



ComFairGNN: Community Fair Graph Neural Network

Yonas Sium^(✉) and Qi Li^(iD)

Iowa State University, Ames, USA
{yas,qli}@iastate.edu

Abstract. Graph Neural Networks (GNNs) have become the leading approach for addressing graph analytical problems in various real-world scenarios. However, GNNs may produce biased predictions against certain demographic subgroups due to node attributes and neighbors surrounding a node. Understanding the potential evaluation paradoxes due to the complicated nature of the graph structure is crucial for developing effective GNN debiasing mechanisms. In this paper, we examine the effectiveness of current GNN debiasing methods in terms of unfairness evaluation. Specifically, we introduce a community-level strategy to measure bias in GNNs and evaluate debiasing methods at this level. Further, We introduce ComFairGNN, a novel framework designed to mitigate community-level bias in GNNs. Our approach employs a learnable coreset-based debiasing function that addresses bias arising from diverse local neighborhood distributions during GNNs neighborhood aggregation. Comprehensive evaluations on three benchmark datasets demonstrate our model's effectiveness in both accuracy and fairness metrics.

Keywords: fairness · graph neural network · bias · evaluation metrics

1 Introduction

In today's interconnected world, graph learning supports various real-world applications, such as social networks, recommender systems, and knowledge graphs [3, 16]. Graph Neural Networks (GNNs) are powerful for graph representation learning and are used in tasks like node classification and link prediction by aggregating neighboring node information. However, GNNs often overlook fairness, leading to biased decisions due to structural bias and attribute bias like gender, race, and political ideology. These bias can cause ethical dilemmas in critical contexts, such as job candidate evaluations, where a candidate might be favored due to shared ethnic background or mutual acquaintances. To tackle the outlined issue, multiple methods were suggested to evaluate and mitigate the fairness of node representation learning on graphs. Most of these methods aim to learn node representations that can yield statistically fair predictions across subgroups defined based on sensitive attributes. Choosing the right metric to evaluate bias in graphs inherently depends on the specific task at hand. The

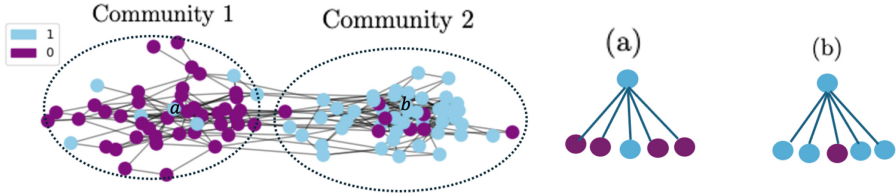


Fig. 1. Illustration of graph polarization: Nodes a and b from different communities share identical labels (same colors), but GNNs learn different embeddings due to varying neighborhood label distributions.

existing methods employed graph-level metrics originally designed for other purposes to address fairness concerns. The primary difficulty in addressing fairness within graphs arises from the frequent correlation between the graph’s topology and the sensitive attribute we aim to disregard. However, due to the complicated nature of graph structures, conducting graph-level fairness evaluations and comparisons of different methods is not as straightforward as commonly reported in the existing methods.

Real-world graphs often polarize into communities based on sensitive attributes (e.g., age, gender, race, ideology), as shown in Fig. 1, where communities form through dense internal connections and sparse external links. For example, U.S. Twitter networks show distinct political segregation [9], where Democratic supporters dominate Northeast and West Coast interactions while Republican supporters influence Texas, Florida, and Iowa networks. Similarly, sports preferences on social media can reflect regional demographic differences.

In this paper, we investigate the critical issue of community fairness in graph neural networks (GNNs) within the context of node classification. Simply using node attributes and structural information as a debiasing mechanism can lead to misleading fairness evaluations and loss of important information [23]. Communities, which reflect local neighborhood structures, often stem from sensitive node attributes [17]. Specifically, GNNs generate node representations by aggregating information from neighboring nodes, which can result in nodes with identical labels but differing neighborhood label distributions, as illustrated in Fig. 1. This leads to structural biases in the learning process. Moreover, such biases may be further amplified through multi-layered recursive neighborhood aggregation in GNNs. While current debiasing methods primarily focus on attribute-based fairness, they fail to address the underlying bias introduced by neighborhood aggregation. As a result, these methods are ineffective at mitigating community-level biases that arise from disparate neighborhood structures.

To address community fairness, we introduce the concept of local structural fairness at the community level for all nodes in the graph. This approach recognizes that the broader structural bias affecting nodes within a community emerges from the GNN’s aggregation process, which is influenced by the

diverse distribution of local neighborhoods. We present ComFairGNN, a novel community-fair Graph Neural Network compatible with any neighborhood aggregation based GNN architecture. Specifically, the coreset-based debiasing function targets the neighborhood aggregation of GNN operation, aiming to bring identical label nodes in different communities closer in the embedding space. This approach mitigates structural disparities across communities, leading to fairer representations for nodes with identical labels.

In summary, our paper makes three key contributions: (1) We measure the problem of fairness and potential evaluation paradoxes at the community level using different fairness evaluation metrics in GNNs. (2) We propose a novel community fairness for GNNs named ComFairGNN that can balance the structural bias for identical label nodes in different communities. ComFairGNN works with any neighborhood aggregation-based GNNs. (3) Comprehensive empirical evaluations validate our method’s effectiveness in enhancing both fairness and accuracy.

2 Preliminaries

We first introduce the notations. A graph $G = (V, E)$ consists of n nodes and edges $E \subseteq V \times V$. The node-wise feature matrix $X \in \mathbb{R}^{n \times k}$ has k dimensions for raw node features, with each row \mathbf{x}_i representing the feature vector for the i -th node. The binary adjacency matrix is $A \in \{0, 1\}^{n \times n}$, and the learned node representations are captured in $H \in \mathbb{R}^{n \times d}$, where d is the latent dimension size and \mathbf{h}_i is the representation for the i -th node. The sensitive attribute $s \in \{0, 1\}^n$ classifies nodes into demographic groups, with edges e_{uv} being intragroup if $s_u = s_v$ and intergroup otherwise. The function $N(u)$ returns the set of neighbors for a node u , providing a structural node embedding. Clusters based on this embedding are denoted as C , with $C_i \subseteq V$ for nodes in the i -th cluster and c as the cluster count. Each C_i represents a community and f measures the statistical notion of fairness across different communities in the graph.

2.1 Graph Neural Network

Graph Neural Networks (GNNs) utilize neighborhood aggregation across multiple layers. At layer l , the representation of node v , denoted as $\mathbf{h}_v^l \in \mathbb{R}^{d_l}$, is:

$$\mathbf{h}_v^l = \sigma \left(\text{AGGR} \left(\{\mathbf{h}_u^{l-1} \mid u \in \mathcal{N}_v\}; \omega^l \right) \right) \quad (1)$$

where $\text{AGGR}(\cdot)$ is an aggregation function like mean-pooling [12] or self-attention [21], σ is an activation function, \mathcal{N}_v is v ’s neighborhood set, and ω^l contains learnable parameters. The initial representation is $\mathbf{h}_v^0 = \mathbf{x}_v$, where \mathbf{x}_v is the input feature vector.

2.2 Node Clustering on Structural Embedding

We partition the graph into communities using node2vec [8] embeddings combined with k -means clustering [15]. Node2vec maps each node $u \in U$ to a d -dimensional vector space $f : U \rightarrow \mathbb{R}^d$. For each node, multiple random walks are performed to capture the network neighborhood $N_s(u)$, maximizing the objective:

$$\arg \max_f \prod_{u \in U} \prod_{u' \in N_s(u)} P(u' | f(u)) \quad (2)$$

where $P(u' | f(u))$ is computed using softmax. These structural embeddings are then clustered using k -means, chosen for its computational efficiency. The k -means algorithm minimizes the within-cluster sum of squares [18]:

$$\arg \min \sum_{i=1}^C \sum_{x \in C_i} \text{DIST}(x, \mu_i) \quad (3)$$

where C_i represents the i -th cluster and μ_i its centroid.

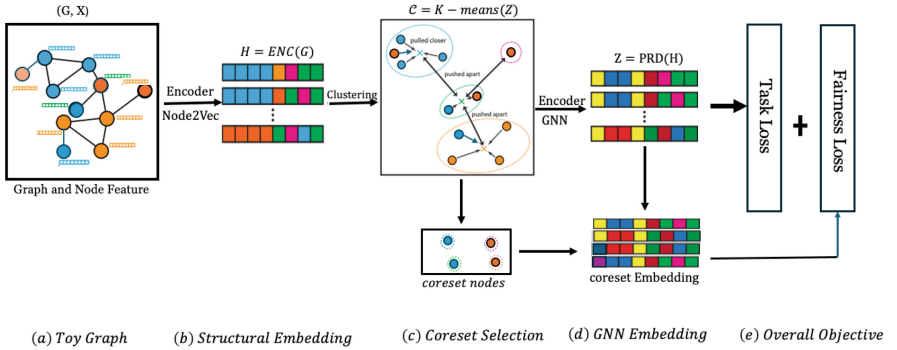


Fig. 2. Overall framework of ComFairGNN

3 Proposed Method

Figure 2 illustrates the ComFairGNN framework. We first identify communities using structural-based clustering (Fig. 2(c)). For debiasing, we divide each community’s nodes into two subgroups based on sensitive features and sample coreset nodes according to neighborhood homophily ratios. These coreset nodes contrast against each other to mitigate structural bias (Fig. 2(d)). The GNN optimizes both task performance and coreset fairness (Fig. 2(e)), debiasing neighborhood aggregation by maximizing similarity between same-labeled nodes.

3.1 Community Level Structural Contrast

We propose a debiasing function using coreset nodes c selected from different communities. Following [20], coresets identify crucial training examples using heuristic criteria. We select nodes equally from subgroups \mathcal{S}_0 and \mathcal{S}_1 across communities. Since GNNs in Eq. 1 only access one-hop contexts and communities tend to be homogeneous in sensitive attributes, selecting coreset nodes from both groups per community helps balance structural disparities.

$$C = \bigcup_{k=1}^K C_{k_{\mathcal{S}_0}} \cup \bigcup_{k=1}^K C_{k_{\mathcal{S}_1}} \subseteq V \quad (4)$$

Here C indicates the coreset sample nodes for all $k \in \{1, \dots, K\}$. K represents the total number of communities, and $C_{k_{\mathcal{S}_0}}$ and $C_{k_{\mathcal{S}_1}}$ represent selected sample nodes from subgroup \mathcal{S}_0 and \mathcal{S}_1 in community k respectively.

To minimize structural bias between groups across communities, we maximize the similarity of similarly-labeled nodes in the coreset, allowing GNNs to implicitly reduce community-based structural differences.

3.2 Coreset Nodes Selection for GNNs Debiasing

To fundamentally eliminate the structural bias of GNNs neighborhood aggregation, which is the key operation in GNNs, we propose to use the neighborhood homophily distribution ratio to select the sample coreset nodes from different communities.

Node Neighborhood Homophily Distribution Ratio: The homophily ratio in graphs is typically defined based on the similarity between connected node pairs. In this context, two nodes are considered similar if they share the same node label. The formal definition of the homophily ratio is derived from this intuition, as follows [14].

Definition 1 (Node Homophily Ratio). *For a graph $G = (\mathcal{V}, \mathcal{E})$ with node label vector y , we define the edge homophily ratio as the proportion of edges linking nodes sharing identical labels. The formal definition is as follows:*

$$h(G, \{y_i; i \in \mathcal{V}\}) = \frac{1}{|n|} \sum_{(i,k) \in \mathcal{E}} f(y_i = y_k), \quad (5)$$

where $|n|$ is the number of neighboring edges of node i in the graph and $f(\cdot)$ is the indicator function.

A node in a graph is generally classified as highly homophilous when its edge homophily ratio $h(\cdot)$ is high, typically falling within the range $0.5 \leq h(\cdot) \leq 1$, assuming an appropriate label context. Conversely, a node is considered heterophilous when its edge homophily ratio is low. Here, $h(\cdot)$ denotes the edge homophily ratio as previously defined.

Figure 3 shows significant variation in neighborhood distribution ratios among same-labeled nodes in pokec-z. Since nodes with identical labels typically share similar features, GNN’s 1-hop aggregation produces different embeddings for these nodes. This embedding disparity across communities may bias the GNN. Therefore, we sample debiasing coreset nodes from all communities based on their neighborhood distributions.

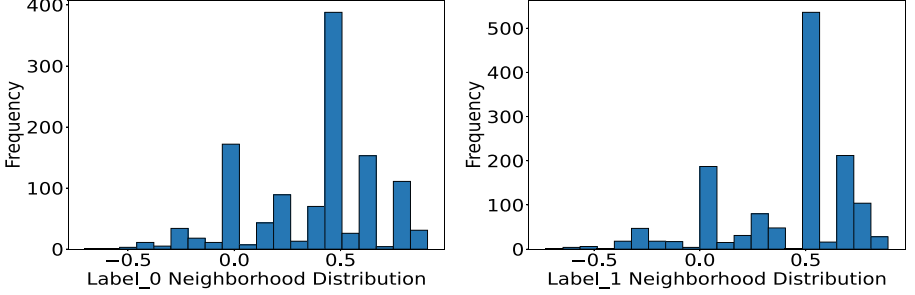


Fig. 3. Edge distribution bias in Pokec-z: GNN embeddings vary for identical nodes based on neighborhood labels

Fairness-Aware Graph Coreset: We aim to identify a representative subset of training nodes $V_w \subset V_{\text{train}} = [n_t]$, where $|V_w| \leq c \ll n_t$, along with an associated set of sample weights. In nearest neighbor message passing models (such as GCN, GAT, and GIN), node i only interacts with nodes j that are within a shortest-path distance of D , denoted as $\{j \in [n] \mid d(i, j) < D\}$. Here, we assume node i neighborhood distribution ratio Sect. 3.2, to select the coreset nodes. This selection is designed to approximate the training loss $\mathcal{L}(\theta)$ across all $\theta \in \Theta$, while maintaining fairness considerations. The graph coreset selection problem is formulated as:

$$\min_{w \in \mathcal{W}} \max_{\theta \in \Theta} \left| \sum_{i \in [n_t]} w_i \cdot \ell([f_\theta(A, X)]_i, y_i) - \mathcal{L}(\theta) \right| \quad (6)$$

The coreset minimizes the maximum error across all θ , requiring both A and X matrices to compute $f_\theta(A, X)$. As illustrated in Algorithm 1, to debias GNN neighborhood aggregation at the community level, we sample from both groups S_0 and S_1 across all communities.

Algorithm 1. Fair Graph Coreset Selection

Require: $G = (V, E)$, features X , groups S_0, S_1 , size K
Ensure: Coreset $C \subseteq V$ {Selected nodes}
1: Initialize $C = \emptyset$, $H = \text{ENC}(G)$ {Initial embedding}
2: **for** each community k in G **do**
3: $n_k = \lfloor K \cdot |V_k| / |V| \rfloor$ {Community budget}
4: $C_{0k} = \text{Sample}(V_k \cap S_0, \lfloor n_k / 2 \rfloor)$
5: $C_{1k} = \text{Sample}(V_k \cap S_1, \lfloor n_k / 2 \rfloor)$
6: $C = C \cup C_{0k} \cup C_{1k}$
7: **end for**
8: $Z = \text{PRD}(H)$
9: Optimize C to minimize Task_Loss + Fairness_Loss
10: **return** C

3.3 Training Objective and Constraints:

We focus on the task of node classification. We incorporate both the classification loss and the fairness loss associated with the classification to enhance the training process as illustrated in Fig. 2(e).

Classification loss: In node classification tasks, the Graph Neural Network (GNNs) final layer typically adapts its dimensionality to align with the number of classes. It utilizes a softmax activation function, wherein the probability of class i is represented by the i^{th} output dimension. To optimize the model’s parameters, we minimize the cross-entropy loss, defined as:

$$\mathcal{L}_{\text{task}} = -\frac{1}{|Y|} \sum_i \sum_j Y_{i,j} \log(\hat{Y}_{i,j}) \quad (7)$$

Here, $|Y|$ denotes the number of labeled examples (rows in Y), while $\hat{Y}_{i,j}$ represents the predicted probability of the i^{th} example belonging to the j^{th} class.

Fairness Loss: As the proposed coreset selection aims to maintain fair representations across different groups, we introduce a fairness loss on the selected coreset samples c . We group the samples in c based on their class label. Let $c_{y=0}$ and $c_{y=1}$ denote the groups of samples in the coreset with labels 0 and 1, respectively for binary classification. We use a similarity-based loss, trying to achieve parity in the average pairwise similarities for the two label groups in c , as follows:

$$\mathcal{L}_{\text{fair}} = \left| \frac{1}{|c_{y=0}|} \sum_{i,j \in c_{y=0}} \text{sim}(x_i, x_j) - \frac{1}{|c_{y=1}|} \sum_{i,j \in c_{y=1}} \text{sim}(x_i, x_j) \right| \quad (8)$$

where $\text{sim}(\cdot, \cdot)$ is a cosine similarity function between two samples of same label. This fairness loss drives the coreset selection toward balancing the average intra-label similarities between the two label groups in c . It aims to constrain the similarity distribution of labels to be similar across the two groups, S_0 and S_1 .

Overall loss: By combining all the above loss terms, we formulate the overall loss as

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{fair}} \quad (9)$$

4 Experiments

We aim to address the following research questions in our experiments. **RQ1:** How reliable is the reported fairness score of graph debiasing methods free from paradox at a community level? **RQ2:** How efficient are the existing graph fairness evaluation methods in estimating the bias of GNNs at a community level? **RQ3:** How efficient is our proposed ComFairGNN method in terms of both accuracy and fairness at the community level?

Table 1. Comparison of GNNs on utility and bias at the graph level with different debiasing methods.

Method	Dataset	ACC \uparrow	AUC \uparrow	$\Delta\text{EO}(\text{w})\downarrow$	$\Delta\text{SP}(\text{w})\downarrow$	$\Delta\text{EO}(\text{w/o})$	$\Delta\text{SP}(\text{w/o})$
Vanilla GCN	Pokec-z	69.29 \pm 1.72	69.51 \pm 2.7	14.43 \pm 0.7	12.88 \pm 0.94	-14.43 \pm 0.7	-12.88 \pm 0.94
	Pokec-n	70.77 \pm 1.32	70.45 \pm 1.7	18.03 \pm 0.57	14.14 \pm 2.01	18.03 \pm 0.57	14.14 \pm 2.01
	Facebook	81.53 \pm 1.52	61.62 \pm 2.1	12.73 \pm 0.7	12.30 \pm 1.0	-12.73 \pm 0.7	-12.30 \pm 1.0
FairGNN	Pokec-z	68.95 \pm 0.1	72.24 \pm 0.3	5.17 \pm 1.3	7.79 \pm 0.7	-5.17 \pm 1.3	-6.79 \pm 0.7
	Pokec-n	69.14 \pm 0.1	70.18 \pm 0.3	4.26 \pm 1.3	3.87 \pm 0.7	4.26 \pm 1.3	3.87 \pm 0.7
	Facebook	85.35 \pm 0.77	64.62 \pm 1.6	2.22 \pm 0.6	6.36 \pm 1.0	2.22 \pm 0.6	6.36 \pm 1.0
NIFTY	Pokec-z	67.34 \pm 1.51	67.49 \pm 1.20	5.89 \pm 1.32	5.85 \pm 0.9	-5.89 \pm 1.32	-5.85 \pm 0.9
	Pokec-n	69.41 \pm 1.51	69.08 \pm 1.20	8.63 \pm 1.32	10.42 \pm 0.9	-8.63 \pm 1.32	-10.42 \pm 0.9
	Facebook	77.07 \pm 1.28	73.05 \pm 1.87	7.88 \pm 1.0	8.95 \pm 1.3	7.88 \pm 1.0	8.95 \pm 1.3
UGE	Pokec-z	63.64 \pm 2.57	64.61 \pm 2.1	14.34 \pm 2.3	11.74 \pm 1.9	-14.34 \pm 2.3	-11.74 \pm 1.9
	Pokec-n	67.48 \pm 2.72	66.67 \pm 2.71	10.34 \pm 2.3	12.74 \pm 1.9	-10.34 \pm 2.3	-12.74 \pm 1.9
	Facebook	64.33 \pm 0.94	63.46 \pm 1.8	5.69 \pm 1.2	13.17 \pm 1.5	5.69 \pm 1.2	13.17 \pm 1.5
ComFairGNN	Pokec-z	70.25 \pm 0.1	71.11 \pm 0.3	0.18 \pm 1.3	1.00 \pm 0.7	0.18 \pm 1.3	1.00 \pm 0.7
	Pokec-n	69.41 \pm 0.7	66.43 \pm 0.97	0.19 \pm 0.45	0.90 \pm 0.67	-0.19 \pm 0.45	-0.90 \pm 0.67
	Facebook	85.35 \pm 0.77	74.62 \pm 2.6	3.96 \pm 0.6	7.63 \pm 1.0	3.96 \pm 0.6	7.63 \pm 1.0

Table 2. Dataset statistics and community distribution

Dataset	Nodes	Edges	Features	Classes	Comm_1	Comm_2	Comm_3	Comm_4	Comm_5
Pokec-z	67,796	882,765	276	2	309	282	259	1412	302
Pokec-n	66,569	198,353	265	2	428	360	361	389	-
Facebook	1,034	26,749	573	2	55	37	65	-	-

4.1 Experimental Setup and Datasets

Datasets. We evaluate on three real-world datasets for node classification: *Pokec-z*, *Pokec-n*, and *Facebook*. The Pokec datasets are subsets from Slovakia’s Pokec social network [19], where nodes represent users, edges represent friendships, and location is the sensitive attribute for classifying users’ work fields. The Facebook dataset [11] similarly uses friendship connections with gender as the sensitive attribute. Dataset statistics are summarized in Table 2.

Evaluation Metrics We use ACC and AUC scores to evaluate the utility performance. To evaluate fairness, we use two commonly used fairness metrics, i.e., $\Delta_{DP} = |P(\hat{y} = 1|s = 0) - P(\hat{y} = 1|s = 1)|$ [7] and $\Delta_{EO} = |P(\hat{y} = 1|y = 1, s = 0) - P(\hat{y} = 1|y = 1, s = 1)|$ [10]. \hat{y} and y denote the node label prediction and ground truth, respectively. For Δ_{DP} and Δ_{EO} , a smaller value indicates a fairer model prediction.

Baseline Methods: We investigate the effectiveness of four state-of-the-art GNN debiasing methods at a graph level and community level, namely FairGNN [4], NIFTY [1], and UGE [22]. FairGNN utilizes adversarial training to eliminate sensitive attribute information from node embeddings. NIFTY improves fairness by aligning predictions based on both perturbed and unperturbed sensitive attributes. This approach generates graph counterfactuals by inverting the sensitive feature values for all nodes while maintaining all other attributes. UGE derives node embeddings from a neutral graph that is free of biases and unaffected by sensitive node attributes, addressing the issue of unbiased graph embedding. We use GCN [12] as the backbone GNN model.

Implementation Setup: We use a two-layer GCN [12] encoder and MLP predictor for all experiments, consistent with baseline methods implemented from PyGDebias [6]. Each method runs for 400 epochs with learning rates in $\{0.1, 0.01, 0.001\}$, using hyperparameters from their original papers and repeating three times. Experiments were conducted on 16GB NVIDIA V100 GPUs.

Fairness Paradoxes: To address **RQ1**, we evaluate the effectiveness of current debiasing techniques for GNNs at the community level and compare the consistency of their debiasing performance when applied to the entire graph. We first measure the group fairness of the GCN model with the debiasing method applied to the entire original input graph G . Next, we measure the group fairness of the same GCN model with the debiasing method applied at the community level. We observe that the fairness performance across the communities is not as consistent as the fairness score reported for the entire graph across datasets for all baseline methods. For example, in Fig. 4 {b, c, d}, we observe that the unfairness estimation for Pokec-z dataset using Δ_{EO} is higher for communities {2, 3, 4, 5} compared to Δ_{EO} for the entire dataset when FairGNN debiasing method is used.

Over-Simplification of Fairness Evaluation RQ2: The absolute values in Δ_{SP} and Δ_{EO} metrics can mask local disparities across communities. While overall metrics might show moderate fairness when averaged, they obscure cases

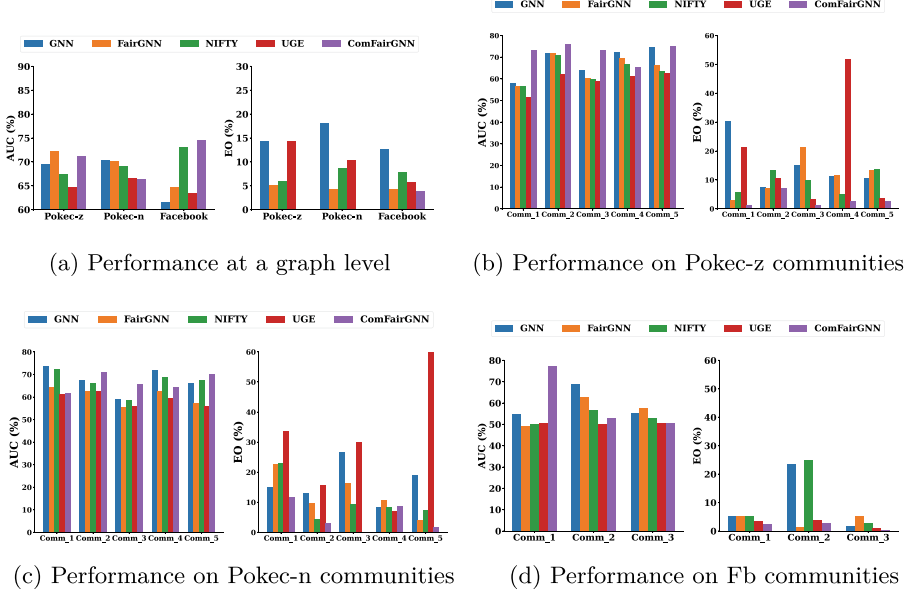


Fig. 4. Performance comparison of different GNN debiasing methods across datasets and communities. (a) Graph-level performance across different datasets (Pokec-z, Pokec-n, Facebook). (b, c, and d) Community-wise performance breakdown on Pokec-z, Pokec-n and Facebook datasets. Showing AUC(%) and EO(%) metrics.

where a group experiences high bias in some communities and low bias in others, as shown in Fig. 4. This averaging effect fails to capture the actual fairness experienced by individuals in different parts of the graph.

Evaluation and Main Results (RQ3). As GNNs learn by neighborhood aggregation Sect. 2.1, we evaluate fairness within different subgraphs or communities within the larger graph. This can help identify and address local disparities more effectively. We select S_0 and S_1 nodes within the community to evaluate the model’s disparity locally, which may present more prominent biases. We select top and bottom 15 sample homophily neighborhood distribution ratio Sect. 3.2, from each community to formulate the representative coreset nodes from the training set during training. These nodes are sufficient to cover the local contextual structures where biases are rooted. As shown in Table 1, ComFairGNN has both the best overall classification performance than the baselines methods. In terms of fairness, ComFairGNN respectively reduces ΔSP and ΔEO by 4.9% and 4.1%, compared with the best performed baseline on both Pokec-z and Pokec-n datasets. Additionally, as shown in Fig. 4, this pattern is consistent across all communities in the datasets. Hence, the fairness improvement achieved by ComFairGNN aligning with our motivation.

Ablation Study: To validate ComFairGNN’s, we investigate different coreset sizes of C hyperparameter. As shown in Fig. 5 $C = 30$ sample size relatively performs better.

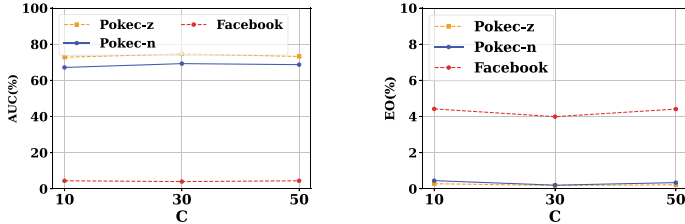


Fig. 5. AUC and EO Performance Comparison Across Different Coreset Sizes C

5 Related Work

5.1 Fairness in Graph

Research in Graph Neural Networks (GNNs) fairness has produced several approaches to measure and mitigate bias. Recent works have primarily focused on group fairness, ensuring models don’t discriminate against certain groups. EDITS [5] modifies the adjacency matrix and node attributes to reduce the Wasserstein distance between different groups. FairGNN [4] incorporates fairness regularization to ensure equitable treatment in both representations and predictions. UGE [22] learns unbiased node representations by eliminating sensitive attribute influences, while NIFTY [2] enhances counterfactual fairness through triplet-based objectives and layer-wise weight normalization. DegFairGNN [13] addresses fairness for nodes with varying degrees within groups. However, this approach is computationally expensive for large graphs. Additionally, like other existing methods, DegFairGNN has not fully addressed the inherent local structural bias in graphs.

6 Conclusion

This paper examines fairness paradoxes in graph debiasing methods at the community level. By analyzing structural communities, we reveal fairness inconsistencies across demographic subgroups that are masked when using absolute measures of ΔSP and ΔEO . To address community-level bias in GNN neighborhood aggregation, we propose ComFairGNN, a fairness-aware framework that modulates representative coreset nodes through embedding similarity contrast. Experiments on three benchmark datasets demonstrate that ComFairGNN effectively improves both accuracy and fairness metrics.

Acknowledgments. This work is partially supported by the National Science Foundation under Grant No. 2152117 and NSF-CAREER 2237831. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Agarwal, C., Lakkaraju, H., Zitnik, M.: Towards a unified framework for fair and stable graph representation learning. In: *Uncertainty in Artificial Intelligence*, pp. 2114–2124. PMLR (2021)
2. Agarwal, C., Lakkaraju*, H., Zitnik*, M.: Towards a unified framework for fair and stable graph representation learning (2021)
3. Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., Gallinari, P.: Learning social network embeddings for predicting information diffusion. In: *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 393–402 (2014)
4. Dai, E., Wang, S.: Say no to the discrimination: learning fair graph neural networks with limited sensitive attribute information. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 680–688 (2021)
5. Dong, Y., Liu, N., Jalaian, B., Li, J.: Edits: modeling and mitigating data bias for graph neural networks. In: *Proceedings of the ACM Web Conference 2022*, pp. 1259–1269 (2022)
6. Dong, Y., Ma, J., Wang, S., Chen, C., Li, J.: Fairness in graph mining: a survey. *IEEE Trans. Knowl. Data Eng.* (2023)
7. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 214–226 (2012)
8. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864 (2016)
9. Guerra, P., Meira Jr, W., Cardie, C., Kleinberg, R.: A measure of polarization on social media networks based on community boundaries. In: *Proceedings of the international AAAI Conference on Web and Social Media*, vol. 7, pp. 215–224 (2013)
10. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. *Adv. Neural Inform. Process. Syst.* **29** (2016)
11. He, R., McAuley, J.: Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: *Proceedings of the 25th International Conference on World Wide Web*, pp. 507–517 (2016)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
13. Liu, Z., Nguyen, T.K., Fang, Y.: On generalized degree fairness in graph neural networks. arXiv preprint [arXiv:2302.03881](https://arxiv.org/abs/2302.03881) (2023)
14. Ma, Y., Liu, X., Shah, N., Tang, J.: Is homophily a necessity for graph neural networks? arXiv preprint [arXiv:2106.06134](https://arxiv.org/abs/2106.06134) (2021)
15. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Oakland, CA, USA, vol. 1, pp. 281–297. (1967)

16. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. *Proc. IEEE* **104**(1), 11–33 (2015)
17. Rahman, T., Surma, B., Backes, M., Zhang, Y.: Fairwalk: Towards fair graph embedding (2019)
18. Shiao, W., Saini, U.S., Liu, Y., Zhao, T., Shah, N., Papalexakis, E.E.: Carl-g: clustering-accelerated representation learning on graphs. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2036–2048 (2023)
19. Takac, L., Zabovsky, M.: Data analysis in public social networks. In: *International Scientific Conference and International Workshop Present Day Trends of Innovations*, vol. 1 (2012)
20. Toneva, M., Sordoni, A., Combes, R.T.d., Trischler, A., Bengio, Y., Gordon, G.J.: An empirical study of example forgetting during deep neural network learning. *arXiv preprint [arXiv:1812.05159](https://arxiv.org/abs/1812.05159)* (2018)
21. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. *arXiv preprint [arXiv:1710.10903](https://arxiv.org/abs/1710.10903)* (2017)
22. Wang, N., Lin, L., Li, J., Wang, H.: Unbiased graph embedding with biased graph observations. In: *Proceedings of the ACM Web Conference 2022*, pp. 1423–1433 (2022)
23. Wu, J., He, J., Xu, J.: Net: degree-specific graph neural networks for node and graph classification. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 406–415 (2019)