FairDP-GNN: Graph Neural Network with Group Fairness and Differential Privacy

Fengrui Hao[1[0000-0002-5951-3789]], Shiyi Zhao[1[0009-0002-2290-5348]], Tianlong Gu[1]✉[0000-0002-1593-1292], Xuemin Wang[2[0000-0002-5041-8443]], Xiaoli Liu[1,3[0000-0001-6488-9732]], and Yuanfeng Liu[4[0009-0003-7856-2169]]

[1] Engineering Research Center of Trustworthy AI (Ministry of Education), Jinan University, Guangzhou 510632, Guangdong, China
gutianlong@jnu.edu.cn

[2] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, Guangxi China

[3] Graduate School of Engineering, Chiba University, Chiba, Japan

[4] Guangzhou Research Institute of Information Technology, Guangzhou 510075, Guangdong, China

Abstract. Graph Neural Networks (GNNs) have shown excellent performance in learning node representations of various types of graph-structured data. However, GNNs may capture sensitive information in the raw data, leading to privacy leakage of nodes or making unfair decisions for specific groups undermining group fairness. Differential Privacy (DP), as a widely accepted privacy-enhancing technology, is increasingly applied in machine learning to ensure privacy protection. Unfortunately, adding DP to GNNs has negative impacts on group fairness, resulting in discriminatory decisions from them and thus reducing user trust. To address the above problems, we propose a fair GNN training method based on DP called Fair Differential Privacy Graph Neural Network (FairDP-GNN). Specifically, we designed an Aggregation Differential Privacy Mechanism (ADPM) to impose the noise to the aggregation in the GNNs, thereby ensures the effectiveness and reasonableness of the noise. Furthermore, we developed a Group Fairness Promotion Strategy (GFPS) to improve the group fairness by counteracting or exploiting the effects of noise that may exacerbate or mitigate group bias. Experimental results on three real-world datasets show that our method can realize a good balance among model utility, privacy preservation, and fairness.

Keywords: Graph neural networks · Group fairness · Differential privacy.

1 Introduction

Nowadays, Graph Neural Networks (GNNs) have shown remarkable performance in processing and analyzing graph data. Their distinctive capabilities have made them a popular choice in fields such as social networks, recommendation systems, bioinformatics, and computer vision [11]. Renowned for their ability to learn and infer complex relationships among nodes and edges, GNNs provide powerful technical support for problem modeling. However, GNNs also pose a series of serious security challenges, particularly in the aspects of privacy protection and fair decision-making [3].

Firstly, the vulnerability of GNNs to privacy attacks (e.g., model stealing attacks) limits their practicality, especially in high-risk domains [12]. Secondly, for some specific groups, GNNs may not yield fair prediction outcomes for certain groups due to unbalanced datasets, or the models may fail to capture the features of minorities, leading to these groups being overlooked in predictions or decision-making processes [8]. To enhance the trustworthiness of GNNs, it is crucial to design a training method that ensures user privacy protection while also guaranteeing fair decision-making [9].

Differential privacy (DP) has become the standard for neural network training with strict protection of training data [11]. Currently, DP imposed in GNNs primarily involves node features, gradients and aggregation results [2]. Compared to the other two ways, adding noise to the aggregation results can usually be applied directly to graph-structured data and somewhat enhances the model's resistance to privacy attacks [6, 7]. However, this way can introduce or exacerbate bias of the GNN models, unfairly influencing the decision-making process and reiterating possible discrimination.

Additionally, substantial efforts have been invested in developing fair GNNs, with the goal of learning fair node representations that can be used to make accurate and fair predictions, thus ensuring that the same decisions are made for different groups of users (known as group fairness) [3, 10]. Unfortunately, current studies have mainly focused on improving the group fairness of model prediction results for GNNs in downstream tasks, without focusing on model leakage of user privacy.

In these regards, we aim to seek answers to the following question:

What are the side effects of DP when applied to protect the privacy of GNNs? How to balance group fairness and utility of GNNs while ensuring privacy?

Considering the issues mentioned previously, our research contributions are outlined as follows:

- We propose a novel training method for GNNs, FairDP-GNN, which promotes group fairness while maintaining accuracy and privacy, and it guarantees the privacy of the node features based on the noise added to the aggregation results.
- We designed an Aggregation Differential Privacy Mechanism (ADPM), which effectively controls the impact of noise on model performance through feature degradation and feature normalization. And we have theoretically proven that ADPM satisfies the requirements of the differential privacy.
- We developed a Group Fairness Promotion Strategy (GFPS), which focuses on the group fairness of model decision outcomes, while quantifying and estimating the impact of noise mechanisms on GNN model bias, and designed a fairness loss function to inhibit variance generation.

- To validate the effectiveness of our method, we conducted extensive experiments on three different graph datasets, which demonstrated that our model strikes a better balance in terms of utility, fairness, and privacy.

2 Preliminaries

Graph Neural Networks. In this paper, we focus on the GNN models in the node classification task. Let $G = (V, E, A, X, Y, S)$ be an input graph, where V and E denote the set of nodes and edges of the graph, respectively, $A \in \mathbb{R}^{|v|\times|v|}$ and $X \in \mathbb{R}^{|v|\times d}$ denote the adjacency matrix and feature matrix of the graph ($|v|$ nodes and d node features), respectively, $Y \in \{0,1\}^{|v|\times c}$denotes the labeling matrix of the nodes, where c denotes the number of classes, and $S \in \{0,1\}^{|v|}$ denotes the labeling of sensitive attributes of the nodes. In addition, the prediction result of the node classification task is denoted by $\hat{Y} \in \{0,1\}^{|v|\times c}$and the prediction result of node $v \in V$ is denoted by $\hat{Y}_v \in \{0,1\}$. GNNs utilize aggregation methods to learn node representations, which can be denoted as H, $h_v \in H$ represent the representations of each node.

Differential Privacy. DP [11] is a privacy notion that ensures an algorithm only outputs general information about its data without revealing the information of individual records.

Definition 1. $\epsilon-$Differential Privacy [11]. *A randomized algorithm $\mathcal{A}$ gives $\epsilon-$differential privacy ($\epsilon-$DP) for every set of outputs $S \subseteq Range\,(\mathcal{A})$, and for any neighboring datasets of G and G', if $\mathcal{A}$ satisfies*

$$Pr\left[\mathcal{A}\left(G\right) \in S\right] \leq exp(\epsilon) \times Pr\left[\mathcal{A}\left(G'\right) \in S\right] \tag{1}$$

where ϵ is privacy budget [8], which controls the level of privacy guarantee achieved by algorithm $\mathcal{A}$.

3 METHODOLOGY

Our method aims to promote group fairness of GNNs under DP, ensuring model privacy and group fairness while maximizing the utility of the backbone GNN models. The specific framework of which is shown in Fig. 1. It consists of two parts: Aggregation Differential Privacy Mechanism (ADPM), and Group Fairness Promotion Strategy (GFPS). Among them, ADPM aims to add noise to the aggregation result of each node. And GFPS aims to minimize discrimination in decision-making.

3.1 Aggregation Differential Privacy Mechanism

When formulating DP for GNNs, it's crucial to address a fundamental challenge: when perturbing features of a single node, the introduced noise might propagate through the message-passing mechanism, affecting the embedded representations

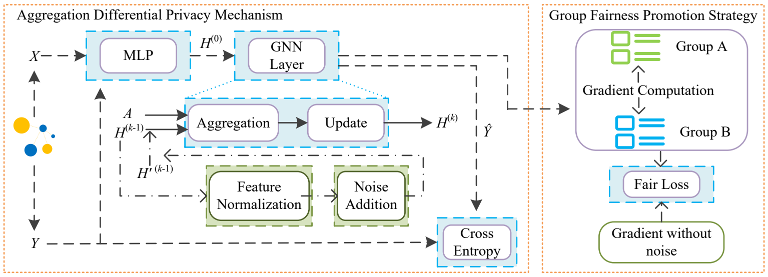

Fig. 1. The Framework of FairDP-GNN.

of neighboring nodes across subsequent layers [7]. Therefore, we develop an aggregation differential privacy mechanism.

Unlike the noise added to the original features of the nodes, we add noise to the aggregation result of each node in the stage of message aggregation. The specific implementation steps are as follows:

Step 1 Node feature aggregation: The node accepts messages from neighboring nodes and uses some aggregation mechanism (e.g. summing or averaging) to obtain the aggregated node feature h_v^k in the k-th layer:

$$h_v^k = AGG(MSG(h_u^{k-1}|u \in N_v), h_v^{k-1}), \tag{2}$$

where $MSG\left(\cdot\right)$ is message passing mechanism, $AGG\left(\cdot\right)$ is message aggregation mechanism, and N_v represents the set of neighboring nodes of node v.

Step 2 Noise Additions: Adding noise to each aggregated node feature is described by the formulae as follows:

$$h_v^k = h_v^k + N\left(\sigma^2 I\right), \forall v \in V, \tag{3}$$

where $N\left(\sigma^2 I\right)$ is Gaussian distributed noise with variance μ.

In the above steps, the noise size is determined by the sensitivity of aggregation function and privacy budget, and the sensitivity of aggregation function still has a direct impact on the degree of privacy protection and the balance of model performance. We need to further determine the size of the sensitivity of aggregation function to small changes in neighboring nodes.

To adjust the sensitivity of aggregation function, we adopt effective feature dimensionality reduction and feature normalization for node features from both feature dimension and feature scale perspectives, as follows:

Feature Dimensionality Reduction. Considering the non-identically distributed (non-IID) nature of the graph, we design a one-layer MLP to capture the complex nonlinear structure of the features. Specifically, we construct a simple fully connected neural network, which consists of an input layer, a hidden layer, and an output layer. The output of this hidden layer is the encoded feature representation. The construction can be described as follows:

Suppose the input feature vector is $x = [x_1, x_2, \ldots, x_n]$, which is first linearly varied through a fully connected layer consisting of a weight matrix W

and a compilation vector b: $z = Wx + b$, where z is the result of the linear transformation and represents the output of the hidden layer.

Next, z is processed through a nonlinear activation function f (e.g., ReLU, Sigmoid, Tanh, etc.) to obtain the final coded features: $h = f\left(Wx + b\right)$.

We learn a more advanced representation of the node features by minimizing the similarity between the feature representation h and the label Y:

$$\varTheta = \underset{\varTheta}{\operatorname{argmin}} \sum_{v \in V} \ell(h_v, Y_v). \tag{4}$$

With feature dimensionality reduction, we can keep the node feature dimensions within a fixed range, meaning that the node feature dimensions remain the same for small changes in neighboring datasets.

Feature Normalization. We take the feature representation extracted from the feature dimensionality reduction as input. Although the features are lowered in dimension, the difference in eigenvalues in each dimension is still high. Thus, before performing aggregation operations, we need to normalize the feature representation. For each feature dimension, we divide it by the standard deviation of that feature, scaling the variance of the features to 1, i.e.

$$h_v^k = h_v^k / \left\| h_v^k \right\|_2, \forall v \in V. \tag{5}$$

With feature normalization, we can control the dimensionality of individual node features to a fixed scale, which means that the maximum change in node feature information is within a certain range for neighboring datasets, and the impact on the message aggregation results is also controlled within a fixed range.

Through features dimensionality reduction and features normalization, we can ensure that the scale of the noise is within the appropriate range when adding noise to the model. This not only protects data privacy but also helps the model to maintain good performance under the influence of noise.

3.2 Group Fairness Promotion Strategy

The GFPS aims to enhance the group fairness of the models. We address group fairness improvement through two key aspects. Firstly, the possible decision bias of the GNN model itself is taken into account. Then, we analyze the possible bias of privacy noise for different groups when integrating noise into the GNNs.

For the first aspect, we start from the definition of group fairness and introduce a penalty term based on group differences into the loss function. Specifically, we assign different weights or penalty terms to each group to emphasize the fairness between groups, taking sensitive groups A and B as an example, as expressed below:

$$\tau_A = \frac{\sum_{s_i \in A} g_i}{|A|}, \tau_B = \frac{\sum_{s_i \in B} g_i}{|B|}, \tag{6}$$

where τ_A and τ_B are the average gradients of groups A and B respectively, $|A|$ and $|B|$ are the number of nodes in groups A and B respectively, s_i is the sensitive label of node i, and g_i is the gradient of node i. Then for group A and B, we have:

$$\mathcal{L}_1 = \alpha\tau_A + \beta\tau_B. \tag{7}$$

For the other aspect, the computed gradients may have an additional bias in the presence of noise, which may affect the direction of updating the model parameters. We evaluate the privacy noise by comparing the gradients computed from the aggregation results with the addition of privacy noise $\bar{g}_i$ with g_i without added noise to assess the change in impact values between aggregations.

The average gradients after noise addition are $\widetilde{\tau}_A, \widetilde{\tau}_B$ respectively, then the difference between the gradients of the two sensitive groups is:

$$D_{A,B} = Diff\{\left|\tau_A - \widetilde{\tau}_A\right|, \left|\tau_B - \widetilde{\tau}_B\right|\}, \tag{8}$$

To minimize the group bias due to noise, we propose the following loss formula for optimizing group fairness:

$$\mathcal{L}_2 = argminD_{A,B}. \tag{9}$$

With the introduction of this fairness loss term, we are able to regulate the impact of the privacy-preserving mechanism on the fairness of the groups. Combining the above analyses, our model loss function is designed as:

$$\mathcal{L} = \mathcal{L}_P + \gamma\mathcal{L}_1 + \rho\mathcal{L}_2, \tag{10}$$

where $\mathcal{L}_P$ is the downstream task prediction loss, and γ, ρ are hyperparameters.

With this loss constraint, it helps to improve the overall performance and group fairness of the model, allowing the model to better treat different groups while privacy preserving and ensuring that the decision is relatively fair in terms of its impact on each group. The trade-off helps to avoid certain groups from being treated unfairly due to the application of the privacy protection mechanism, thus improving the overall utility and group fairness of the model.

4 Privacy analysis

In this section, we prove that FairDP-GNN is qualified for differential privacy by theoretical analysis.

Theorem 1. *ADPM can guarantee ϵ-DP for edge-level ϵ-Differential Privacy.*

Proof. For the aggregation result $\widetilde{\mathrm{X}^{(k+1)}}$ after noise addition and the aggregation result $\widetilde{\mathrm{X}'^{(k+1)}} = x^*$ after noise addition on neighboring data, there are:

$$\frac{\Pr\left[\widetilde{\mathrm{X}^{(k+1)}} = x^*\right]}{\Pr\left[\widetilde{\mathrm{X}'^{(k+1)}} = x^*\right]} = \frac{\Pr\left[\mathrm{X}^{(k+1)} + \mathrm{N}(\sigma^2\mathrm{I}) = x^*\right]}{\Pr\left[\mathrm{X}'^{(k+1)} + \mathrm{N}(\sigma^2\mathrm{I}) = x^*\right]}$$

$$= \frac{e^{-\frac{\left[x^* - \mathrm{X}^{(k+1)}\right]^2}{2\sigma^2}}}{e^{-\frac{\left[x^* - \mathrm{X}'^{(k+1)}\right]^2}{2\sigma^2}}} = \frac{e^{-\frac{\left[x^* - \mathrm{X}^{(k+1)}\right]^2}{2\sigma^2}}}{e^{-\frac{\left[x^* - \mathrm{X}^{(k+1)} + \Delta X_v^{k+1}\right]^2}{2\sigma^2}}}.$$

where ΔX_v^{k+1} is the edge-level sensitivity of the aggregation function. This leads to the following:

$$\begin{aligned}
\left| \ln \frac{e^{-\frac{\left[x^* - \mathrm{X}^{(k+1)}\right]^2}{2\sigma^2}}}{e^{-\frac{\left[x^* - \mathrm{X}^{(k+1)} + \Delta X_v^{k+1}\right]^2}{2\sigma^2}}} \right| &= \left| \ln e^{-\frac{1}{2\sigma^2}\left[\left[x^* - \mathrm{X}^{(k+1)}\right]^2 - \left[x^* - \mathrm{X}^{(k+1)} + \Delta X_v^{k+1}\right]^2\right]} \right| \\
&= \left| -\frac{1}{2\sigma^2}\left[\left[x^* - \mathrm{X}^{(k+1)}\right]^2 - \left[x^* - \mathrm{X}^{(k+1)} + \Delta X_v^{k+1}\right]^2\right] \right| \\
&= \left| -\frac{1}{2\sigma^2}\left[2\left[x^* - \mathrm{X}^{(k+1)}\right]\Delta X^{k+1} + \left(\Delta X_v^{k+1}\right)^2\right] \right| \leq \varepsilon.
\end{aligned}$$

5 Experiment evaluation

5.1 Experimental settings

Experimental Setup Our proposed method is validated on three publicly available datasets [13]: Recidivism, Pokec-z and Pokec-n. To demonstrate that our method achieves a good balance between the utility and group fairness of private data, we compare our method with state-of-the-art fair classification methods, which can be categorized into two categories:(1) fair node classification methods: BIND [4], NIFTY [1] and FairGKD [13]; (2) private node classification method: NodeDP [2]. Our proposed FairDP-GNN is a plug-and-play method that can be easily applied to any popular GNN architecture, and we use Graph Convolutional Networks (GCN), GraphSAGE, and GIN as the backbone of the node classification task [4, 13].

Parameter Setting We randomly divide the dataset into training, validation and test sets with 50%, 25%, and 25% respectively. In the privacy setting, we take values of [0.01, 0.1, 1, 4] for the privacy budget respectively. Furthermore, we empirically measure the privacy guarantees of FairDP-GNN by conducting the latest model stealing attack for GNNs [5] as the most relevant adapted privacy attack.

Metrics Setting To assess model utility, we employ classification accuracy (ACC) and Area Under the Curve (AUC). For evaluating group fairness, we focus on two specific metrics [10]: Statistical Parity (ΔSP) and Equality of Opportunity (ΔEO). For privacy, we adopt the fidelity metric [5] to measure whether the surrogate model can generate classification results similar to the target model on the test set.

5.2 Results and analysis

How does FairDP-GNN balance GNN model utility and group fairness compared to the state-of-the-art GNNs? Tab. 1 shows the accuracy and group fairness metrics of FairDP-GNN and baselines on the three datasets. The experimental results indicate our method achieves a balance between accuracy and group fairness, outperforming the baselines in terms of both utility and group fairness metrics. This suggests that our method effectively enhances model performance while ensuring group fairness in graph-based classification tasks. Specifically, our method consistently achieves higher ACC, and AUC compared to the baseline methods (BIND, NIFTY, NodeDP, FairGKD). Moreover, in terms of group fairness metrics, our method demonstrates notable reductions in the disparities measured by ΔSP and ΔEO compared to the baselines.

Table 1. Model utility and bias of node classification. The best and runner-up results are bolded and underlined, respectively. ↑ indicates that larger values reflect better performance, while ↓ represents the opposite.

Backbone	Baseline	Redicivism				Pokec-z				Pokec-n			
		ACC(↑)	AUC(↑)	ΔSP(↓)	ΔEO(↓)	ACC(↑)	AUC(↑)	ΔSP(↓)	ΔEO(↓)	ACC(↑)	AUC(↑)	ΔSP(↓)	ΔEO(↓)
GCN	BIND	**90.57±0.22**	**93.61±0.23**	8.24±0.25	2.32±0.35	61.45±0.19	65.79±0.13	9.06±0.4	6.13±0.56	58.54±0.12	62.9±0.1	4.42±0.45	**1.52±0.41**
	NIFTY	86.26±0.37	89.21±0.23	6.83±0.28	2.67±0.55	63.1±0.73	68.28±0.7	5.62±2.36	5.14±2.66	59.38±0.87	63.42±0.88	3.38±2.85	2.52±2.07
	NodeDP	84.4±1.4	81.09±2.03	9.25±1.73	8.03±2.86	54.09±2.38	53.84±2.33	4.15±3.89	4.56±3.87	54.99±1.54	53.16±1.44	5.76±3.98	5.37±4.21
	FairGKD	89.24±0.46	93.3±0.34	6.85±0.25	4.39±0.73	62.37±0.38	66.75±0.45	8.49±1.63	7.05±1.95	59.0±0.46	63.28±0.45	5.98±1.55	1.54±1.25
	OURS	86.27±1.34	89.9±0.83	**6.4±0.95**	**1.79±0.86**	**64.44±3.35**	**69.56±4.66**	**2.35±1.33**	**1.79±1.26**	**67.27±3.07**	**73.21±3.82**	**1.92±1.41**	3.53±1.61
GIN	BIND	90.58±0.3	93.76±0.19	8.11±0.26	1.75±0.32	63.66±0.74	68.63±0.62	7.34±2.2	6.64±1.8	59.67±0.8	63.66±0.97	5.64±2.84	**1.83±1.7**
	NIFTY	75.74±3.38	78.62±3.61	7.19±1.83	2.7±1.84	62.03±1.29	67.75±0.72	4.12±2.61	3.77±2.54	57.91±2.14	63.12±2.14	3.72±2.67	3.6±2.43
	NodeDP	84.71±2.14	81.52±3.09	10.62±2	8.86±2.82	56.72±1.69	56.45±1.86	6.09±4.2	4.69±3.33	57.94±2.23	55.44±2.41	5.29±3.44	5.17±4.53
	FairGKD	**92.15±0.44**	**94.6±0.52**	6.76±0.45	2.06±0.77	56.71±3.42	59.26± 4.4	2.96± 2.38	4.13± 1.84	57.84± 0.67	61.47±0.7	4.05±2.45	3.83±2.69
	OURS	86.84±1.32	91.33±0.67	**6.06±0.65**	**1.55±1.07**	**67.21±2.37**	**72.73±2.57**	**2.72±1.53**	**2.74±1.69**	**71.52±3.61**	**78.3±4.74**	**1.83±1.67**	3.16±1.73
GraphSAGE	BIND	90.61±0.23	93.66±0.22	8.17±0.22	1.74±0.38	63.95±0.83	68.76±0.91	7.33±1.82	6.3±2.44	59.66±0.55	63.75±0.53	5.56±2.41	3±2.47
	NIFTY	88.15±0.92	92.03±0.82	10.42±1.14	3.6±1.41	65.25±0.44	70.04±0.44	3.5±1.94	3.86±2.66	60.77±1.86	64.21±0.73	2.72±0.89	2.01±1.49
	NodeDP	86.84±1.5	84.56±1.82	9.75±0.99	7.37±1.77	53.55±1.43	52.82±1.63	3.45±2.34	3.52±2.04	54.71±1.99	51.82±1.7	1.58±1.33	2.08±1.92
	FairGKD	**96.83±0.38**	**98.5±0.43**	8.06±0.62	2.44±0.79	62.8±0.63	67.01±1.15	5.41 ± 1.81	5.0 ± 1.7	59.72±0.9	62.91±1.06	3.44±1.31	**1.59±1.14**
	OURS	91.44±1.12	95.96±0.51	**7.12±0.57**	**1.42±0.81**	**67.44±3.72**	**73.9±4.28**	**2.59±1.37**	**2.44±1.59**	**69.95±3.22**	**75.87±3.62**	**1.55±1.11**	2.75±1.14

How does FairDP-GNN perform on privacy-preserving task compared to the state-of-the-art GNNs? We empirically measure the privacy guarantees of FairDP-GNN by conducting the latest model stealing attack for GNNs [5] as the most relevant adapted privacy attack. Fig. 2 summarizes the fidelity performance in the model stealing attack for the respective surrogate models of FairDP-GNN and baselines. With different datasets and backbones, our method can maintain lower fidelity score under different privacy budgets, i.e., the target model is stolen to a lesser degree.

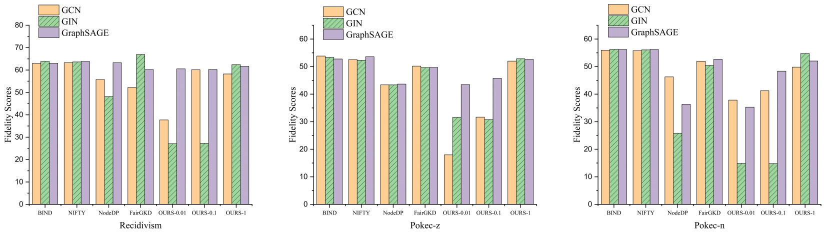

Fig. 2. The Fidelity Scores(↓) of Model Stealing Attacks.

How does privacy budget affect the performance of FairDP-GNN? The hyperparameter epsilon controls the amount of noise in practice. To explore the impact of epsilon, we vary it between 0.1, 0.5, 1, 2, and 4, and we give

Table 2. Impact of ϵ-Values on Accuracy and Group Fairness.

Epsilon(ϵ)	Recidivism			Pokec-z			Pokec-n		
	ACC(↑)	ΔSP(↓)	ΔEO(↓)	ACC(↑)	ΔSP(↓)	ΔEO(↓)	ACC(↑)	ΔSP(↓)	ΔEO(↓)
0.1	61.26±2.68	2.83±1.94	1.47±0.99	54.13±0.85	3.32±2.3	2.35±1.73	51.66±0.55	2.01±1.12	2.13±1.64
0.5	78.69±2.44	5.52±0.78	1.47±1.4	59.03±1.89	1.64±1.34	1.65±1.01	58.44±1.55	2.07±1.55	3.31±3.99
1	86.84±1.32	6.06±0.65	1.55±1.07	67.21±2.37	2.72±1.53	2.74±1.69	71.52±3.61	1.83±1.67	3.16±1.73
2	90.45±0.93	7.58±0.6	0.76±0.89	69.46±3.17	1.96±1.19	2.27±1.2	69.72±1.13	1.19±0.35	3.64±1.48
4	92.41±0.18	8.26±0.77	1.06±0.58	75.15±3.06	3.38±1.44	2.77±1.84	75.91±0.58	1.48±0.97	2.42±1.41

the utility and group fairness performance in Tab. 2. As can be seen in Tab. 2, a larger epsilon fetches good model performance, but in combination with the previous attack experiments, we can speculate that the privacy protection is insufficient when the epsilon is taken to 4. However, an appropriate epsilon can protect privacy while maintaining utility, such as epsilon = 1.

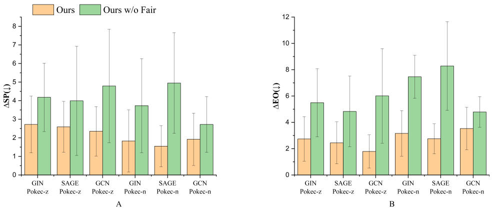

Fig. 3. Model bias(↓) after removing GFPS.

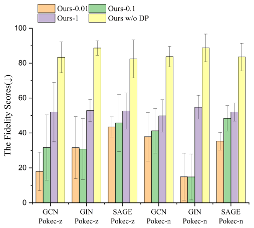

Fig. 4. Results of different privacy guarantee.

Ablation Experiments. We conduct ablation studies to fully understand the role of FairDP-GNN in mitigating discrimination and protecting privacy. Specifically, we denote Ours w/o Fair as the result of not enabling the GFPS, and Ours w/o DP as the result of removing the ADPM. The utility and bias of these variables are shown in Fig. 3 and Fig. 4. In Fig. 3, We can clearly see that after removing GFPS, Ours w/o Fair performs worse than Ours, which validates the effectiveness of the group fairness enhancement strateg in FairDP-GNN for fair classification. In addition, we observe in Fig. 4 that Ours w/o DP does not perform as well as Ours-0.01, Ours-0.1, and Ours-1 in the face of model steal attack, which suggests that the proposed differential privacy model is effective in protecting privacy to a large extent.

6 Conclusion

In this paper, we propose a trustworthy GNN training method, FairDP-GNN, focusing on the group fairness, privacy, and utility of the GNNs. Firstly, we pay attention to the limitations of implementing DP in graph structures, and design Aggregation Differential Privacy Mechanism to maintain model privacy while mitigating utility loss. Subsequently, we develop Group Fairness Promotion Strategy to mitigate model discrimination caused by both model training and noise addition. Finally, the experimental results demonstrate the effectiveness of FairDP-GNN.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant No. U22A2099 and Grant No. 62336003).

References

1. Agarwal, C., Lakkaraju, H., Zitnik, M.: Towards a unified framework for fair and stable graph representation learning. In: Proceedings of Conference on Uncertainty in Artificial Intelligence, UAI. pp. 2114–2124. PMLR (2021)
2. Ayle, M., Schuchardt, J., Gosch, L., Zügner, D., Günnemann, S.: Training differentially private graph neural networks with random walk sampling. In: Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS (2022)
3. Chen, A., Rossi, R.A., Park, N., Trivedi, P., Wang, Y., Yu, T., Kim, S., Dernoncourt, F., Ahmed, N.K.: Fairness-aware graph neural networks: A survey. ACM Transactions on Knowledge Discovery from Data **18**(6), 1–23 (2024)
4. Dong, Y., Wang, S., Ma, J., Liu, N., Li, J.: Interpreting unfairness in graph neural networks via training node attribution. Proceedings of the AAAI Conference on Artificial Intelligence **37**(6), 7441–7449 (2023)
5. He, X., Jia, J., Backes, M., Gong, N.Z., Zhang, Y.: Stealing links from graph neural networks. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2669–2686 (2021)
6. Sajadmanesh, S., Gatica-Perez, D.: Progap: Progressive graph neural networks with differential privacy guarantees. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. pp. 596–605 (2024)
7. Sajadmanesh, S., Shamsabadi, A.S., Bellet, A., Gatica-Perez, D.: Gap: Differentially private graph neural networks with aggregation perturbation. In: USENIX Security 2023-32nd USENIX Security Symposium (2023)
8. Tran, C., Fioretto, F., Van Hentenryck, P., Yao, Z.: Decision making with differential privacy under a fairness lens. In: IJCAI. pp. 560–566 (2021)
9. Wang, X., Gu, T., Bao, X., Chang, L., Li, L.: Individual fairness for local private graph neural network. Knowledge-Based Systems **268**, 110490 (2023)
10. Wang, Y., Zhao, Y., Dong, Y., Chen, H., Li, J., Derr, T.: Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. pp. 1938–1948 (2022)
11. Zhang, Y., Zhao, Y., Li, Z., Cheng, X., Wang, Y., Kotevska, O., Philip, S.Y., Derr, T.: A survey on privacy in graph neural networks: Attacks, preservation, and applications. IEEE Transactions on Knowledge and Data Engineering (2024)
12. Zhao, T., Hu, H., Cheng, L.: Unveiling the role of message passing in dual-privacy preservation on gnns. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 3474–3483 (2023)
13. Zhu, Y., Li, J., Chen, L., Zheng, Z.: The devil is in the data: Learning fair graph neural networks via partial knowledge distillation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. pp. 1012–1021 (2024)