Equipping Federated Graph Neural Networks with Structure-aware Group Fairness

Nan Cui, Xiuling Wang, Wendy Hui Wang, Violet Chen, Yue Ning
Stevens Institute of Technology
{ncui, xwang193, hwang4, vchen3, yue.ning}@stevens.edu

***Abstract*—Graph Neural Networks (GNNs) are used for graph data processing across various domains. Centralized training of GNNs often faces challenges due to privacy and regulatory issues, making federated learning (FL) a preferred solution in a distributed paradigm. However, GNNs may inherit biases from training data, causing these biases to propagate to the global model in distributed scenarios. Addressing this, we introduce F^{2}GNN, a Fair Federated Graph Neural Network, to enhance group fairness. Recognizing that bias originates from both data and algorithms, F^{2}GNN aims to mitigate both types of bias under federated settings. We offer insights into the relationship between data bias and statistical fairness metrics in GNNs. Building on this, F^{2}GNN features a *fairness-aware local model update scheme* and a *fairness-weighted global model update scheme,* considering both data bias and local model fairness during aggregation. Empirical evaluations show F^{2}GNN outperforms baselines in fairness and accuracy.**

***Index Terms*—Graph Neural Networks, Federated Learning, Group Fairness**

I. Introduction

Graph neural networks (GNNs) have emerged as a formidable tool for generating meaningful node representations [14] and making accurate predictions on nodes by leveraging graph topology [29]. Despite its success in graph analytics, training GNNs over centralized graph data has been limited in practice due to privacy concerns. Federated learning, a trending distributed learning paradigm, has emerged as a promising solution. Several federated GNNs [26], [28] have been designed recently.

Recent studies have revealed that predictions of GNNs (over centralized data) can be unfair and have undesirable discrimination [3], [11]. Under the federated learning setting, the bias in the local GNN models can be easily propagated to the global model. While most of the prior works [5], [6] mainly consider the federated learning models trained over non-graph data, none of them have studied federated GNNs. How to enhance the fairness of GNNs under federated settings remains to be largely unexplored.

In general, the bias of GNNs can be stemmed from two different sources: (1) bias in graph data [11], [15], and (2) bias in learning algorithms (e.g., the message-passing procedure of GNNs) [2], [12]. In this paper, we aim to enhance the group fairness of federated GNNs by mitigating both types of bias. In particular, in terms of data bias, we group the nodes by their sensitive attribute (e.g., gender and race). Then we group the links between nodes based on the sensitive attribute values of the connecting nodes: (1) the *intra-group links* that connect nodes belonging to the same node groups (i.e., the same sensitive attribute value), and (2) the *inter-group links* that connect nodes that belong to different node groups. Based on the link groups, we consider the data bias that takes the form of the *imbalanced distribution between inter-group and intra-group links*, which is a key factor to the disadvantage of minority groups by GNNs [11]. On the other hand, in terms of model fairness, we consider two well-established group fairness definitions: *statistical parity* (SP) [1], [2] and *equalized odds* (EO) [12], [24].

In this paper, we propose F^{2}GNN, one of the first federated graph neural networks that enhance group fairness of both local and global GNN models. Our contributions are summarized as follows: 1) We provide theoretical insights on the connection between data bias (imbalanced distributions between inter-group and intra-group links) and model fairness (SP and EO); 2) Based on the theoretical findings, we design F^{2}GNN that enhances group fairness of both local and global GNN models by taking both the data bias of local graphs and statistical fairness metrics of local models into consideration during the training of both local and global models; 3) Through experiments on two real-world datasets, we demonstrate that F^{2}GNN outperforms the baseline methods in terms of both fairness and model accuracy.

II. Federated Graph Learning

Notations. We consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, comprising nodes $\mathcal{V}$, edges $\mathcal{E}$, and node features $\mathbf{X}$, with $\mathbf{A}$ as its adjacency matrix. $\mathbf{Z}$ represents the node representations (embedding) learned by a GNN model. This paper focuses on node classification, using y and $\hat{y}$ to represent the ground truth and predicted labels, respectively.

Federated Graph Learning. This study explores *horizontal federated learning* (HFL) setting [4], involving K clients $\{C_1, \ldots, C_K\}$ and a server $\mathcal{S}$. Each client C_i possesses a unique private graph $\mathcal{G}_i$, with different $\mathcal{V}_i$ and $\mathbf{A}_i$, while sharing the same set of node features.

In HFL, clients retain their local graphs, avoiding sharing with the server $\mathcal{S}$ for reasons such as privacy. The goal of HFL is to optimize the global objective function while maintaining the privacy of local graphs:

$$\arg\min_{\omega} \sum_{i=1}^{K} \mathcal{L}_i(\omega) = \arg\min_{\omega} \sum_{i=1}^{K} \sum_{j=1}^{N_i} \boldsymbol{\ell}_i(v_j, \mathcal{G}_i; \omega).$$

Here, $\mathcal{L}_i(\omega)$ is the loss function for client C_i, parameterized by ω, N_i is the number of nodes in the local graph $\mathcal{G}_i$, and $\boldsymbol{\ell}_i(v_j, \mathcal{G}_i; \omega)$ denotes the average loss over the local graph of client C_i.

In this work, we adapt the SOTA FedAvg framework [18] to our context. In FedAvg, only model parameters are exchanged between $\mathcal{S}$ and each client C_i. Specifically, in each round t, $\mathcal{S}$ sends the current global model parameters ω_t to a selected subset of clients for local training. Each selected client C_i refines ω_t^i locally and returns the updated parameters to $\mathcal{S}$, which then aggregates them to form the new global model ω_{t+1}, subsequently broadcasted for the next round of training. The process continues until convergence is achieved.

III. Group Fairness and Data Bias

This paper emphasizes group fairness, where the model output should not discriminate between *protected groups* (e.g., female) and *un-protected groups* (e.g., male) based on a sensitive attribute like gender [1].

Node groups. Adapting conventional group fairness, we assume each graph node has a *sensitive attribute* s (e.g., gender) [19]. Nodes are grouped by this attribute's values. Considering a binary attribute, we use $s = 0$ and $s = 1$ for protected and un-protected node groups.

Edge categorization. Edges are categorized based on node groups into: (1) *inter-group edges* connecting nodes from different groups (e.g., the edges between female and male nodes); (2) *intra-group edges* connecting nodes within the same group (e.g., the edges between male and male nodes).

Data bias in edge distribution. Recent studies indicate that GNN unfairness can arise from imbalanced edge distributions in training graphs [11]. We focus on this imbalance between inter-group and intra-group edges. In particular, we define *group balance score* (GBS) to quantify this bias.

Definition 1. *For a graph $\mathcal{G}$, with $|\mathcal{E}_{inter}|$ as inter-group edges and $|\mathcal{E}_{intra}|$ as intra-group edges, and $|\mathcal{E}| = |\mathcal{E}_{intra}| + |\mathcal{E}_{inter}|$ as total edges, GBS (denoted as B) is defined as follows:*

$$B = 1 - (|H_{intra}| - |H_{inter}|), \tag{1}$$

where $H_{intra} = \frac{|\mathcal{E}_{intra}|}{|\mathcal{E}|}$ and $H_{inter} = \frac{|\mathcal{E}_{inter}|}{|\mathcal{E}|}$.

A higher B indicates a better balance between different edge types. Maximum balance ($B = 1$) occurs when both edge groups are equal (i.e., $|H_{\text{intra}}| = |H_{\text{inter}}|$).

Group fairness of models. To assess the group fairness of the GNN model, we employ two established fairness definitions: *statistical parity* (SP) [1], [2], [15] and *equalized odds* (EO) [12], [24]. SP measures the difference in positive outcome probabilities between node groups as: $\Delta_{\text{SP}} = |P(\hat{y} = 1|s = 0) - P(\hat{y} = 1|s = 1)|$ reflecting the positive rate difference between two node groups. EO gauges the difference in true positive rates of node groups as: $\Delta_{\text{EO}} = |P(\hat{y} = 1|y = 1, s = 0) - P(\hat{y} = 1|y = 1, s = 1)|$ indicating the true positive rate difference for these groups.

In this paper, we adapt both fairness definitions to node classification tasks. Both SP and EO provide complementary insights into fairness in node classification. While other accuracy measurements exist, they are topics for future exploration. We also consider both *local fairness* and *global fairness*. Previous research [22] suggests that while local and global fairness do not necessarily imply each other, global fairness is influenced by local fairness under IID distribution. Our empirical studies (Section VI) show that F^{2}GNN achieves both local and global fairness, even for non-IID distributions.

IV. Connection between Data Bias and Model Fairness

Data bias has been identified as a significant source of model unfairness. In this section, we investigate how the imbalanced distribution between intra-group and inter-group links affects the model fairness, specifically in terms of SP and EO.

Given the complexity of directly examining the impact of data bias on SP and EO, we approach model fairness through graph representations. Intuitively, group fairness requires that model outputs should not differ based on sensitive attributes. If graph representations are invariant to these attributes, group fairness can be achieved [15]. To measure the correlation between the graph representation and the sensitive attribute, we utilize the *Point-Biserial Correlation* [9].

Definition 2 (Point-Biserial Correlation Coefficient)**.** *The point-biserial correlation coefficient quantifies the degree of association between a continuous variable and a dichotomous variable. Given a dichotomous variable s ($s \in \{0, 1\}$) and a continuous variable X, the point-biserial correlation coefficient $\rho_{X,s}$ is defined as:*

$$\rho_{X,s} = \frac{\mu_0 - \mu_1}{\sigma_X} \sqrt{\frac{N_0 N_1}{N^2}},$$

where μ_0 (μ_1) is the mean value of X that are associated with $s = 0$ ($s = 1$), N_0 (N_1) is the number of samples in the class $s = 0$ ($s = 1$), $N = N_0 + N_1$, and σ_X is the standard deviation of X.

The traditional point-biserial correlation is defined with a 1-dimensional variable, X. Consistent with prior work [15], we expand this to accommodate higher-dimensional data via a node feature matrix $\mathbf{X}$, represented as $\mathbf{X} \in \mathbb{R}^{N \times d}$, with N as the node number and d the feature dimension. Here, μ_0 (μ_1) is the mean of the j-th dimension of node features for nodes with a sensitive attribute $s = 0$ ($s = 1$), where $j \in 1, ..., d$. We will use μ_0 (μ_1) and $\mu_j^{s=0}$ ($\mu_j^{s=1}$) interchangeably. After computing the means μ_0 and μ_1 for a specific column j, the point-biserial correlation is calculated as traditionally defined.

Given a graph $\mathcal{G}$, with s as its sensitive attribute and $\mathbf{Z}$ as its representation (node embeddings), we analyze the relationship between data bias (group balance score) and the correlation $\rho_{\mathbf{Z},s}$ between s and $\mathbf{Z}$. Our theoretical analysis is grounded on two assumptions:

- **Model assumption**: We assume the GNN model contains a linear activation function [25].
- **Data assumption**: Let $\mathbf{X}_j^{s=0}$ and $\mathbf{X}_j^{s=1}$ be the j-th feature of those data samples associated with the sensitive attribute $s = 0$ and $s = 1$ respectively. Following prior

works [12], [17], we assume the data values of $\mathbf{X}_j^s$ follow a Gaussian distribution $\mathcal{N}(\mu_j^s, \sigma_j^s)$, where μ_j^s and σ_j^s are the mean and variance of $\mathbf{X}_j^s$.

Based on these assumptions, we derive the following:

Lemma 3. *Given a graph $\mathcal{G}$ and a GNN model that satisfies the data and model assumptions respectively, the Point-biserial correlation (denoted as $\rho_{\mathbf{Z},s}$) between the graph representation $\mathbf{Z}$ and the sensitive attribute s is measured as the following:*

$$\rho_{\mathbf{Z},s} = (N_0\mu_0 - N_1\mu_1)(H_{intra} - H_{inter}) \cdot \frac{\sqrt{N_0\, N_1}}{\sigma_{\mathbf{Z}} N^2},$$

where H_{intra} and H_{inter} are defined by Definition 1, N_0 (N_1) is the number of nodes associated with $s = 0$ ($s = 1$), $\sigma_{\mathbf{Z}}$ is the standard deviation of $\mathbf{Z}$, N is the total number of nodes of $\mathcal{G}$, and μ_0 (μ_1) is the mean value of nodes that are associated with $s = 0$ ($s = 1$).

When $\rho_{\mathbf{Z},s} = 0$ (indicating graph representation's independence from the sensitive attribute), it implies $N_0\mu_0 = N_1\mu_1$ or $H_{\text{intra}} = H_{\text{inter}}$. Given that N_0 and N_1 are constants for a graph, we present the subsequent theorem:

Theorem 4. *Let $\mathcal{G}$ be a graph that satisfies the given data assumption, and $\mathbf{Z}$ be the representation of $\mathcal{G}$ learned by a GNN model that satisfies the given model assumption. Then, the Point-biserial correlation $\rho_{\mathbf{Z},s}$ between the graph representation $\mathbf{Z}$ and the sensitive attribute s is positively correlated/monotonically increasing with $|H_{intra} - H_{inter}|$. In particular, $\rho_{\mathbf{Z},s} = 0$ when $H_{intra} = H_{inter}$.*

The Theorem 4 suggests that the balance between inter-group and intra-group links is crucial for the correlation between graph representations and the sensitive attribute. A balanced distribution should lead to a fairer model. This insight guides the design of our method.

V. Details of F^{2}GNN Algorithm

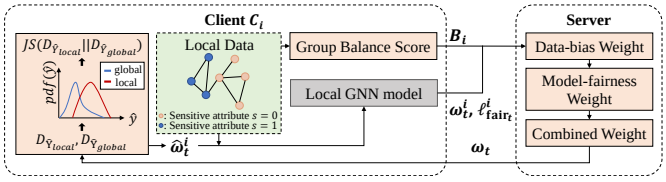


Fig. 1: The overview of F^{2}GNN. In each iteration, client C_i receives the global model, calculates the Jensen-Shannon (JS) divergence js_t^i, and updates its local model $\hat{\omega}_t^i$ using a 2-layer GCN over several epochs. The updated model ω_t^i and the group balance score B_i are sent to the server. The server then computes the data-bias and model-fairness weights, integrates them, and updates the global model ω_t using the combined weight.

In a nutshell, F^{2}GNN follows the principles of FedAvg [18]: in each round t, the server $\mathcal{S}$ uniformly samples a set of clients. Each selected client C_i receives the current global parameters ω_t, and updates its local model ω_t^i by learning over its local graph $\mathcal{G}_i$ with ω_t as the initialized model parameters. Then, each client C_i sends ω_t^i back to $\mathcal{S}$. The server aggregates all received local model updates on ω_t to obtain ω_{t+1}. To ensure both local and global fairness within the federated learning framework, F^{2}GNN comprises two main components:

- *Client-side fairness-aware local model update scheme*: Each client C_i determines ω_t^i by incorporating local fairness into the loss function of its local model as a penalty term.
- *Server-side fairness-weighted global model update scheme*: When $\mathcal{S}$ aggregates all local model updates from the clients, it considers both the fairness metrics of local models (like SP or EO) and the data bias of local graphs as aggregation weights.

Figure 1 illustrates the framework of F^{2}GNN. Next, we present the details of each component.

A. Client-side Fairness-aware Local Model Update

Intuitively, the local model update scheme pursues two objectives: 1) Ensuring convergence of both the local models on clients and the global model on the server, particularly when client data is non-iid, and 2) Integrating fairness (both SP and EO) into local models by seamlessly adding the fairness constraint during training. To meet these goals, we employ two techniques: (i) model interpolation using the Jensen-Shannon (JS) divergence [20], and (ii) introducing fairness as a penalty in the local models' loss function. Next, we discuss the details of each technique.

JS divergence between global and local models. In federated GNN training, a challenge arises when local client data is not IID [16]. This data distribution variance can hinder the global model's convergence due to conflicting goals of minimizing local and global empirical losses. To address this, F^{2}GNN, inspired by Zheng *et al.* [30], evaluates the disparity between local and global models, incorporating this difference during local model parameter updates.

Specifically, consider ω_t as the global model parameters at epoch t and ω_{t-1}^i as the local parameters for client C_i at epoch $t-1$. The distributions of node labels in C_i's local graph, predicted by the global and local models, are represented as $D_{\hat{\mathbf{Y}}_{\text{global}}}$ and $D_{\hat{\mathbf{Y}}_{\text{local}}}$ respectively. When client C_i receives ω_t, it computes the Jensen-Shannon (JS) divergence, js_t^i, between these distributions: $\text{js}_t^i = \text{JS}(D_{\hat{\mathbf{Y}}_{\text{global}}} || D_{\hat{\mathbf{Y}}_{\text{local}}})$.

A higher js_t^i suggests a significant difference between the global and local models. Using this, client C_i determines its initial local model update, $\hat{\omega}_t^i$, as:

$$\hat{\omega}_t^i \leftarrow (1 - \text{js}_t^i) \cdot \omega_{t-1}^i + \text{js}_t^i \cdot \omega_t. \tag{2}$$

From Eqn. (2), if the global model diverges considerably from the local one, ω_t gains prominence over ω_t^i. As a result, ω_t^i aligns more with ω_t, aiding the global model's convergence to a stationary point.

Fairness penalty term. To instill fairness in local models, we adopt an in-process approach, integrating fairness during training. This is commonly done by adding a fairness constraint or penalty to the objective function [27]. Instead of using sensitive features as GNN inputs, they're incorporated during training as a disparity penalty. The fairness penalty for

client C_i at epoch t (denoted as $\boldsymbol{\ell}^i_{\text{fair}_t}$) is the sum of the norm-1 of both SP and EO fairness types over C_i's local graph $\mathcal{G}_i$:

$$\boldsymbol{\ell}^i_{\text{fair}_t} = ||\Delta^i_{\text{SP}_t}||_1 + ||\Delta^i_{\text{EO}_t}||_1.$$

The fairness penalty is incorporated into the local model's loss function as:

$$\boldsymbol{\ell}(\omega^i_t; \mathbf{Z}_i) = \boldsymbol{\ell}^i_{\text{util}_t} + \alpha\, \boldsymbol{\ell}^i_{\text{fair}_t}, \tag{3}$$

with $\boldsymbol{\ell}^i_{\text{util}_t}$ representing the utility loss (like entropy loss for node classification) and α being an adjustable hyper-parameter.

Local model parameter updates. To finalize the local model update, client C_i performs several SGD steps:

$$\omega^i_t \leftarrow \hat{\omega}^i_t - \eta \cdot \nabla \boldsymbol{\ell}(\omega^i_t; \mathbf{Z}_i),$$

where $\hat{\omega}^i_t$ is the initialized local model update, η is the learning rate, and $\nabla \boldsymbol{\ell}(\omega^i_t; \mathbf{Z}_i)$ is the gradient of the loss function.

Upon concluding local training, client C_i transmits its local model parameters ω^i_t, the local fairness loss $\boldsymbol{\ell}^i_{\text{fair}_t}$, and a group balance score B^i to the server $\mathcal{S}$. We remark that sharing aggregated information of the local graphs such as $\boldsymbol{\ell}^i_{\text{fair}_t}$ and B^i with the server will make attacking individual information harder and thus protect their privacy.

B. Server-side Fairness-weighted Global Model Update

The server-side model update scheme in each iteration consists of four steps:

- *Step 1.* Calculate the *data-bias weight* $\gamma_{\mathcal{E}_t}$ based on the group balance scores provided by each client;
- *Step 2.* Derive the *model-fairness weight* $\gamma_{\mathcal{F}_t}$ using the fairness loss metrics uploaded by the clients, ensuring fairness metrics are integrated into the global model aggregation process;
- *Step 3.* Combine the data-bias weight and model-fairness weight into a combined weight (γ_t); and
- *Step 4.* Aggregate the local model updates with the global model utilizing the combined weight.

The following elaboration provides a detailed description of each step.

Data-bias weight. Upon obtaining local parameters from the selected subset of clients at epoch $t-1$, the server first constructs a data-bias weight $\gamma_{\mathcal{E}_t} \in \mathbb{R}^{K'}$, where K' represents the number of clients in the subset:

$$\gamma_{\mathcal{E}_t} = \text{softmax}\left(B^1, ..., B^{K'}\right).$$

Each element in $\gamma_{\mathcal{E}_t}$ corresponds to the group balance score B^i for each client C_i. Given that each client C_i maintains a graph $\mathcal{G}_i$, the data-bias weight $\gamma_{\mathcal{E}_t}$ effectively gauges the equilibrium between inter- and intra-edges within each client graph.

Model-fairness weight. While the group balance score provides a useful measure of the balance between inter- and intra-edges within each client graph, it is static and unable to capture the model's evolving learning process regarding fairness. To address this, we also consider a dynamic *fairness metric weight* $\gamma_{\mathcal{F}} \in \mathbb{R}^{K'}$ that evaluates the statistical parity $\Delta^i_{\text{SP}_t}$ and equalized odds $\Delta^i_{\text{EO}_t}$ of each client's local model at each iteration:

$$\gamma_{\mathcal{F}_t} = \exp\left(\text{softmax}\left(\gamma'_{\mathcal{F}_t}\right)\right).$$

To calculate $\gamma_{\mathcal{F}_t}$, we start by a weight vector $\gamma'_{\mathcal{F}_t}$, which takes each client's sum of two types of group fairness metrics as the element. Therefore, $\gamma'_{\mathcal{F}_t}$ can be illustrated as:

$$\gamma'_{\mathcal{F}_t} = \left[\Delta^1_{\text{SP}_t} + \Delta^1_{\text{EO}_t}, ..., \Delta^{K'}_{\text{SP}_t} + \Delta^{K'}_{\text{EO}_t}\right].$$

To magnify the effect of this dynamic model-fairness weight $\gamma'_{\mathcal{F}_t}$, we use an exponential function to rescale it after passing through a softmax function. This expands its range while maintaining its relative proportions as exponential functions grow faster than linear ones.

The softmax function is crucial for normalizing the two weight vectors, ensuring their comparability. Given the unique nature of the group balance score B, confined to the interval $[0, 1]$, every component of $\gamma_{\mathcal{E}_t}$ respects these boundaries. In contrast, each element in $\gamma'_{\mathcal{F}_t}$ merges two statistical fairness metrics for each client, resulting in a value within the $[0, 2]$ range. Through the softmax function, we align the scales of these weights, streamlining their integration and subsequent hyperparameter adjustments.

Combined weight. After deriving $\gamma_{\mathcal{E}_t}$ and $\gamma_{\mathcal{F}_t}$, they are merged into a single weight vector $\gamma_t \in \mathbb{R}^{K'}$ as:

$$\gamma_t = \text{softmax}\left((\lambda \cdot \gamma_{\mathcal{E}_t} + \gamma_{\mathcal{F}_t})/\tau\right), \tag{4}$$

where λ is a hyperparameter, and τ denotes the temperature parameter of the softmax function. This formula transforms the weights $\gamma_{\mathcal{E}_t}$ and $\gamma_{\mathcal{F}_t}$ into probability distributions that total one, resulting in the weight vector γ_t. The temperature parameter modulates the distribution's smoothness, and λ controls the emphasis on data bias in the final combination.

Global model updating. Utilizing the consolidated weights, the global model ω_t is updated by aggregating the local models ω^i_t ($i \in 1, ..., K'$) from K' clients, weighted by the vector γ_t:

$$\omega_t \leftarrow \sum_{i=1}^{K'} \gamma^i_t \cdot \omega^i_t, \tag{5}$$

where γ^i_t represents the i-th element of the combined weight vector. The server then disseminates the refreshed global model ω_t to all clients.

VI. Experimental Evaluation

In this section, we present the results of our experiments on two real-world datasets for node classification. Our proposed approach, F^{2}GNN, is evaluated and compared against several baseline schemes in terms of both utility and fairness metrics.

A. Experimental Setup

Datasets. We employ two real-world datasets, Pokec-z and Pokec-n, that are widely used for GNN training [2], [3], [12], [15]. Both Pokec-z and Pokec-n graphs are social network data collected from two regions in Slovakia [21]. We pick the `region` attribute as the sensitive attribute and `working fields` as the label for both Pokec-z and Pokec-n graphs. We binarize the sensitive attributes and prediction labels for all the two datasets.

Evaluation metrics. In terms of GNN performance, we measure *Accuracy* and *AUC score* for node classification. Regarding fairness, we measure Δ_{EO} and Δ_{SP} of the predictions. Intuitively, smaller Δ_{EO} and Δ_{SP} indicate better fairness.

TABLE I: Node classification and fairness evaluation on the test sets of the **Pokec-z** and **Pokec-n** datasets.

Method	Global test set							
	Pokec-z				Pokec-n			
	Accuracy(%) ↑	AUC(%) ↑	Δ_{SP} (%) ↓	Δ_{EO}(%) ↓	Accuracy(%) ↑	AUC(%) ↑	Δ_{SP} (%) ↓	Δ_{EO}(%) ↓
FairAug$_{+FL}$	64.62 ± 0.65	64.73 ± 0.67	4.01 ± 0.28	3.98 ± 0.51	64.38 ± 0.38	62.45 ± 0.47	4.31 ± 1.32	5.98 ± 1.96
FairFed	65.48 ± 2.63	69.72 ± 2.7	2.92 ± 1.28	3.08 ± 1.21	61.16 ± 2.46	61.97 ± 1.93	2.66 ± 1.20	3.69 ± 2.95
F^{2}GNN	$\mathbf{68.17 \pm 0.19}$	$\mathbf{73.47 \pm 0.52}$	$\mathbf{1.66 \pm 1.02}$	$\mathbf{1.49 \pm 0.52}$	$\mathbf{67.00 \pm 0.27}$	$\mathbf{69.62 \pm 1.34}$	$\mathbf{0.85 \pm 0.31}$	$\mathbf{1.00 \pm 1.03}$
	Local test sets							
Method	Local Acc(%) ↑	Local AUC(%) ↑	Local Δ_{SP} (%) ↓	Local Δ_{EO}(%) ↓	Local Acc(%) ↑	Local AUC(%) ↑	Local Δ_{SP} (%) ↓	Local Δ_{EO}(%) ↓
FairAug$_{+FL}$	63.33 ± 4.91	57.8 ± 3.76	8.54 ± 0.53	8.54 ± 0.53	61.03 ± 5.89	52.50 ± 4.23	21.12 ± 0.65	28.11 ± 0.71
FairFed	59.44 ± 3.60	63.7 ± 12.13	8.12 ± 3.32	7.69 ± 1.96	56.26 ± 2.42	58.41 ± 1.75	8.38 ± 4.93	7.79 ± 4.36
F^{2}GNN	$\mathbf{68.43 \pm 1.99}$	$\mathbf{75.74 \pm 2.13}$	$\mathbf{7.24 \pm 0.99}$	$\mathbf{7.35 \pm 1.58}$	$\mathbf{68.19 \pm 2.40}$	$\mathbf{69.49 \pm 3.81}$	$\mathbf{8.38 \pm 2.27}$	$\mathbf{7.23 \pm 5.11}$

Furthermore, we measure two kinds of trade-offs between fairness and model accuracy as follows:

$$\text{Trade-off}_{\text{ACC}} = \text{Accuracy} \,/\, (\Delta_{\text{EO}} + \Delta_{\text{SP}}),$$
$$\text{Trade-off}_{\text{AUC}} = \text{AUC} \,/\, (\Delta_{\text{EO}} + \Delta_{\text{SP}}).$$

Intuitively, a model that delivers higher accuracy and lower EO and SP values will lead to a better trade-off.

Baselines. To the best of our knowledge, there is no existing work on fairness issues in federated GNNs. Thus we adapt existing methods of fair federated learning and fair GNNs. We consider two baseline methods specifically:

- We deploy GNNs under an existing fairness-enhancing federated learning framework named **FairFed** [6].
- We deploy the fairness-aware GNN model **FairAug** [15] in the federated framework (denoted as FairAug$_{+FL}$), replacing local models with it.

To ensure a fair comparison between different models on each dataset, we remove all the isolated nodes in the datasets.

GNN Setup. We deploy a 2-layer graph convolutional networks (GCNs), tailoring neuron configurations to each dataset. Specifically, the GCNs are set up with 64 neurons for the Pokec-n dataset, whereas 128 neurons per layer for the Pokec-z dataset. We utilize the Rectified Linear Unit (ReLU) [8] and the Adam optimizer [13]. Our F^{2}GNN implementation is developed using the Deep Graph Library (DGL) [23], PyTorch Geometric [7], and NetworkX [10].

Federated learning setup. In the federated learning setting, we randomly select a set of nodes from each graph data set to serve as the centers of egocentric networks, which also determine the number of clients, and we specify a certain number of hops to grow the networks. Each network (or subgraph) is considered as a client. We evaluate our proposed approach on the Pokec datasets in a setting with 30 clients, each holding a 3-hop ego-network.

B. Performance Evaluation

Tables I provide a comprehensive performance comparison of various models on the Pokec datasets, with a particular focus on their fairness attributes Δ_{SP} and Δ_{EO}.

Specifically, we evaluate the models on a global test set as well as local test sets. Each local client has its corresponding test set. After we get the metrics from all clients, we report the median value of these clients for each metric. Among the evaluated methods, our proposed framework, F^{2}GNN consistently outperforms other methods across two datasets in both global and local sets. In particular, F^{2}GNN achieves the lowest Δ_{SP} and Δ_{EO} values, indicating the effectiveness of the proposed scheme in federated settings.

The trade-off performance of each model on the two datasets is illustrated in Table II. The proposed method achieves the highest trade-off values for all datasets, respectively, demonstrating its exceptional ability to balance accuracy and fairness.

TABLE II: Trade-offs of F^{2}GNN and baselines.

Method	Pokec-z		Pokec-n	
	Trade-off$_{\text{ACC}}$ ↑	Trade-off$_{\text{AUC}}$ ↑	Trade-off$_{\text{ACC}}$ ↑	Trade-off$_{\text{AUC}}$ ↑
FairAug$_{+FL}$	11.129	12.0854	8.2651	8.235
FairFed	5.6581	5.6598	5.6581	12.5286
F^{2}GNN	21.6445	23.3264	36.2460	37.6649

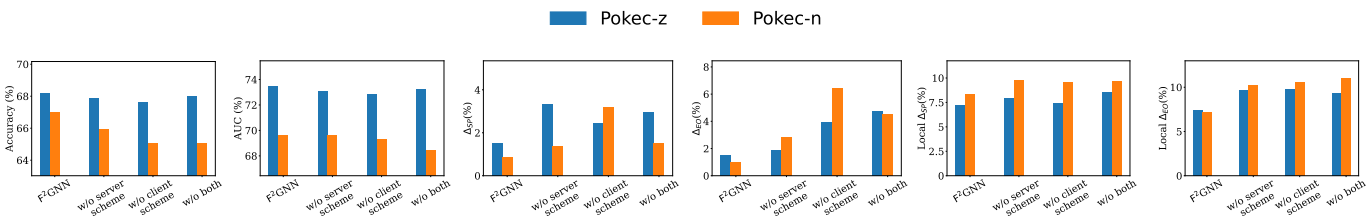
In summary, our proposed F^{2}GNN framework outperforms other baselines in terms of Δ_{SP}, Δ_{EO}, and trade-off values across all datasets. These results validate the effectiveness and robustness of F^{2}GNN in achieving competitive performance while maintaining group fairness in node classification tasks. These findings also highlight the broad applicability of our approach in the field of federated learning, suggesting its potential for adaptation to other domains.

C. Ablation Study

In the ablation study illustrated in Figure 2, we assess the impact of client-side fairness-aware and server-side fairness-weighted updates in our model on the Pokec datasets. Excluding either client-side or server-side schemes results in decreased accuracy and increased disparity in Δ_{SP} and Δ_{EO}, emphasizing their role in balancing performance and fairness. Also, neglecting the server-side fair aggregation adversely affects fairness metrics, underscoring its crucial role in fairness enhancement. The complete model, F^{2}GNN, optimally balances performance and fairness, proving the efficacy of all components in federated graph learning.

VII. Conclusion

In this study, we explore data bias in federated learning with a focus on graph neural networks. We highlight the need to account for the relationship between sensitive attributes



(a) Global Model ACC (b) Global Model AUC (c) Global Δ_{SP} (d) Global Δ_{EO} (e) Local Δ_{SP} (f) Local Δ_{EO}

Fig. 2: Ablation Study

and graph structures to achieve model fairness. Introducing F^{2}GNN, we integrate fairness into both local and global training phases. Our tests on real datasets show our method effectively balances fairness and performance. As a future direction, we recognize the value of enhancing privacy, possibly through methods like homomorphic encryption, ensuring secure data aggregation.

Acknowledgments

This work is supported in part by the US National Science Foundation under grants 1948432 and 2047843. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Barocas, M. Hardt, and A. Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019.

[2] E. Dai and S. Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining*, pages 680–688, 2021.

[3] Y. Dong, N. Liu, B. Jalaian, and J. Li. Edits: Modeling and mitigating data bias for graph neural networks. In *Proceedings of the ACM Web Conference*, pages 1259–1269, 2022.

[4] H. Du, M. Shen, R. Sun, J. Jia, L. Zhu, and Y. Zhai. Malicious transaction identification in digital currency via federated graph deep learning. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops*, pages 1–6, 2022.

[5] W. Du, D. Xu, X. Wu, and H. Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining*, pages 181–189, 2021.

[6] Y. H. Ezzeldin, S. Yan, C. He, E. Ferrara, and S. Avestimehr. Fairfed: Enabling group fairness in federated learning. *CoRR*, abs/2110.00857, 2021.

[7] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[8] K. Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 1969.

[9] G. V. Glass and K. D. Hopkins. *Statistical methods in education and psychology*. Allyn and Bacon Boston, 1996.

[10] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, 2008.

[11] H. Hussain, M. Cao, S. Sikdar, D. Helic, E. Lex, M. Strohmaier, and R. Kern. Adversarial inter-group link injection degrades the fairness of graph neural networks. *arXiv preprint arXiv:2209.05957*, 2022.

[12] Z. Jiang, X. Han, C. Fan, Z. Liu, N. Zou, A. Mostafavi, and X. Hu. FMP: toward fair graph message passing against topology bias. *CoRR*, abs/2202.04187, 2022.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[14] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, Conference Track Proceedings*, 2017.

[15] Ö. D. Köse and Y. Shen. Fair node representation learning via adaptive data augmentation. *arXiv preprint*, abs/2201.08549, 2022.

[16] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations*, 2020.

[17] H. Ling, Z. Jiang, Y. Luo, S. Ji, and N. Zou. Learning fair graph representations via automated data augmentations. In *The Eleventh International Conference on Learning Representations*, 2023.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[19] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6):115:1–115:35, 2022.

[20] F. Nielsen. On a variational definition for the jensen-shannon symmetrization of distances based on the information radius. *Entropy*, 23(4):464, 2021.

[21] L. Takac and M. Zabovsky. Data analysis in public social networks. In *International scientific conference and international workshop present day trends of innovations*, volume 1, 2012.

[22] G. Wang, A. Payani, M. Lee, and R. Kompella. Mitigating group bias in federated learning: Beyond local fairness. *arXiv preprint arXiv:2305.09931*, 2023.

[23] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

[24] 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 *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1938–1948, 2022.

[25] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6861–6871, 2019.

[26] H. Xie, J. Ma, L. Xiong, and C. Yang. Federated graph classification over non-iid graphs. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems*, pages 18839–18852, 2021.

[27] S. Yao and B. Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 2921–2930, 2017.

[28] Y. Yao and C. Joe-Wong. Fedgcn: Convergence and communication tradeoffs in federated training of graph convolutional networks. *CoRR*, abs/2201.12433, 2022.

[29] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji. On explainability of graph neural networks via subgraph explorations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 12241–12252, 2021.

[30] L. Zheng, J. Zhou, C. Chen, B. Wu, L. Wang, and B. Zhang. ASFGNN: automated separated-federated graph neural network. *Peer-to-Peer Netw. Appl.*, 14(3):1692–1704, 2021.