, the ranking list of an instance u_i is obtained by ranking based on the similarity between u_i and other instances in descending order. The ranking lists for the $\mathbf{S}_{\mathcal{G}}$ and $\mathbf{S}_{\widehat{\mathbf{Y}}}$ are denoted as R_1 and R_2, respectively. The ranking-based individual fairness requires R_1 and R_2 of each instance to be consistent. For example, given an instance u_i and R_1 is $\{u_2,u_3,u_4\ \}$, if R_2 is also $\{u_2,u_3,u_4\ \}$, the prediction for u_i aligns with ranking-based individual fairness.

D. Fair and privacy-preserving GNNs on limited sensitive attributes

With the notation given in 3.1 and the ranking-baed individual fairness described in 3.3, we formulate the problem of training fair and privacy-preserving GNNs on the limited sensitive attributes as follows:

Problem 1. Given a graph $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X}), \mathcal{V}_S$ with sensitive attributes s, $\mathcal{V}_L$ with label matrix $\mathbf{Y}$, the prediction matrix $\widehat{\mathbf{Y}}$ for node classification tasks, the similarity matrices $\mathbf{S}_{\mathcal{G}}$ and $\mathbf{S}_{\widehat{\mathbf{Y}}}$ obtained from $\mathbf{X}$ and $\widehat{\mathbf{Y}}$, respectively, we aim to promote ranking-based individual fairness for each node, exclude users' sensitive information from the released graph embedding, and make $\widehat{\mathbf{Y}}$ close to $\mathbf{Y}$.

E. Fair and privacy-preserving GNNs on limiteds and private sensitive attribute

The LDP has been adopted to protect sensitive attributes. Concretely, the sensitive attributes are flipped according to the following distribution:

$$p\left(s' \mid s\right) = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+c-1}, & \text{if } s' = s \\ \frac{1}{e^{\epsilon}+c-1}, & \text{otherwise} \end{cases} \tag{3}$$

where s' and s denote the perturbed sensitive attributes and clean sensitive attributes, respectively, ϵ denotes the privacy budget, and c denotes the class number. In this paper, we denote $\rho = \frac{1}{e^{\epsilon}+c-1}$ as the probability of flipping sensitive attributes. Many efforts have been made to learn from noisy labels. Specifically, the forward correction loss (i.e., $\ell\left(s', \hat{s}'\right)$) [15] is equal to the original loss calculated on the clean labels (i.e., $\ell(s, \hat{s})$)). Here, s' denotes the perturbed sensitive attributes and $\hat{s}'$ denotes the perturbed prediction of the sensitive estimator. Hence, the attacker can train a classifier to predict true sensitive attributes, and we provide the formulation of this attack as follows:

Definition 1. (Attribute inference attack on private and partially observed sensitive attributes) Given graph $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$, all the node set $\mathcal{V}$, the node set $\mathcal{V}_S$ with sensitive attributes s', the flipping probability ρ, the adjacency matrix $\mathbf{A}$, the attribute inference attack on private and partially observed sensitive attributes is to infer the sensitive attributes for the node $u \in (\mathcal{V} - \mathcal{V}_S)$ by training a classifier using forward correction loss $\ell\left(s', \hat{s}'\right)$ where s' denotes the perturbed sensitive attributes and $\hat{s}'$ denotes the perturbed prediction.

To defend the attribute inference attack in Definition 1, the main challenge is how to purge the sensitive factor from the released graph embedding on the noisy sensitive attributes. Hence, we further formulate the problem of fair and privacy-preserving GNNs on private and limited sensitive attributes as follows:

Problem 2. Given graph $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$, $\mathcal{V}_S$ with sensitive attributes s, $\mathcal{V}_L$ with label matrix $\mathbf{Y}$, the prediction matrix $\widehat{\mathbf{Y}}$ for node classification tasks, the similarity matrices $\mathbf{S}_{\mathcal{G}}$ and $\mathbf{S}_{\widehat{\mathbf{Y}}}$ obtained from $\mathbf{X}$ and $\widehat{\mathbf{Y}}$, respectively, we aim to learn a GNN, satisfying three aims: 1) prediction aligns with the ranking-based individual fairness; 2) prediction maintains high accuracy (i.e., make $\widehat{\mathbf{Y}}$ close to $\mathbf{Y}$; 3) the released graph embedding can defend the attribute inference attack described in Definition 1.

IV. FPGNN FOR PARTIAL OBSERVED SENSITIVE ATTRIBUTES

In this section, we provide the details of FPGNN to learn fair and privacy-preserving GNNs and show the framework of FPGNN in Fig. 1. We aim to achieve the balance between fairness, privacy, and utility. Since it is difficult to determine the relationship between fairness, privacy, and utility, we set up three separate modules: an individual fairness promotion module (Module 1), a privacy-preserving module (Module 2), and a utility maximization module (Module 3). Specifically, **the individual promotion module** ensures two ranking lists

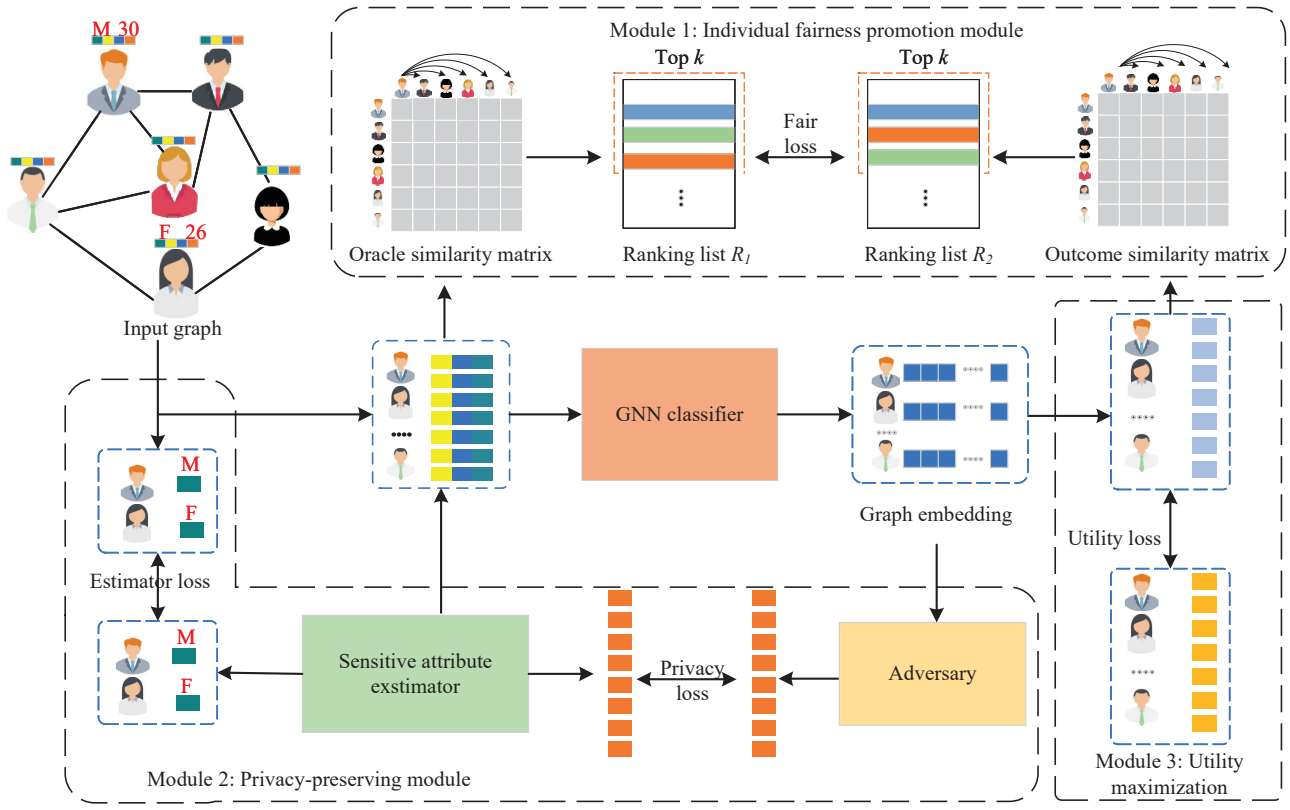


Fig. 1: The overall framework of FPGNN.

derived from $\mathbf{S}_{\mathcal{G}}$ and $\mathbf{S}_{\hat{\mathbf{Y}}}$ to be consistent for each instance; **the privacy-preserving module** purges sensitive factors from the released graph representation; **the utility maximization module** preserves the competitive accuracy of the backbone model for specific downstream tasks. We let them compete with each other to achieve balance. In the previous studies, the common sense is that fairness promotion and privacy protection may decrease the utility. Hence, the balance is achieved by improving fairness and privacy with a small reduction in utility.

A. The GNN classifier and utility maximization module

This module aims to preserve the high utility performance of the GNN classifier by learning a mapping function $f_{\mathcal{G}} : V_L \rightarrow y_L$ with parameter $\theta_{\mathcal{G}}$. The representation of node v for GNN with K layers is formulated as follows:

$$\mathbf{h}_v = f_{\mathcal{G}}^{(K)}\left(\mathbf{x}_v, \mathrm{A}, \theta_{\mathcal{G}}\right). \tag{4}$$

Since we focus on node classification tasks, we adopt a liner classification layer with parameter $\mathbf{W}$ to gain the predictions $\hat{\mathbf{y}}_{\mathbf{v}}$, which is formulated as follows:

$$\hat{\mathbf{y}}_{\mathbf{v}} = \sigma\left(\mathbf{h}_v\right) \tag{5}$$

where σ is the sigmoid function. The utility maximization module aims to maximize the utility of the backbone GNN. The prediction $\hat{\mathbf{y}}_{\mathbf{v}}$ is required to be close to the ground truth $\mathbf{y}_v$. The loss function based on the cross-entropy loss is formulated as follows:

$$\min_{\theta_{\mathcal{G}}} \mathcal{L}_U = -\sum_{v \in V_L} \mathbf{y}_{\mathbf{v}} \ln \hat{\mathbf{y}}_{\mathbf{v}}. \tag{6}$$

B. Fairness promotion module

Ranking-based individual fairness requires the ranking lists derived from the output space to be consistent with the ranking list derived from the input space. Since the input space is fixed, we only optimize the GNN parameters to generate appropriate predictions. One straight way to formulate the loss function is to quantify the difference between two ranking lists. We take the node u_i as an example. The i^{th} row of $\mathbf{S}_{\mathcal{G}}$ is denoted as L_1 which provides relevant information. R_2 is a ranking list from $\mathbf{S}_{\hat{\mathbf{Y}}}$, which provides the position information. The aim is to align instances on r^{th} position with r^{th} highest relevance value. However, the positions of instances in the two ranking lists may not be consistent. We first find the instance on r^{th} position in R_2 and then return the similarity value of this instance in $rel_i(r)$ from L_1. For NDCG@K, we calculate the NDCG@K across all nodes and combine them to derive the loss function. The formulation of the loss function is as follows:

$$\min_{\theta_{\mathcal{G}}} \mathcal{L}_F = \frac{1}{N} \sum_{i}^{I} \left(1 - \sum_{r=1}^{K} \frac{2^{rel_i(r)} - 1}{\log_2(r+1)}\right). \tag{7}$$

where N denotes a normalizing constant, and I is the number of nodes. However, the non-differentiability of the loss functions (7) prevents the optimization of GNN model parameters

using gradient-based methods. Get inspired by differentiable ranking metric [14], we adopt a rank indicator I_j^r to denote whether instance j is the r^{th} highest similarity value. $rel_i(r)$ can be written as follows:

$$rel_i\left(j_r\right) = \sum_{j=1}^{N} rel_i(j) I_j^r. \tag{8}$$

If instance j is the r^{th} highest similarity value, $I_j^r = 1$, otherwise, $I_j^r = 0$. To return which instance is the r^{th} highest similarity value, we adopt the argmax function formulated as follows:

$$\begin{cases} \underset{j' \in \{1,\ldots,N\}}{\operatorname{argmax}} \\ \forall l < r, I_{j'}^l = 0 \end{cases} S_{j'} = \underset{j' \in \{1,\ldots,N\}}{\operatorname{argmax}} S_{j'} \prod_{l=1}^{r-1} \left(1 - I_{j'}^r\right). \tag{9}$$

where $S_{j'}$ refers to the similarity value of j' from $\mathbf{S}_{\widehat{\mathbf{Y}}}$. Since argmax is not differentiable, the $I_{j'}^r$ calculated by parameterized softmax is formulated as follows:

$$I_j^{r,\alpha} = \frac{e^{\alpha S_j \Pi_{l=1}^{r-1}\left(1 - I_j^{r,\alpha}\right)}}{\sum_{j'} e^{\alpha S_{j'} \prod_{l=1}^{r-1}\left(1 - I_{j'}^{r,\alpha}\right)}}. \tag{10}$$

where α is a hyperparameter that controls the quality of the approximation. Hence, the smooth $rel_i(r)$ is formulated as follows:

$$rel_i(r) = \sum_{j=1}^{N} rel\left(j_r\right) I_j^{r,\alpha}. \tag{11}$$

We replace the $rel_i(r)$ in loss function (7) with a differentiable approximation of $\sum_{j=1}^{N} rel\left(j_r\right) I_j^{r,\alpha}$ and the fairness loss function is formulated as follows:

$$\min_{\theta_{\mathcal{G}}} \mathcal{L}_F = \frac{1}{N} \sum_{i}^{I} \left(1 - \sum_{r=1}^{K} \frac{2^{\sum_{j=1}^{N} \mathrm{rel}(jr) I_j^{r,\alpha}} - 1}{\log_2(k+1)}\right). \tag{12}$$

Theorem 1. Loss function (12) is differentiable regarding the parameter $\theta_{\mathcal{G}}$.

Proof. The prediction $\hat{\mathbf{y}}_{\mathbf{v}}$ is differentiable to the parameters $\theta_{\mathcal{G}}$. In the calculation of $I_j^{r,\alpha}$, each element $\mathbf{S}_{ij}$ in $\mathbf{S}_{\widehat{\mathbf{Y}}}$ can be obtained by calculating the cosine similarity of the $\hat{\mathbf{y}}_{\mathbf{v}}$. Hence, the loss function can be written as $f_I\left(f_c\left(f_s\left(f_{\mathcal{G}}\left(\mathbf{x}_v, \mathrm{A}, \theta_{\mathcal{G}}\right)\right)\right)\right)$ where $\mathcal{L}_F = f_I\left(I_j^{r,\alpha}\right)$, $I_j^{r,\alpha} = f_c\left(S_j\right)$, $S_j = f_s\left(\hat{\mathbf{y}}_{\mathbf{v}}\right)$, and $\hat{\mathbf{y}}_{\mathbf{v}} = f_{\mathcal{G}}\left(\mathbf{x}_v, \mathrm{A}, \theta_{\mathcal{G}}\right)$. Hence $\frac{\partial \mathcal{L}_F}{\partial \theta_{\mathcal{G}}} = \frac{\partial f_I}{\partial I_j^{r,\alpha}} \times \frac{\partial f_c}{\partial S_j} \times \frac{\partial f_s}{\partial \hat{\mathbf{y}}_{\mathbf{v}}} \times \frac{\partial f_G}{\partial \theta_{\mathcal{G}}}$ and $\mathcal{L}_F$ is differentiable to the parameters $\theta_{\mathcal{G}}$.

Theorem 2. If $\alpha \to +\infty$, the differential loss function (12) is equal to the loss function (17).

Proof. $\lim_{\alpha \to +\infty} I_j^{r,\alpha} = I_j^r$ has demonstrated in [14], hence loss function (12) is equal to loss function (7).

C. Privacy-preserving module

To defend against the attribute inference attack, the privacy-preserving module aims to purge the sensitive factor about the sensitive attributes from the released graph embedding. Hence, clean sensitive attributes are necessary in the training process. However, the sensitive attributes are limited in our setting. Since we assume that a trust third parity exists for learning graph embedding, we can employ a sensitive attribute estimator to predict accurate sensitive attributes without the leakage of sensitive attributes. Specifically, the sensitive estimator leverages the non-sensitive attributes, and the graph structure information to train a GNN classifier. It is still possible to predict the sensitive attributes accurately. The reason is that the message-passing of GNN captures two dependencies: 1) users and their neighbors tend to possess the same sensitive attributes; 2) non-sensitive attributes are naturally correlated with the sensitive attributes. The mapping function of the estimator is denoted as $f_{\boldsymbol{E}} : V_S \to S$ with the parameter θ_E, and the objective function is formulated as:

$$\min_{\theta_E} \mathcal{L}_E = -\sum_{v \in \mathcal{V}_S} s_{\boldsymbol{v}} \ln \hat{s}_v. \tag{13}$$

We denote the node set with estimated sensitive attributes as V_p. By combining the V_p with the existing node set V_S, we gain $V_c = V_S \cup V_p$. With V_c, we employ adversarial training to remove the information about the sensitive attribute from the released graph embedding. The adversarial learning-based framework consists of an attacker and an obfuscator. The attacker utilizes the graph structure and releases graph embedding $\mathbf{h}_v$ to predict the sensitive attributes, employing a separate GNN network f_A with parameters θ_A. The prediction of f_A is denoted as:

$$\hat{s} = f_A\left(f_{\mathcal{G}}\left(\mathbf{h}_v\right)\right). \tag{14}$$

The attack aims to make $\hat{s}$ close to the clean sensitive attributes s_v of node $v \in V_c$. In contrast, an obfuscator aims to make the prediction $\hat{s}$ far from the sensitive attributes. The adversarial game is formulated as a min-max problem:

$$\min_{\theta_{\mathcal{G}}} \max_{\theta_A} \mathcal{L}_P = -\sum_{v \in V_c} s_v \ln \hat{s}_v. \tag{15}$$

To decrease the inference accuracy, the obfuscator minimizes the loss function (15) by optimizing the parameters $\theta_{\mathcal{G}}$ of backbone GNN model $f_{\mathcal{G}}$.

Algorithm 1 Training Algorithm of FPGNN.

Input: Feature matrix $\mathbf{X}$, adjacency matrix $\mathbf{A}$, true labels $\mathbf{y}_{\mathbf{v}}$, the hyperparameters α, β, and γ, and the limited sensitive attributes s_v for node $v \in \mathcal{V}_S$.

Output: $\theta_{\mathcal{G}}$, θ_A, θ_E

1: Initialize the parameters θ_E of f_E by optimizing the loss function (13)
2: Initialize the parameters $\theta_{\mathcal{G}}$ of $f_{\mathcal{G}}$ by optimizing the loss function (6)
3: Calculate the similarity matrix $\mathbf{S}_{\mathcal{G}}$ for feature matrix $\mathbf{X}$.
4: **while** the stopping condition is not met **do**
5: Obtain the estimated sensitive attributes with f_E for the node
6: Obtain the graph embedding $\mathbf{h}_v$ according to (4)
7: Obtain the prediction $\hat{\mathbf{y}}_{\mathbf{v}}$ according to (5)
8: Calculate the similarity matrix $\mathbf{S}_{\widehat{\mathbf{Y}}}$ for prediction matrix $\widehat{\mathbf{Y}}$.
9: Compute $L_U, L_F, L_P, \mathcal{L}_E$ and $\mathcal{L}_{\text{total}}$ according to (6) (12) (13) (15) (16), respectively.
10: Optimize $\theta_{\mathcal{G}}$, θ_A, θ_E

11: **end while**
12: **return** $\theta_{\mathcal{G}}$, θ_A, θ_E

D. Overall training algorithm

We have introduced the loss functions of the fairness promotion module, the privacy-preserving module, and the utility maximization module, and the overall objective function is formulated as follows:

$$\min_{\theta_E,\theta_{\mathcal{G}}} \max_{\theta_A} \mathcal{L}_{\text{total}} = \mathcal{L}_U + \mathcal{L}_E + \beta\mathcal{L}_F + \gamma\mathcal{L}_P. \tag{16}$$

where β and γ are hyperparameters used to control the promotion strength of the terms. The first term in (16) ensures the preservation of high accuracy; the second term improves the accuracy of the sensitive attributes estimator; the third term enforces the efficiency of fairness promotion; and the final term guarantees the efficiency of privacy protection. To gain the objective function (16), we provide the overall training algorithm in Algorithm 1. To gain an accurate sensitive attribute estimator, we first initialize the parameters θ_E of f_E by optimizing the loss function (13) (line 1). Then, we initialize the parameters $\theta_{\mathcal{G}}$ by optimizing the loss function (6) (line 2), and the following operations are conducted based on this original model. To promote individual fairness and protect privacy, we calculate each item L_U, L_F, L_P and $\mathcal{L}_E$, and sequentially, we gain the overall objective function $\mathcal{L}_{\text{total}}$ (line 9) and optimize the whole module utilizing the Adam optimizer. Since optimizing the three parameters jointly is impossible, we update the three model parameters iteratively. Specifically, the two parameters are fixed when optimizing another parameter. The time complexity of FPGNN consists of three parts. The complexities of the privacy module and the utility module are both $O(n)$ and the complexity of the fairness promotion module is $O(Kn^2)$ where n refers to the number of nodes in the training set and K refers to the top-K instances considered. Therefore, the computational complexity of FPGNN is $O(Kn^2)$.

Algorithm 2 Training Algorithm of PL-FPGNN.

Input: Feature matrix $\mathbf{X}$, adjacency matrix A, true labels the observed sensitive attributes, the hyperparameters α, β, and γ, the noisy sensitive attributes s'_v for node $v \in \mathcal{V}_S$, and the probability of flipping sensitive attributes ρ.
Output: $\theta_{\mathcal{G}}$, θ_A, θ_E
1: Flip the sensitive attribute using a random response mechanism with probability ρ
2: Initialize the parameters θ_E of f_E by optimizing the loss function (17)
3: Initialize the parameters $\theta_{\mathcal{G}}$ of $f_{\mathcal{G}}$ by optimizing the loss function (6)
4: Calculate the similarity matrix $\mathbf{S}_{\mathcal{G}}$ for feature matrix $\mathbf{X}$.
5: **while** the stopping condition is not met **do**
6: Obtain the estimated sensitive attributes with f_E for the node
7: Obtain the graph embedding $\mathbf{h}_v$ according to (4)
8: Obtain the prediction $\hat{\mathbf{y}}_{\mathbf{v}}$ according to (5)
9: Calculate the similarity matrix $\mathbf{S}_{\widehat{\mathbf{Y}}}$ for prediction matrix $\widehat{\mathbf{Y}}$.
10: Compute $L_U, L_F, L_P, \mathcal{L}_E$ and $\mathcal{L}_{\text{total}}$ according to (6) (12) (18) (15) (16), respectively.
11: Optimize $\theta_{\mathcal{G}}$, θ_A, θ_E
12: **end while**
13: **return** $\theta_{\mathcal{G}}$, θ_A, θ_E

V. PL-FPGNN FOR PRIVATE AND PARTIAL OBSERVED SENSITIVE ATTRIBUTES

In this section, we present the details of PL-FPGNN which can promote individual fairness, defend against the attribute inference attack described in Definition 1, and preserve the competitive accuracy (Problem 2). The inference attack in Definition 1 considers the limited sensitive attributes perturbed by the LDP. By employing the approach for learning noisy labels, the attacker can infer the accurate sensitive attributes. However, without clean sensitive attributes, adversarial training cannot purge the sensitive factor efficiently. To defend against inference attacks in this situation, we extend FPGNN to PL-FPGNN. The framework of PL-FPGNN is shown in Fig. 2. It consists of three key modules: the individual fairness promotion module, the privacy-preserving module, and the utility maximization module. For the privacy-preserving module, we require the sensitive attribute estimator to predict the clean sensitive attributes. To learn about the noisy sensitive attributes, we first calculate $\hat{p}\left(\hat{s}' \mid \mathbf{x}\right)$ from $\hat{p}(\hat{s} \mid \mathbf{x})$:

$$\hat{p}\left(\hat{s}' \mid \mathbf{x}\right) = \sum_{s} p\left(\hat{s}' \mid \hat{s}\right) \cdot \hat{p}(\hat{s} \mid \mathbf{x}). \tag{17}$$

where $p\left(\hat{s}' \mid \hat{s}\right)$ is directly gained from (3) and $\hat{p}(\hat{s} \mid \mathbf{x})$ is the prediction of the sensitive attribute estimator. The forward correction loss function is formulated as:

$$\min_{\theta_E} \mathcal{L}_{E_p} = -\sum_{v \in \mathcal{V}_S} s'_v \ln \hat{p}\left(\hat{s}'_v \mid \mathbf{x}\right). \tag{18}$$

Unlike FPGNN, all the nodes in the set V_S are required to be predicted by a sensitive attribute estimator as the existing sensitive attribute in V_S are perturbed by LDP. Based on the clean data in V_c, the loss function (15) can also be used to purge the sensitive information from the released graph embedding. The PL-FPGNN algorithm is presented as Algorithm 2.

The difference between PL-FPGNN and FPGNN is that PL-FPGNN only accesses the noisy sensitive attributes denoted as s'_v and the probability ρ of flipping sensitive attributes. Using forward correction loss, the sensitive estimator f_E can predict the clean sensitive attributes, which are recorded in V_c. The calculations of the terms L_U, L_F, L_P and $\mathcal{L}_{\text{total}}$ are the same as FPGNN, while L_E is replaced by L_{E_p}. Besides, the parameters optimization is also the same as FPGNN. In PL-FPGNN, the extra operation such as perturbing the sensitive attributes, and calculation of forward correction loss both exhibit $O(n)$ time complexity. Therefore, the complexity of PL-FPGNN is $O(Kn^2)$, which is the same as that of FPGNN.

Theorem 3. Forward correction loss (18) on the noisy sensitive attributes is equal to the original loss (13) on the clean sensitive attributes.

Proof. The details for proving that forward correction loss is equal to the original loss can be found in [15].

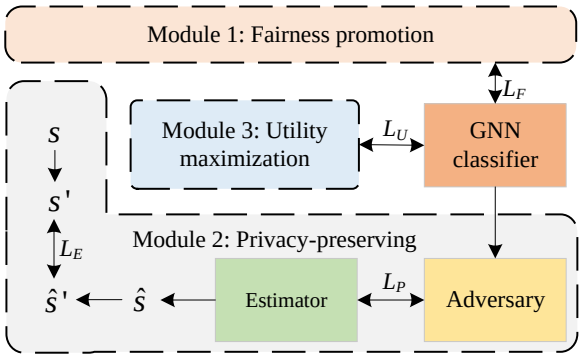


Fig. 2: The overall framework of PL-FPGNN.

TABLE I: Detailed statistics of datasets

Dataset	Nodes	Edges	Features	Sens.Attr	Label
German	1000	22242	27	Gender	Good/Bad Credit
Recidivism	18876	321308	18	Race	Bail/No bail
Credit	30000	1436858	13	Mattied	Default/No default

VI. EXPERIMENTS

In this section, we conduct empirical experiments to validate the effectiveness of our proposed methods. Specifically, we aim to answer the following research questions.

- **RQ1**: Can our proposed FPGNN promote individual fairness and defend against attribute inference attacks while maintaining high accuracy on limited sensitive attributes?
- **RQ2**: Can our proposed PL-FPGNN on private and limited sensitive attributes well balance the utility, privacy-preserving, and fairness promotion?
- **RQ3**: How do the fairness promotion module, privacy-preserving module, and utility maximization module affect our framework?
- **RQ4**: How do the hyper-parameters affect the performance of our framework?
- **RQ5**: How do the size of sensitive attributes affect the performance of our approaches?

A. Datasets

We conduct our experiments on three ethical datasets [9]: German credit, Recidivism, and Credit defaulter. All these datasets are public and available for access. The details of these datasets are listed in Table 1. Specifically, the German credit dataset consists of 1000 nodes. Each node represents a client in German credit banks and is described by 27 attributes. The sensitive attribute is set as the client's gender. Each edge denotes the similarity between users' accounts. The task is to classify the client into good or bad credit risks. Recidivism has 18876 nodes which represent the defendants released on bail in the U.S. state during 1990-2009. Each node contains 18 attributes and the race is a protected attribute. The edges denote the similarity between the defendant based on past criminal records and the demographics. The aim is to The edges denote the similarity between the defendant based on past criminal records and the demographics. The aim is to predict whether the defendants commit a crime again. The Credit defaulter dataset contains 30000 nodes that represent the individuals, and age is a sensitive attribute. The edges refer to the similarity between users, which is calculated using the spending and payment patterns of the users. The goal is to classify the users into default or not default using credit card payments.

B. Evaluation Metrics

Following [9], we adopt node classification accuracy to evaluate both node classification performance and privacy-preserving performance. For the first task, high accuracy is desired, while low accuracy is better for the second task. The reason is that the second task is to predict the sensitive attributes. Low accuracy indicates fewer accurate sensitive attributes are predicted by the attacker and the privacy of the private user is preserved. Following [13], ranking-based individual fairness is employed to measure the performance of fairness promotion. Essentially, this fairness is the ranking accuracy. For each node, the order of the ranking list R_1 is desired, ranking list R_2 derived from $\mathbf{S}_{\hat{\mathbf{Y}}}$ should be consistent with R_1. To measure the quality of R_2, we adopt two widely applied ranking metrics NDCG@K and ERR@K. The fairness metric is the average of the NDCG@K and ERR@K values across all nodes. Here, the K is set as 10 for the quantitative comparison.

C. Baselines and Comparisons

To demonstrate the efficiency of our methods in balancing utility, fairness promotion, and privacy preservation, we compare our approaches against state-of-the-art baselines: 1) fair and privacy-preserving GNN training algorithms; 2) fair GNN training algorithms; 3) privacy-preserving GNN training algorithms; 4) backbones GNN models such as GCN [30], GAT [31], and GIN [32]. The details of the baseline models are listed as follows:

- REDRESS [12]: REDRESS promotes ranking-based individual fairness which ensures the ranking lists for each node from input and output spaces are consistent.
- DP-GCN [9]: DP-GCN publishes no-sensitive latent representations to defend against the attribute inference attack.
- LPGNN [29]: This method is proposed to promote ranking-based individual fairness on the perturbed data.
- FairGNN [17]: FairGNN uses adversarial learning to make the released graph embedding decouple from the sensitive features.
- NT-FairGNN [26]: NT-FairGNN achieves group fairness in situations where limited sensitive attributes are perturbed by LDP.
- GUIDE [21]:This method promotes individual fairness that relies on the Lipschitz condition and further considers the equality of individual fairness between groups.
- Vanilla: Vanilla refers to the backbone model such as GCN, GAT, and GIN without any additional modifications and enhancements.

REDRESS, LPGNN, LPF-IFGNN, GUIDE, and backbone models such as GCN, GAT, and GIN need to complete

TABLE II: The comparisons of our proposed methods with the baselines on limited sensitive attributes

	Methods	German		Recidivism		Credit	
		ACC(U)	ACC(Privacy)	ACC(Utility)	ACC(Privacy)	ACC(Utility)	ACC(Privacy)
GCN	Vanilla	**70.20 ± 1.78**	92.48 ± 2.38	90.77 ± 0.96	58.45 ± 0.74	80.24 ± 0.22	97.27 ± 0.09
	REDRESS	66.20 ± 4.38	91.91 ± 1.43	89.25 ± 0.65	58.34 ± 0.57	78.99 ± 0.70	97.21 ± 0.65
	DP-GCN	69.56 ± 0.88	91.14 ± 1.05	85.40 ± 0.55	54.95 ± 0.63	79.29 ± 0.18	90.99 ± 0.15
	LPGNN	65.00 ± 4.68	87.14 ± 2.86	72.83 ± 1.86	**50.10 ± 0.54**	77.64 ± 0.40	90.49 ± 1.45
	LPF-IFGNN	65.48 ± 3.94	**84.95 ± 3.48**	70.63 ± 7.01	50.56 ± 0.74	75.69 ± 3.08	**89.49 ± 2.57**
	FairGNN	69.64 ± 0.87	91.52 ± 2.48	89.72 ± 0.30	58.93 ± 0.62	**80.30 ± 0.11**	95.88 ± 0.57
	GUIDE	62.36 ± 1.18	93.90 ± 1.57	**91.85 ± 0.36**	62.87 ± 0.96	66.24 ± 0.65	90.65 ± 0.30
	FPGNN	66.44 ± 1.46	90.67 ± 4.09	90.80 ± 0.17	57.50 ± 0.18	78.87 ± 0.51	94.46 ± 1.31
GAT	Vanilla	**67.68 ± 1.68**	86.29 ± 1.90	93.18 ± 0.23	58.80 ± 0.83	**80.46 ± 0.11**	97.03 ± 0.51
	REDRESS	62.90 ± 5.02	92.26 ± 5.00	82.09 ± 7.53	58.02 ± 0.60	76.92 ± 1.71	96.90 ± 0.22
	DP-GAT	61.12±12.14	88.86 ± 3.14	87.87 ± 0.30	54.90 ± 0.76	80.22 ± 0.17	90.98 ± 0.16
	LPGNN	67.68 ± 1.90	70.19 ± 4.28	66.51 ± 2.44	50.61 ± 1.28	77.49 ± 0.37	86.54 ± 3.36
	LPF-IFGNN	67.28 ± 2.56	**67.43 ± 7.62**	57.33 ± 4.42	**50.04 ± 0.48**	74.64 ± 3.99	**85.97 ± 5.75**
	FairGNN	67.08 ± 1.82	87.81 ± 1.86	89.64 ± 0.37	58.75 ± 0.56	80.11 ± 0.19	96.31 ± 0.25
	GUIDE	59.28 ± 1.38	79.71 ± 1.33	**95.05 ± 0.49**	61.18 ± 0.73	67.63 ± 1.64	90.00 ± 0.58
	FPGNN	65.68 ± 1.38	85.72 ± 3.52	92.16 ± 0.83	57.57 ± 0.53	78.23 ± 0.31	93.63 ± 1.34
GIN	Vanilla	63.16 ± 1.70	85.53 ± 1.91	**92.39 ± 0.44**	55.96 ± 0.78	79.00 ± 0.18	92.41 ± 0.44
	REDRESS	60.52 ± 3.46	84.76 ± 3.09	87.90 ± 1.25	54.79 ± 0.77	76.86 ± 1.99	92.16 ± 0.52
	DP-GIN	**68.00 ± 4.12**	83.71 ± 6.14	57.64 ± 11.11	**51.20 ± 1.51**	78.53 ± 0.18	90.95 ± 0.21
	LPGNN	61.00 ± 0.74	82.86 ± 2.38	87.71 ± 0.33	52.58 ± 0.57	78.53 ± 0.18	90.95 ± 0.21
	LPF-IFGNN	63.40 ± 1.60	83.52 ± 2.81	87.27 ± 0.91	53.13 ± 1.27	78.24 ± 0.19	91.00 ± 0.37
	FairGNN	63.00 ± 2.56	83.71 ± 2.71	93.29 ± 0.30	55.39 ± 0.26	**79.56 ± 0.15**	91.67 ± 0.28
	GUIDE	60.64 ± 1.96	83.24 ± 3.33	92.12 ± 0.18	56.38 ± 1.15	64.79 ± 0.44	**90.74 ± 0.19**
	FPGNN	62.36 ± 2.28	**79.43 ± 3.52**	90.79 ± 1.37	54.22 ± 0.66	78.25 ± 0.58	91.08 ± 0.22
		NDCG(fair)	ERR(fair)	NDCG(fair)	ERR(fair)	NDCG(fair)	ERR(fair)
GCN	Vanilla	42.38 ± 0.61	74.12 ± 1.03	33.20 ± 0.44	74.24 ± 0.33	53.43 ± 1.97	75.54 ± 1.26
	REDRESS	44.24 ± 1.46	75.21 ± 0.85	33.75 ± 1.02	74.48 ± 0.31	56.87 ± 4.44	77.60 ± 2.86
	DP-GCN	43.10 ± 0.75	74.27 ± 0.62	32.99 ± 0.21	73.90 ± 0.20	58.69 ± 0.68	79.51 ± 0.52
	LPGNN	41.06 ± 1.97	73.22 ± 1.74	31.60 ± 0.15	73.04 ± 0.80	37.15 ± 0.46	66.99 ± 0.84
	LPF-IFGNN	42.23 ± 1.04	75.79 ± 2.67	31.76 ± 0.21	73.84 ± 0.31	37.36 ± 0.81	67.41 ± 0.63
	FairGNN	38.06 ± 1.12	74.59 ± 1.47	30.81 ± 0.76	71.17 ± 2.99	31.48 ± 1.79	63.94 ± 3.49
	GUIDE	32.02 ± 1.79	69.43 ± 3.80	20.85 ± 0.41	67.82 ± 4.68	35.90 ± 0.97	71.95 ± 2.43
	FPGNN	**45.61 ± 1.61**	**76.20 ± 1.36**	**34.46 ± 0.94**	**74.60 ± 0.83**	**65.27 ± 1.20**	**83.71 ± 0.63**
GAT	Vanilla	43.51 ± 0.84	74.92 ± 1.35	33.26 ± 0.53	73.92 ± 0.28	55.87 ± 1.68	77.55 ± 1.31
	REDRESS	45.35 ± 0.61	74.74 ± 0.55	34.10 ± 0.96	74.54 ± 0.35	57.94 ± 4.26	78.61 ± 2.54
	DP-GAT	44.51 ± 1.91	74.66 ± 1.03	32.86 ± 0.26	73.96 ± 0.12	59.50 ± 0.38	79.04 ± 0.26
	LPGNN	43.25 ± 1.14	78.64 ± 3.26	32.58 ± 0.08	73.83 ± 0.48	43.23 ± 2.45	69.75 ± 1.22
	LPF-IFGNN	42.07 ± 2.91	75.53 ± 3.18	32.98 ± 0.32	74.21 ± 0.39	49.17 ± 1.99	74.08 ± 1.25
	FairGNN	35.08 ± 0.78	70.36 ± 3.96	31.29 ± 1.44	72.31 ± 5.26	31.82 ± 1.51	63.62 ± 4.71
	GUIDE	32.45 ± 1.45	72.52 ± 2.62	20.92 ± 1.20	68.08 ± 4.75	37.73 ± 1.02	79.11 ± 2.22
	FPGNN	**46.54 ± 0.77**	**75.80 ± 0.40**	**34.67 ± 0.88**	**74.56 ± 0.46**	**65.70 ± 2.73**	**82.96 ± 2.72**
GIN	Vanilla	40.45 ± 0.84	73.20 ± 0.68	32.44 ± 0.33	73.76 ± 0.18	41.82 ± 0.77	69.28 ± 0.38
	REDRESS	42.09 ± 0.68	74.68 ± 0.52	33.11 ± 0.40	74.07 ± 0.30	45.97 ± 1.32	71.66 ± 0.91
	DP-GIN	42.12 ± 1.17	74.36 ± 1.38	33.93 ± 1.51	**74.78 ± 1.23**	37.65 ± 0.88	67.21 ± 0.51
	LPGNN	40.45 ± 0.91	74.12 ± 0.69	31.80 ± 0.05	73.64 ± 0.04	37.65 ± 0.88	67.21 ± 0.51
	LPF-IFGNN	41.20 ± 1.09	74.66 ± 1.28	31.90 ± 0.09	73.65 ± 0.20	37.94 ± 1.17	67.28 ± 0.66
	FairGNN	37.25 ± 1.72	73.88 ± 2.29	32.01 ± 1.27	73.34 ± 2.30	31.03 ± 2.15	63.25 ± 3.71
	GUIDE	33.02 ± 1.04	73.35 ± 1.75	21.26 ± 1.00	70.07 ± 4.15	35.20 ± 1.59	69.38 ± 2.34
	FPGNN	**44.91 ± 0.65**	**75.83 ± 0.85**	**34.58 ± 0.50**	74.63 ± 0.24	**57.04 ± 1.40**	**78.35 ± 0.96**

sensitive attributes during training. Hence, we employ GCN as a sensitive attribute estimator. By treating non-sensitive attributes as inputs and sensitive attributes as outputs, we train a GCN classifier to predict missing sensitive attributes in the training data. Additionally, FairGNN and DP-GCN are specifically designed to handle scenarios where only limited sensitive attributes are available. These methods are well-suited for our experimental setup.

D. Implementation Details

For each dataset, we randomly split 30% of nodes for training, 20% of nodes for validation, and the remaining 50% of nodes for testing. To generate limited sensitive attributes, we randomly sample 30% of nodes from the training set, the sensitive attributes of the remaining nodes are unknown. Each experiment is conducted 5 times and the result is reported with average value and the standard deviation. For generalization purposes, we employ three widely applied GNNs: GCN [30], GAT [31], and GIN [32] as the backbone model. Each backbone consists of two layers with 32 hidden units. To facilitate the information flow, we employ SELU as the activation function between two layers. In the experimental setup, we set the default values for weight decay, learning rate, and dropout rate as 5e-6, 0.01, and 0.3, respectively. The initialization of the backbones requires 300 epochs, while the subsequent steps for fairness promotion and privacy protection encompass 15 epochs. To identify the optimal model, we vary the hyperparameters α, β, and γ among $\{0.5, 1, 10, 20, 50\}$, $\{0.1, 1, 2, 4, 6, 8, 10\}$, and $\{0.1, 1, 2, 4, 6, 8, 10\}$. The selection of the most suitable hyperparameter is based on the performance of the model on the validation set.

E. Performance of FPGNN

To answer RQ1, we compare FPGNN against state-of-the-art alternatives on balancing utility, privacy, and fairness. For generalization purposes, FPGNN and other baselines

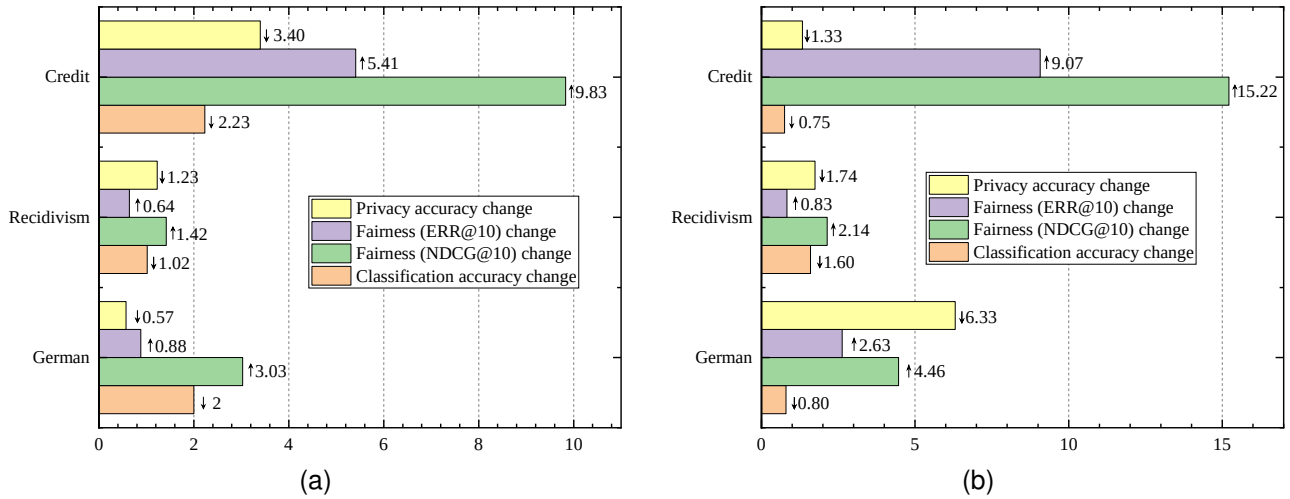


Fig. 3: The changes in privacy inference accuracy, fairness promotion (NDCG, ERR), and classification accuracy compared with GAT (a) and GIN (b).

are conducted under various GNN backbones. Quantitative results for the experiment are shown in Table 2. In this table, higher ACC(Utility) indicates better performance on model utility; higher NDCG(Fair) and ERR(Fair) represent better performance on ranking-based individual fairness, and lower ACC(Privacy) indicates better performance on privacy protection. We can make the following observations from Table 2:

From the perspective of model utility, our proposed framework FPGNN demonstrates competitive performance compared to other state-of-the-art baselines. Compared to the backbone model, the accuracy of FPGNN decreases by no more than 3%. Compared with individual fairness methods such as REDRESS and GUIDE, FPGNN has better performance in most cases. Besides, compared with the privacy-preserving GNN baselines, FPGNN also outperforms them in most cases. In some cases, the alternatives such as privacy-preserving GNNs or fair GNNs outperform the backbones. The reason can be conjectured that the privacy-preserving or fairness promotion methods play the role of regularization to prevent overfitting.

From the perspective of ranking-based individual fairness, our framework outperforms all baseline methods in all cases with different levels of improvement w.r.t the fairness evaluation metric NDCG@10 and ERR@10. This verifies the effectiveness of the individual fairness promotion of FPGNN. The reason is that our fair loss function is a good approximation to the true loss. Although we adopt a differentiable version for NDCG@10, ERR@10 is also improved as they both measure ranking accuracy in terms of similarity information and position information. GUIDE does not improve NDCG@10 and ERR@10 in some cases as it is not designed for rank-ing-based individual fairness.

From the perspective of privacy protection, our framework provides competitive performance. Compared with fair GNNs such as REDRESS, and GUIDE, our methods can significantly improve privacy-preserving performance. Besides, FPGNN also outperforms the backbones such as GCN, GIN, and GAT. Compared with the privacy-preserving baselines, FPGNN protects less privacy as the aim of FPGNN is not just to protect privacy.

From the perspective of balancing the model utility and individual fairness and privacy protection, Fig. 3 present the changes in privacy inference accuracy, fairness promotion (NDCG@10 and ERR@10), and classification accuracy compared with the backbone GNNs. We take FPGIN as an example, compared with GIN, FPGIN reduces the privacy inference accuracy by 3.4% on the Credit dataset, improving the fairness by 5.41% for ERR@10 and 9.83% for NDCG@10 with only a 2.23% of accuracy decrease. Based on such observations, we contend that FPGNN achieves a superior balance between utility, privacy, and individual fairness compared with other available alternatives. utility, privacy, and individual fairness compared with other available alternatives.

F. Performance of PL-FPGNN

To address RQ2, we conducted further experiments to evaluate the performance of PL-FPGNN in scenarios where sensitive attributes are limited and private. While accuracy and individual fairness are minimally impacted by the sensitive attributes, our focus is on defending against inference attacks. We compare PL-FPGNN with four baseline methods, including GNN, DPGNN, NT-FairGNN, and FPGNN on three backbones: GCN, GAT, and GIN. To provide a more intuitive representation of the level of privacy protection, we employ the flipping probabilities instead of the privacy budget, which is varied across {10%, 20%, 30%, 40%}. Each experiment was repeated five times to ensure statistical reliability. The outcomes of these experiments are shown in Fig. 4. In Fig. 4, we draw the following observations: The noise ratio of released sensitive attributes does not have a consistent impact on both fairness and accuracy. The PL-FPGNN exhibits lower inference accuracy compared to FPGNN across various noise ratios. The reason is that the sensitive attribute estimator

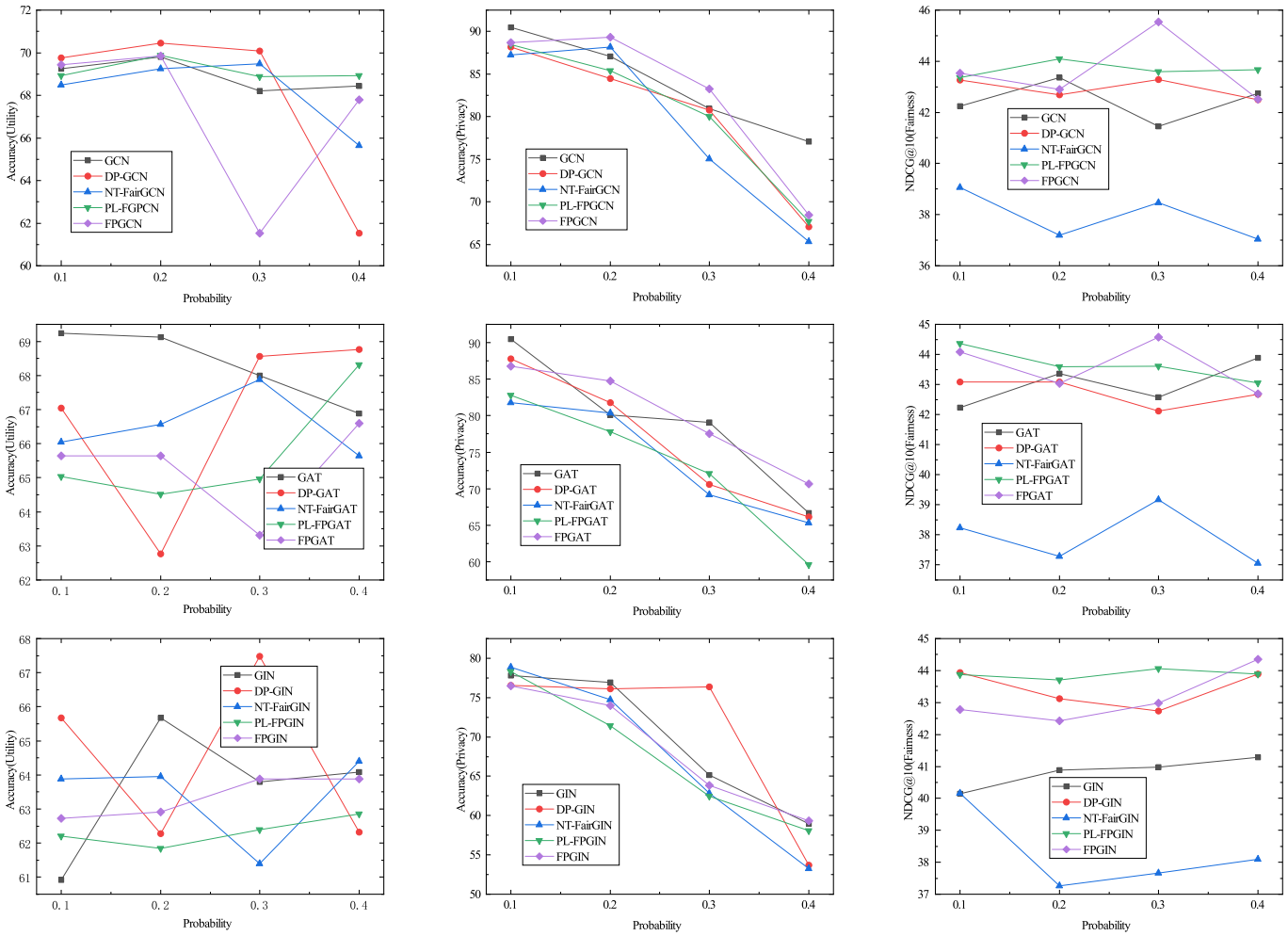


Fig. 4: The comparisons of PL-FPGNN with baselines on private and limited sensitive attributes.

employs the forward correction loss and can provide accurate predictions of sensitive attributes. The FPGNN and PL-FPGNN exhibit moderate accuracy and outperform the other baseline model in terms of individual fairness.

G. Ablation study

To answer RQ3, we discuss the functionality of the components of FPGNN in this section. The FPGNN consists of three modules: the individual fairness promotion module, the privacy-preserving module, and the utility maximization module. To explore the necessity of each module of FPGNN, we design three variants: FPGNN\F, FPGNN\P, and FPGNN\U. Specifically, FPGNN\F eliminates the individual fairness promotion module; FPGNN\P means FPGNN without the privacy-preserving module; FPGNN\U doesn't consist of the utility maximization module. We only show the result on the German dataset as we have similar observations on the other datasets. The experiment results are shown in Fig. 5 and we make the following observations. From the perspective of privacy protection, FPGAT\P has higher inference accuracy compared with other alternatives. This indicates the necessity of the privacy module for protecting privacy. For fairness promotion, FPGCN\F has less performance than the alternatives such as FPGNN\P, FPGNN\U, and FPGNN. In some cases, the privacy-preserving module may promote individual fairness. For instance, FPGCN\F and FPGIN\F have better fairness promotion than the corresponding backbones. However, in other cases, increasing privacy performance may decrease fairness promotion. Hence, a fairness module is necessary for stable fairness promotion. For utility maximization, FPGNN\U achieves lower node classification accuracy compared to other alternatives. It is worth noting that the accuracy of FPGAT\U is 43.5, which is significantly lower than the performance of the GAT backbone. Hence, it is necessary to preserve accuracy when promoting individual fairness and protecting privacy. Although, sometimes, the fairness terms or the privacy terms may play a role in regularization. It can prevent the over-fitting of backbone GNN models. However, in most cases, fairness promotion and privacy protection may decrease the accuracy as achieving these requirements may change the outputs of the original model.

PL-FPGNN has an estimator with a denoising mechanism that provides clean sensitive attributes for adversarial training. Since the other parts of PL-FPGNN are the same as FPGNN, PL-FPGNN only needs to be compared with FPGNN, aiming to verify the necessity of the denoising mechanism. FPGNN

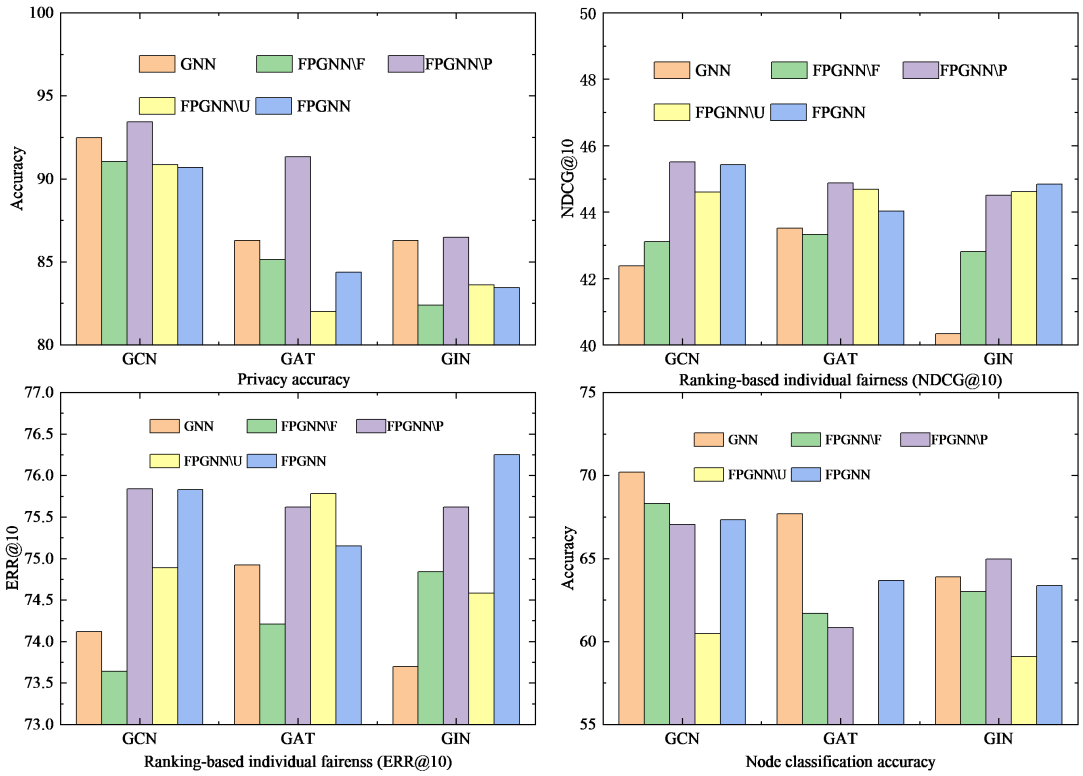


Fig. 5: The results of ablation studies for FPGNN.

and PL-FPGNN have been compared on the dataset with perturbed sensitive attributes in section 6.6. We find that PL-FPGNN has better performance on privacy preservation, which is benefit from the accurate sensitive attributes learned from the denoise mechanism. Therefore, the validity and necessity of the estimator are demonstrated.

H. Parameter-sensitive analysis

To answer RQ4, we analyze the parameter's effect on the performance of FPGNN. In our framework, the quality of the approximation is controlled by α, the fairness promotion is controlled by β, and the contribution of the privacy term is controlled by γ. Since α and β both affect the fairness promotion, we keep γ unchangeable and very α and β across $\{5,\ 10,\ 50,\ 100\}$ and $\{1,\ 5,\ 10,\ 15,\ 20\}$, respectively. The results of this experiment are presented in Fig. 6. To investigate the impact of γ, we conduct experiments by keeping α and β unchangeable and varying γ across $\{1,\ 5,\ 10,\ 15,\ 20\}$, and the results are presented in Fig. 7 (a). To explore the impacts of the size of sensitive attributes (RQ5), we vary the sizes of sensitive attributes as $\{0.1,\ 0.2,\ 0.4,\ 0.6,\ 0.8\}$, and keep the other hyperparameters unchangeable. We only show the results of FPGNN on German datasets and have similar observations on the other datasets and for PL-FPGNN. The results are shown in Fig. 7 (b).

From Fig. 6, we find that as α and β increase, the accuracy of attribute inference attacks on graph embeddings also increases. This reason is that, in graph data, neighboring nodes tend to possess similar predictions. The fairness promotion may provide a more similar prediction for the nodes and their neighbors, which is also incorporated into graph embedding and results in increasing the accuracy of the inference attack. This also highlights the necessity of striking a balance between fairness and privacy. For model utility, as the α and β increase to an appropriate value, the accuracy of the model reaches the maximum, which indicates that fairness loss serves as a regulation to prevent overfitting. However, as α and β continue to increase, the utility of the model decreases. When these parameters reach the maximum, the accuracy is lower than the accuracy of the baseline model. With the increase of α and β, fairness metrics such as NDCG@10 and ERR@10 initially increase until reaching a maximum. However, further increasing α and β leads to a fairness decline.

Fig. 7 (a) shows the effect of γ on the accuracy of attribute inference attacks. As γ increases, the accuracy of inferring sensitive attributes initially decreases and then reaches a plateau. Similarly, the utility of the model follows a similar pattern, initially decreasing and then stabilizing. This observation indicates that privacy protection is increased in a certain range of γ, which may decrease the model utility. Furthermore, the variations in γ do not significantly impact the fairness metrics NDCG@10 and ERR@10.

In Fig. 7 (b), the accuracy and fairness both improve slightly as the number of sensitive attributes gradually increases in the training set. This indicates that clean sensitive attributes may slightly lead to better performance on accuracy and fairness. The inference accuracy (i.e., the performance of privacy protection) improves with the increased size of sensitive attributes. Since the model has access to more sensitive attributes, it can infer more accurate sensitive attributes.

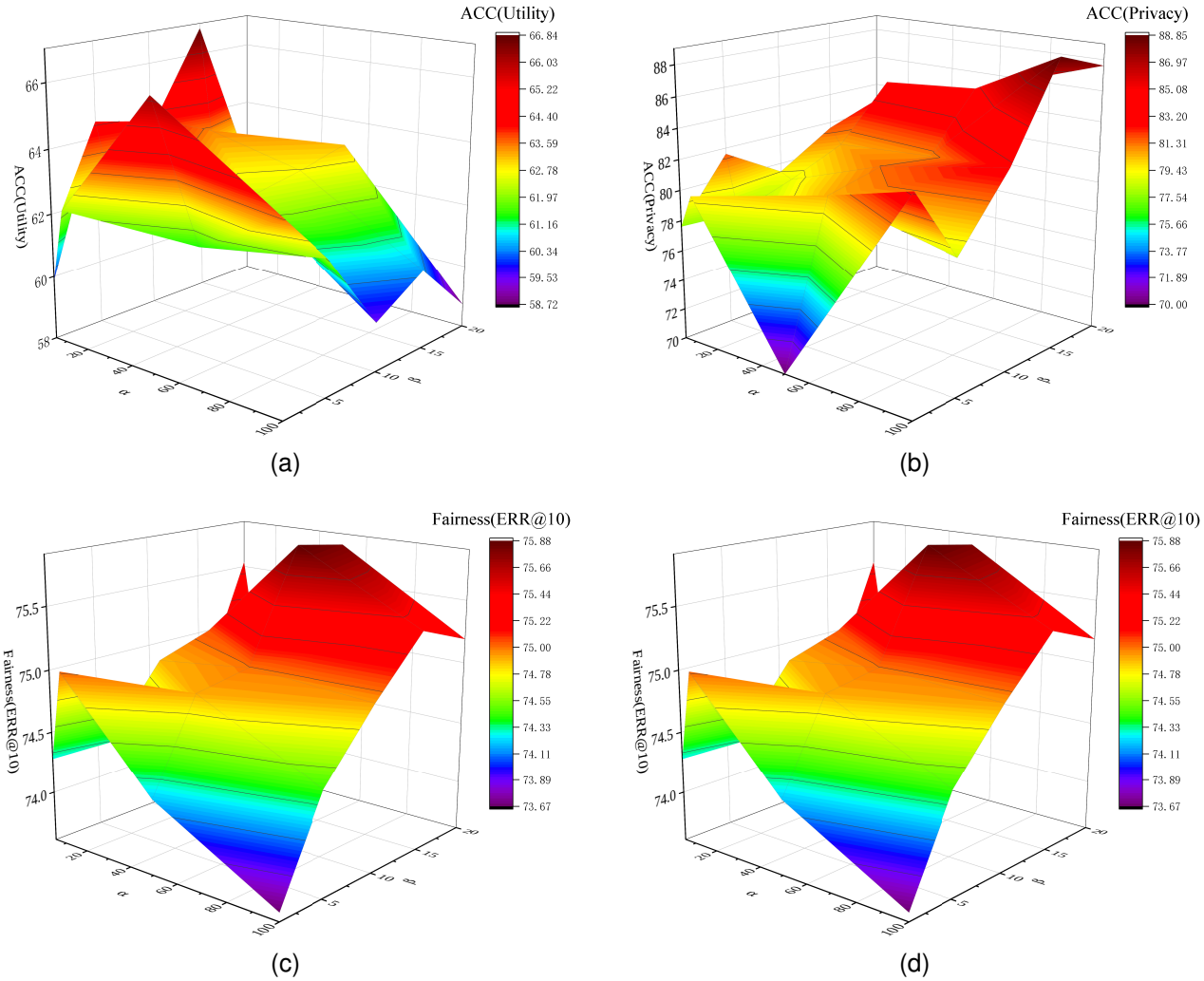


Fig. 6: Parameter sensitivity analysis of FPGIN (α and β).

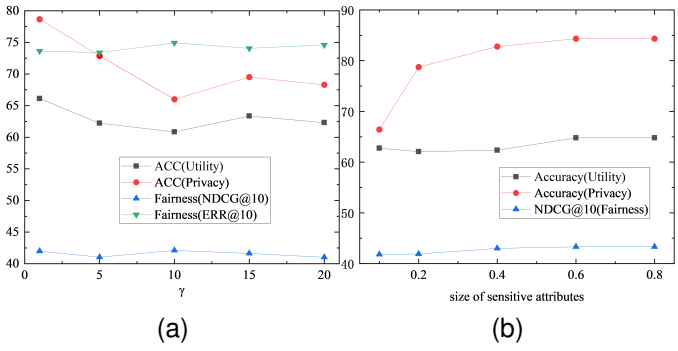


Fig. 7: (a) Parameter sensitivity analysis of FPGIN (γ) ; (b) Impact of the size of sensitive attributes to FPGNN

VII. Conclusion

In this paper, we propose a novel approach to tackle the challenge of promoting individual fairness and mitigating the leakage of sensitive attributes in graph embedding with limited and private sensitive attributes. We propose two GNN training algorithms, namely FPGNN and PL-FPGNN, which are designed to be individual fairness-aware and privacy-preserving GNN models. FPGNN specifically focuses on scenarios where the sensitive attributes are limited. It tackles the fairness issue by employing our proposed ranking-based individual fairness methods and removes sensitive information from graph embeddings through adversarial training. Additionally, FPGNN also considers the task of maximizing downstream task accuracy. We further consider the situation that the limited sensitive attributes are perturbed by LDP. To explore the privacy issues in this situation, we propose a novel attribute inference attack. Since the privacy-preserving module of FPGNN needs clean sensitive attributes, we propose PL-FPGNN to defend against this novel inference attack. Experimental evaluations conducted on three benchmark datasets demonstrate that both FPGNN and PL-FPGNN achieve a good balance between individual fairness promotion, privacy protection, and utility maximization. An interesting avenue for future research involves integrating other privacy-preserving GNNs and fair GNNs to offer diverse solutions that cater to different real-world requirements. Besides, trustworthy GNNs

consist of four ethical principles, namely respect for human autonomy, prevention of harm, fairness, and explainability. Our work primarily focuses on achieving fairness and preventing harm. In the future, we will investigate how to explain the bias and mitigate it based on the explanation.

REFERENCES

[1] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *NeurIPS*, Long Beach, CA, USA, 2017, pp. 1024-1034.

[2] D. Q. Nguyen, V. Tong, D. Q. Phung, D. Q. Phung, and D. Q. Nguyen, "Node Cooccurrence based Graph Neural Networks for Knowledge GraphLink Prediction," in *Proc. WSDM*, Tempe, AZ, USA, 2022, pp. 1589-1592.

[3] J. Wang, S. Zhang, Y. Xiao, and R. Song, "A Review on Graph Neural Network Methods in Financial Applications," 2022, *arXiv: 2111.15367.*

[4] T. Xiao, Z. Chen, D. Wang, and S. Wang, "Learning How to Propagate Messages in Graph Neural Networks," in *Proc. KDD*, Singapore, 2021, pp. 1894-1903.

[5] V. Shumovskaia, K. Fedyanin, I. Sukharev, D. Berestnev, and M. Panov, "Linking bank clients using graph neural networks powered by rich transactional data," *Int. J. Data Sci. Anal.*, vol. 12, no. 2, pp. 135-145, 2021, Doi: 10.1007/s41060-021-00247-3.

[6] J. Liu, Y. Lyu, X. Zhang, and S. Xie, "Subgroup Fairness in Graph-based Spam Detection," 2022, *arXiv: 2204.11164*.

[7] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, "Graph Neural Networks for Social Recommendation," in *Proc. WWW*, San Francisco, CA, USA, 2019, pp. 417-426.

[8] E. Dai, T. Zhao, H. Zhu., J. Xu, Z. Guo, H. Liu, J. Tang, and S. Wang, "A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability," 2022, *arXiv: 2204.08570* .

[9] H. Hu, L. Cheng, J. P. Vap, and M. Borowczak, "Learning Privacy-Preserving Graph Convolutional Network with Partially Observed Sensitive Attributes," in *Proc. WWW*, Lyon, France, pp. 3552-3561.

[10] Y. Dong, J. Ma, C. Chen, and J. Li, "Fairness in Graph Mining: A Survey," *IEEE Trans. Knowl. Data Eng.*, to be published, early access, Doi: 10.1109/TKDE.2023.3265598.

[11] J. Kang, J. He, R. Maciejewski, and H. Tong, "InFoRM: Individual Fairness on Graph Mining," in *Proc. KDD*, CA, USA, 2020, pp. 379-389.

[12] Y. Dong, J. Kang, H. Tong, and J. Li, "Individual Fairness for Graph Neural Networks: A Ranking based Approach," in *Proc. KDD*, Singapore, 2021, pp. 300-310.

[13] X. Wang, T. Gu, X. Bao, L, Chang, and L. Li, "Individual fairness for local private graph neural network," *Knowl. Based Syst.*, to be published, early access, Doi: 10.1016/j.knosys.2023.110490.

[14] T. Thonet, Y. G. Cinar, E. Gaussier, M. Li, and J. Renderse, "Listwise Learning to Rank Based on Approximate Rank Indicators," in *Proc. AAAI*, *Proc. IAAI*, *Proc. EAAI*, Virtual Event, 2022, pp. 8494-8502.

[15] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach," in *Proc. CVPR*, Honolulu, HI, USA, 2017, pp. 2233-2241.

[16] A. J. Bose, and W. L. Hamilton, "Compositional Fairness Constraints for Graph Embeddings," in *Proc. ICML*, Long Beach, CA, USA, 2019, pp. 715-724.

[17] E. Dai, and S. Wang, "Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information," in *Proc. WSDM*, Israel, 2021, pp. 680-688.

[18] Y. Wang, Y. Zhao, Y. Dong, H. Chen, J. Li, and T. Derr, "Improving Fairness in Graph Neural Networks via Mitigating Sensitive Attribute Leakage," in *Proc. KDD*, Washington, DC, USA, 2022, pp. 1938-1948.

[19] C. Agarwal, H. Lakkaraju, and M. Zitnik, "Towards a unified framework for fair and stable graph representation learning," in *Proc. UAI*, Virtual Event, 2021, pp. 2114-2124.

[20] J. Ma, R. Guo, M. Wan, L. Yang, A. Zhang, and J. Li, "Learning fair node representations with graph counterfactual fairness," in *Proc. WSDM*, Tempe, AZ, USA, 2022, pp. 695-703.

[21] W. Song, Y. Dong, N. Liu, and J. Li, "GUIDE: Group Equality Informed Individual Fairness in Graph Neural Networks," in *Proc. KDD*, Washington, DC, USA, 2022, pp. 1625-1634.

[22] P. Liao, H. Zhao, K. Xu, T. S. Jaakkola, G. J. Gordon, S. Jegelka, and R. Sala-khutdinov, "Graph Adversarial Networks: Protecting Information against Adversarial Attacks," 2020, *arXiv: 2009.13504.*

[23] K. Li, G. Luo, Y. Ye, W. Li, S. Ji, and Z. Cai, "Adversarial Privacy-Preserving Graph Embedding Against Inference Attack," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6904-6915, 2021, Doi: 10.1109/JIOT.2020.3036583.

[24] M. Jiang, T. Jung, R. Karl, and T. Zhao, "Federated Dynamic Graph Neural Networks with Secure Aggregation for Video-based Distributed Surveillance," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 56:1-56:23, Oct. 2022, doi: 10.1145/3501808.

[25] D. Xu, S. Yuan, X. Wu, and N. Phan, "DPNE: Differentially Private Network Embedding," in *Proc. PAKDD*, Melbourne, VIC, Australia, 2018, pp. 235-246.

[26] E. Dai, and S. Wang, "Learning Fair Graph Neural Networks with Limited and Private Sensitive Attribute Information," *IEEE Trans. Knowl. Data Eng.*, to be published, early access, Doi: 10.1109/TKDE.2022.3197554.

[27] L. Zheng, D. Zhou, H. Tong, J. Xu, Y. Zhu, and J. He, "FairGen: Towards Fair Graph Generation," 2023, *arXiv: 2303.17743*.

[28] H. Zhang, X. Yuan, Q. V. H. Nguyen, and S. Pan, "On the Interaction between Node Fairness and Edge Privacy in Graph Neural Networks," 2023, *arxiv:2301.12951.*

[29] S. Sajadmanesh, and D. Gatica-Perez, "Locally Private Graph Neural Networks," in *Proc. CCS*, Korea, 2021, pp. 2130-2145.

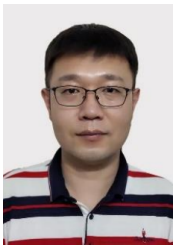
[30] T. N. Kipf, and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proc. ICLR-Poster Track*, Toulon, France, 2017.

[31] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," in *Proc. ICLR-Conference Track*, Vancouver, BC, Canada, 2018.

[32] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?," in *Proc. ICLR-Conference Track*, New Orleans, LA, USA, 2019.

Xuemin Wang received the B.E. degree from Nanjing University of Science and Technology, Nanjing, China, in 2019. He is currently pursuring the Ph.D. degree with the school of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research fields include graph neural network, machine learning fairness and privacy-preserving data mining.

Tianlong Gu received the Ph.D. degree from Zhejiang University, China, in 1996. From 1998 to 2002, he was a Post-Doctoral Fellow with the School of Electrical and Computer Engineering, Curtin University, Australia, and a Research Fellow with the School of Engineering, Murdoch University, Australia. He is currently a Professor and the director of the Engineering Research Center of Trustworthy AI, Ministry of Education, Jinan University, China. His research interests include formal methods, trustworthy artificial intelligence, artificial intelligence ethics, and data governance.



Xuguang Bao received the Ph.D. degree from Yunnan University, Kunming, China, in 2019. He is currently an associate professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His main research interests include spatial data mining, knowledge engineering, machine learning and human-computer interaction.

Liang Chang Liang Chang received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include information security, knowledge representation and reasoning, description logics, and the semantic Web.