Fair Graph Representation Learning via Diverse Mixture-of-Experts

Zheyuan Liu[1*], Chunhui Zhang[1*], Yijun Tian[2], Erchi Zhang[1], Chao Huang[3], Yanfang Ye[2], Chuxu Zhang[1†]
[1]Brandeis University, MA, USA [2]University of Notre Dame, IN, USA [3]University of Hong Kong, Hong Kong, China
{zheyuanliu,chunhuizhang,erchizhang,chuxuzhang}@brandeis.edu,chuang@cs.hku.hk,{yijun.tian,yye7}@nd.edu

ABSTRACT

Graph Neural Networks (GNNs) have demonstrated a great representation learning capability on graph data and have been utilized in various downstream applications. However, real-world data in web-based applications (e.g., recommendation and advertising) always contains bias, preventing GNNs from learning fair representations. Although many works were proposed to address the fairness issue, they suffer from the significant problem of insufficient learnable knowledge with limited attributes after debiasing. To address this problem, we develop ***G**raph-**Fa**irness **M**ixture of **E**xperts* (**G-Fame**), a novel plug-and-play method to assist any GNNs to learn distinguishable representations with unbiased attributes. Furthermore, based on G-Fame, we propose **G-Fame++**, which introduces three novel strategies to improve the representation fairness from node representations, model layer, and parameter redundancy perspectives. In particular, we *first* present the embedding diversified method to learn distinguishable node representations. *Second*, we design the layer diversified strategy to maximize the output difference of distinct model layers. *Third*, we introduce the expert diversified method to minimize expert parameter similarities to learn diverse and complementary representations. Extensive experiments demonstrate the superiority of G-Fame and G-Fame++ in both accuracy and fairness, compared to state-of-the-art methods across multiple graph datasets.

KEYWORDS

Graph Representation Learning, Mixture-of-Experts, Fairness

ACM Reference Format:
Zheyuan Liu, Chunhui Zhang, Yijun Tian, Erchi Zhang, Chao Huang, Yanfang Ye, Chuxu Zhang. 2023. Fair Graph Representation Learning via Diverse Mixture-of-Experts. In *WWW '23: The ACM Web Conference, April 30–May 4, 2023, Austin, Texas, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3543507.3583207

1 INTRODUCTION

In recent years, GNNs have gained a significant of attentions on various web-based applications, including node classification [37, 44], link prediction [56, 57], scene graph reasoning [6, 53], and recommendation system [12, 46, 47]. Most GNNs leverage message passing, a fundamental technique introduced by [16], to perform calculations and make predictions. However, message passing-based GNNs are vulnerable to sensitive attributes (e.g. race, gender, and nationality) [9, 22], which leads to unfair graph representations. Furthermore, message passing exacerbates the unfair learning given nodes aggregate sensitive attributes from their neighbors. Hence, it is necessary to come up with effective graph fairness algorithms to overcome the vulnerability issue and fairness problem in graph representation learning [8] .

Numerous works have been proposed to address the fairness problem on GNNs, where the concentrations can be mainly divided into two categories: individual fairness [14] and group fairness [3]. Individual fairness methods ensure to generate similar predictions to similar nodes [1, 33], while group fairness approaches assign equal weights to different groups so that no group receives any preference [3, 21, 23]. Later, some studies [7, 30, 49] have been proposed to solve the graph fairness problem via a dyadic approach, which requires the prediction of two groups to be completely independent of their sensitive attributes. Most of the existing algorithms construct an augmented graph by deleting biased attributes or removing prejudiced information [30, 39], resulting in limited learnable knowledge and discouraging GNNs from learning more distinguishable representations.

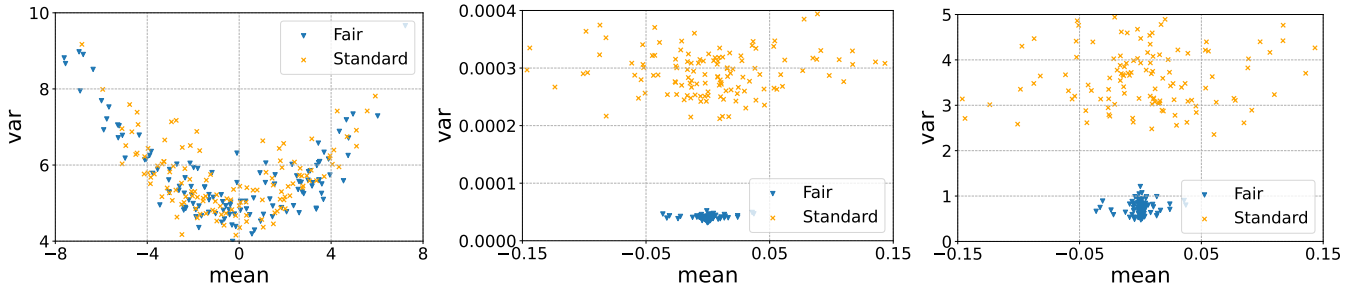
For investigating the limited learnable knowledge, we begin by comparing the statistical distributions between the latent node representations on the standard graph and the fairness-aware augmented graph. As shown in Figure 1, the statistical distributions of the input graph under standard and fairness settings are quite similar. In addition, the running variables generated by the model during fairness training are grouped together (i.e., the points are overlapped) while more diverse on standard training (i.e., points are well dispersed). Compared to the distribution of the standard setting, the batch normalization distribution under the fairness setting lacks representation diversity across different model layers, providing deficient learnable knowledge. Therefore, it is challenging to maintain the performance of graph fairness training with limited knowledge on fairness augmented graphs, compared to the standard training.

To address the challenge of the limited learnable knowledge on fairness training, we develop ***G**raph-**Fa**irness **M**ixture of **E**xperts* (**G-Fame**), a novel plug-and-play method to assist any GNNs learn distinguishable representations with unbiased attributes. In particular, G-Fame is composed of multiple expert neural networks that each contains its own parameters to learn different knowledge for diversifying node representations. In addition, to improve the

∗ The first two authors Liu and Zhang contributed equally to this research.
† The corresponding author

WWW '23, April 30–May 4, 2023, Austin, Texas, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583207



(a) Input (b) First BN Layer (c) Last BN Layer

Figure 1: Distributions of node representations generated by two GNNs trained on standard graphs and fairness-aware augmented graphs. These two distributions are remarkably similar in figure (a). However, as layer grows deeper and deeper, in figure (b) and figure (c), as the model becomes more complex, the differences between the two distributions grow.

model resistance against deficiency of learnable knowledge, we propose **G-Fame++**, in which we design three different strategies from different perspectives: (1) from node representation perspective, we introduce *embedding diversity regularization* to enable nodes to capture more different information from their neighbors during the message passing process; (2) from layer perspective, we design *layer diversity regularization* to diversify the outputs of different layers so that the shallow layers and deeper layers can obtain disparate representations; (3) from the parameter weight redundancy perspective, we present *expert weight regularization* to diversify the weight parameters of experts so that each of them can capture different information. To summarize, our contributions lie in the following aspects:

- To the best of our knowledge, this paper is the first attempt to study the deficiency of learnable information under the fairness setting. We discover that the standard and fairness-aware augmented graphs contain different statistical distributions, making it challenging for current GNNs to learn.
- To address the problem, we propose G-Fame, a novel plug-and-play method to assist any GNNs learn distinguishable representations. In addition, we propose G-Fame++ to further improve the diversity by designing three regularization methods from perspectives of node representations, model layer, and parameter redundancy.
- Extensive experiments on multiple datasets demonstrate the superiority of G-Fame and G-Fame++ over state-of-the-art methods across different AUC/accuracy and fairness metrics on fairness graph learning.

2 RELATED WORK

Graph Neural Networks. In recent years, numerous GNNs [5, 13, 19, 27, 28, 43, 54] were proposed to encode complicated graph-structured data, which utilize the message-passing mechanism to learn node representations. For instance, GAT [43] develops an attention mechanism to aggregate features from nodes with different weights. GraphSAGE [19] is a framework for inductive learning that implements an efficient aggregation function to learn node representations from neighbor nodes. DeepGCNs [28] and GCNII [5] try to alleviate the over-smoothing problem by aggregating adjacent nodes from multi-hop via residual connections. This message passing paradigm relies on node features and graph structures to learn expressive representations [24, 40, 41]. Recently, Mixture-of-Experts is also introduced into GNN for more robust graph representation [55]. However, in situations where node features contain sensitive attributes, the performance of GNNs is jeopardized by unfair predictions based on biased inputs. In this paper, we propose to enhance the model capacity in handling sensitive node attributes and producing fair predictions.

Fair Graph Representation Learning. Though fairness representation has become increasingly popular in recent years, studies under fair graphs learning are still underdeveloped [8, 45]. The majority of contemporary works attempt to resolve fairness issues on graphs via fairness-aware augmentations or adversarial training. In particular, [36] proposed Fairwalk, a random walk-based algorithm that aims to address the fairness issues in graph node embedding method node2vec [18]. [31] utilized adversarial training to minimize the marginal difference between vertex representations. Followed by that, [2, 11] focused on using GANs [17] to learn fair graph embeddings and encourage classifier assigning unbiased weight to different groups [35]. In addition, multiple fairness methods have been designed and applied to various graph applications such as fair private learning [10, 15] and fair recommendation [4, 50]. However, none of those fairness algorithms has been considered to address the insufficient learnable knowledge of graphs. Hence, we propose to enrich the learnable information from fair graph representation learning.

3 PRELIMINARIES

Fairness-Based Graph Augmentation. Let M denote a mask for the adjacency matrix. We define each element $m_{ij} \in M$ as follows:

$$m_{ij} = \begin{cases} 1 & s_i \neq s_j \quad \forall i, j \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $m_{ij} = 1$ represents that two nodes with *different* sensitive attributes are connected, whereas $m_{ij} = 0$ represents that two nodes sharing the *same* sensitive attributes are disconnected. M is utilized to mask the original adjacency matrix and encourages message passing between different minority and majority groups. As a result, GNNs trained on the modified M are able to produce more diverse and fair representations than standard graphs that are

not modified. Based on the fairness-aware mask M, a randomized response component $rr(\cdot)$ is utilized to adjust the strength of the fairness-aware adjacency modification, which can be integrated as follows:

$$rr(m_{ij}) = \begin{cases} m_{ij} & \text{with probability: } p(m_{ij}) = \frac{1}{2} + \delta \\ 1 - m_{ij} & \text{with probability: } p(1 - m_{ij}) = \frac{1}{2} - \delta \end{cases}, \tag{2}$$

where $\delta \in [0, \frac{1}{2}]$. Lastly, the unfair connections (i.e., connecting nodes with the same sensitive attributes) are dropped from the original adjacency matrix:

$$A_{fair} = A \circ rr(M), \tag{3}$$

where A_{fair} denotes the resulting matrix after dropping the unfair edges and $\circ$ indicates the Hadamard product between the original matrix A and the fairness-aware mask M. Specifically, with $rr(M)$, the algorithm promotes fairness by dropping edges between nodes with the same sensitive attributes (i.e., $s_i = s_j$). When $\delta = \frac{1}{2}$, the probability of m_{ij} is 1, which means removing all unfair connections (i.e., those connecting nodes with the same sensitive attributes) from the original graph. Consequently, regardless of the value of δ, the total amount of information in the graph is always decreasing after the mask's modification.

Fairness Training. Given a training dataset $\mathcal{D}$ and a model $f_\theta(\cdot)$ where θ denotes the model parameters. Fairness training tries to learn distinguishable fair representations under fairness constraints, which can be expressed as the following constraint optimization problem:

$$\min_{\theta} \quad \mathcal{L}(\mathcal{D}; \theta) + \lambda \|\theta\|_2^2, \quad \text{s.t. } \Omega(\mathcal{D}; \theta) < 0, \tag{4}$$

where $\mathcal{L}(\mathcal{D}; \theta)$ represents the loss of any downstream tasks (e.g., the link prediction for recommendation systems), $\|\theta\|_2^2$ denotes L_2 regularizer, and λ is a coefficient to adjust its importance. In addition, the fairness constraint $\Omega(\cdot)$ is often defined as the covariance between sensitive attributes and the signed distance of the feature vectors to the decision boundary [51, 52].

Mixture of Experts. Mixture of Experts [38] uses a gating network that decomposes a dense layer into a list of expert subnetworks, $E_1, E_2, ..., E_n$, which are trained to process each corresponding task under individual subset. A gating network is developed to select an optimal combination of the expert subnetworks based on the output of each expert. Given the input x, we denote the output of the gating network as $Q(x) = \{q_i(x)\}_{i=1}^n$ and the i-th expert output as $E_i(x)$. The output of the MoE module y can be formulated as:

$$y = \sum_{i \in \mathcal{A}} q_i(x) E_i(x), \tag{5}$$

where n denotes the number of experts and $\mathcal{A}$ indicates the set of activated top-k expert subnetworks. The gating network $Q(x)$ enables the activated experts to have the same size as the normal network, hence promoting the efficient learning of a large network. In particular, we calculate the gate value for $i-$th expert as follows:

$$q_i(x) = \frac{\exp(H(x)_i)}{\sum_{j=0}^{N} \exp(H(x)_j)}, \tag{6}$$

where $H(x)$ denotes a function to compute the weight of each expert given the current input x, and $H(x)_i, H(x)_j$ indicate the i-th and j-th value of the obtained weight of the corresponding expert in the current layer, respectively.

4 METHODOLOGY

In order to address the fairness training problem mentioned in the introduction, we first present G-Fame, a novel mechanism that can aid any GNNs in learning distinguishable representations under the fairness setting (Figure 2 (a)). Based on G-Fame, we then design G-Fame++ to comprehensively alleviate the deficiency of learnable information caused by fairness-aware augmented graphs via three regularizers, including i) an embedding diversity regularization to learn distinguishable node representations (Figure 2 (b)); ii) a layer diversity regularization to minimize the similarity between different layers (Figure 2 (c)); and iii) an expert diversity regularization to reduce expert parameter redundancy (Figure 2 (d)).

4.1 G-Fame: Graph-Fairness Mixture of Experts

The pipeline of G-Fame is shown in Figure 2 (a). G-Fame can be applied to any GNNs by substituting each GNN layer with a plug and play G-Fame layer, in order to learn distinct representations. Each G-Fame layer introduces multiple expert networks and only activates a subset of them for each input, while each expert is able to capture different aspects of knowledge and learn distinguishable representations. Specifically, given a graph $G = (V, E)$, where V is the node set and E is the edge set, we extract the node feature vector X_v for each node $v \in V$. We initialize the input feature $h_v^{(0)} = X_v$. Subsequently, in order to obtain the learned node representations, G-Fame combines the features of neighboring nodes and then aggregates them to the target node via message passing. This learning procedure can be formulated as follows:

$$h_v^{(l)} = \text{COMBINE}\left(\text{G-FAME}^{(l)}(h_v^{(l-1)}), m_v^{(l)})\right), \tag{7}$$

$$m_v^{(l)} = \text{AGGREGATE}\left(\left\{\text{G-FAME}^{(l)}(h_u^{(l-1)}), \forall u \in N(v)\right\}\right), \tag{8}$$

where $h_v^{(l)}$ represent the feature vectors of node v at l-th layer, $m_v^{(l)}$ indicates the message aggregated to node v at l-th layer, $\mathcal{N}_u$ is the set of neighbouring nodes for node u. AGGREGATE$(\cdot)$, COMBINE$(\cdot)$ are the aggregation and combination functions, respectively. In detail, the l-th G-Fame layer is consist of a set of n expert fully-connected networks $\mathcal{W}^{(l)} = \{W_i^{(l)}(\cdot)\}_{i=0}^n$ and a gating network $Q^{(l)}(\cdot) = \{q_i^{(l)}(\cdot)\}_{i=1}^n$. Then, we formulate the l-th G-Fame layer as follows:

$$\text{G-FAME}^{(l)}(h_v^{(l-1)}) = \sum_{i \in \mathcal{A}^{(l)}} q_i^{(l)}(h_v^{(l-1)}) W_i^{(l)}(h_v^{(l-1)}), \tag{9}$$

where $\mathcal{A}^{(l)}$ indicates the set of activated top-k expert networks at l-th G-Fame layer.

4.2 G-Fame++: Diversifying Representations From All Levels

To further enhance the representation diversity of G-Fame, we introduce G-Fame++ with three novel regularization-based strategies including embedding diversity regularization, layer diversity regularization, and expert diversity regularization.

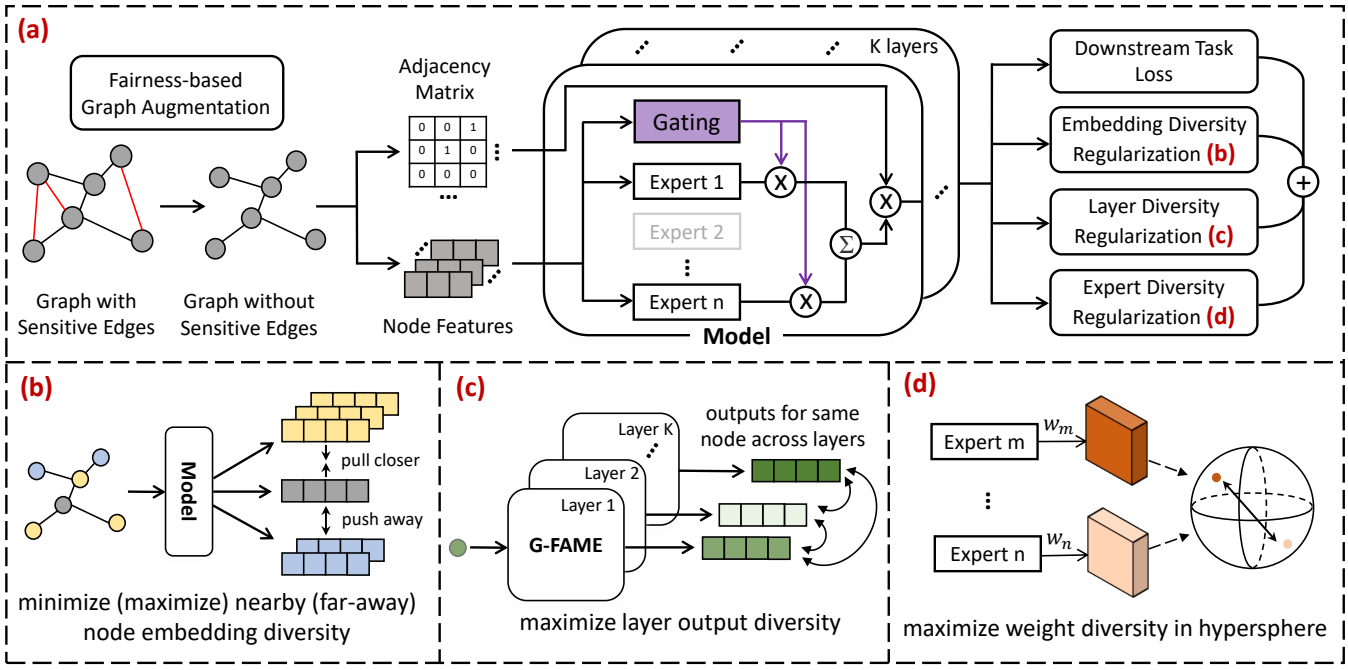


Figure 2: The overall framework of proposed methods. In (a) G-Fame++ pipeline, the graph is preprocessed with fairness-aware augmentation to drop the unfair edges and then proceeds to G-Fame layers while partial experts are activated. Then, the final output is regularized towards more diversity from three levels: In (b) embedding diversity regularization, the nodes with similar features are brought closer, whereas the nodes with different features are pulled away. (c) Layer diversity regularization maximizes the difference of different G-Fame layer output to diversify the information. Lastly, (d) expert diversity regularization enriches the weight diversity of each activated expert by maximizing the distance between each weight matrix projected in a hypersphere. The overall loss is consist of original downstream task loss regularized by (b), (c), (d) three components.

Embedding diversity regularization. To enrich the diversity of learned node embeddings, we aim to maximize the agreement between nodes that are proximate to each other (i.e., within r-hop neighborhood) while pushing away the irrelevant nodes that are far away. In particular, given a node $v_i \in V$ and a node $v_j \in \mathcal{N}(v_i)$, where $\mathcal{N}(v_i)$ is a set of r-hop neighbor nodes of node v_i, we denote the representations of node v_i and v_j as positive pair $\{z_i, z_j\}$. In addition, we randomly select a different node v_k such that k is not within the r-hop neighborhood of node v_i, i.e., $k \neq i$ and $v_k \notin \mathcal{N}(v_i)$. Next, we consider the representations of node v_i and v_k as negative pair $\{z_i, z_k\}$. Then, we bring positive pairs together while maximizing the distance between negative pairs. The procedure is formulated as follows:

$$\mathcal{L}_{ED} = -\log \frac{\sum_{v_j \in V} \exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{v_k \in V} \exp(\text{sim}(z_i, z_k)/\tau)}, \tag{10}$$

where $\mathcal{L}_{ED}$ denotes the obtained embedding diversity regularization loss, τ is the temperature parameter, and z_i, z_j are node representations of nodes v_i and v_j, respectively. The function $\text{sim}(\cdot)$ calculates the similarity between two node feature vectors, i.e., $\text{sim}(z_i, z_j) = z_i^\top z_j / (||z_i||_2 ||z_j||_2)$.

Layer diversity regularization. Due to the deficiency of learnable knowledge in fairness-aware augmentations caused by high cross-layer similarities, we design a layer diversity regularizer to diversify each layer. In particular, layer diversity regularizer maximizes the output difference across distinct layers and enlarges the discrepancy of learned information between layers:

$$r_{cosine}(z^{l_a}, z^{l_b}) = \frac{1}{|V|} \sum_{v_i \in V} \frac{|{z_i^{l_a}}^\top z_i^{l_b}|}{||z_i^{l_a}||_2 ||z_i^{l_b}||_2}, \tag{11}$$

where $r_{cosine}(z^{l_a}, z^{l_b})$ denotes the obtained cosine similarity, N is the total number of nodes, and z^{l_a}, z^{l_b} are the learned embeddings from layer l_a and layer l_b, respectively. Intuitively, similar cross-layer embeddings indicate that the model cannot diversify different layers and learn distinctive representations for each layer, resulting in poor performance. Therefore, we introduce the contrastive regularization to boost the diversity of cross-layer embeddings and further improve the model learning capability. The regularization term can be defined as follows:

$$r_{contrast}(z^{l_a}, z^{l_b}) = -\frac{1}{|V|} \sum_{v_i \in V} \log \frac{\exp({z_i^{l_a}}^\top z_i^{l_b})}{\exp({z_i^{l_a}}^\top z_i^{l_b}) + \exp({z_i^{l_a}}^\top (\frac{\sum_{j \neq i} z_j^{l_b}}{n-1}))}, \tag{12}$$

where $r_{contrast}(z^{l_a}, z^{l_b})$ is the calculated cross-layer embedding diversity. The rationale behind the contrastive regularization is that it increases the discrepancy between different layers and enforces each layer to learn unique representations. In addition, it improves

the learning capacity of each layer by pulling the same node embeddings across layers together while pushing away the embeddings of different nodes. The overall objective function of the all-layer diversity regularization $\mathcal{L}_{LD}$ is defined as:

$$\mathcal{L}_{LD} = \sum_{l_a, l_b \in L|_{a \neq b}} r_{cosine}(z^{l_a}, z^{l_b}) + r_{contrast}(z^{l_a}, z^{l_b}), \tag{13}$$

where L denotes a set of all layers in our model.

Expert diversity regularization. Although G-Fame is able to learn fairness information via a large number of experts, redundant parameters naturally exist among different experts, preventing the model from obtaining further diversified learnable information. Even worse, the limited learnable knowledge in fairness based augmented graphs induces each expert to obtain similar representations. Hence, to reduce the expert parameter redundancy, we present expert diversity regularization to maximize the difference among experts and obtain expert-wise diversified representations. Specifically, we introduce minimum hyperspherical separation (MHS) [32] to maximize the separation distance among expert weight vectors:

$$\max_{\{\hat{W}_1, ..., \hat{W}_m\} \in \mathbb{S}^{t-1}} \{\mathcal{L}_{\text{MHS}}(\hat{\mathcal{W}}) := \min_{i \neq j} \rho(\hat{W}_i, \hat{W}_j)\}, \tag{14}$$

where $\mathcal{L}_{\text{MHS}}(\cdot)$ is the separation distance between each weight vector in $\mathcal{W} = [W_1, W_2, ..., W_m]$. We define $\hat{\omega}_i = \frac{\text{vec}(W_i)}{||\text{vec}(W_i)||_2}$ which means vectorizing one expert weight matrix W_i then project it onto a unit hypersphere $\mathbb{S}^{t-1} := \{\hat{\omega} \in \mathbb{R} | \ ||\hat{\omega}||_2 = 1\}$, and $\rho(\cdot, \cdot)$ represents the shortest distance between two vertices. Accordingly, MHS benefits G-Fame from the following two aspects: 1) reducing the parameter redundancy of experts and facilitating the model to learn diversified learnable information; 2) empowering the model with better optimization and generalization ability (as shown in Figure 5). The overall loss function $\mathcal{L}_{\text{G-Fame++}}$ for G-Fame++ is the summation of the ground truth cross-entropy loss $\mathcal{L}_{GT}$, node embedding diversity regularization $\mathcal{L}_{ED}$, layer-wise diversity regularization $\mathcal{L}_{LD}$, and expert weight diversity regularization $\mathcal{L}_{MHS}$:

$$\mathcal{L}_{\text{G-Fame++}} = \mathcal{L}_{GT} + \mathcal{L}_{ED} + \mathcal{L}_{LD} + \mathcal{L}_{MHS}. \tag{15}$$

5 EXPERIMENT

In this section, we conduct extensive experiments to validate the effectiveness of G-Fame and G-Fame++. In addition, we show the ablation study, expressivity analysis, representation diversity analysis, and optimization landscape visualization to demonstrate the superiority of the proposed models in the fairness setting.

5.1 Experiment Setup

Datasets and Baseline Models. We test the performance of our methods on three benchmark graph datasets, i.e., *Cora*, *CiteSeer* and *PubMed*. The details of the datasets is shown in Appendix C.3. For baselines, we compare with general GNN models GCN [27], GAT [43], GIN [48], GraphSAGE [19] as well as graph fairness learning methods DropEdge [37] and FairDrop [39]. Besides, we apply two data augmentation techniques on general GNN models, i.e., node feature masking and edge drop.

Evaluations Metrics. We utilize AUC/accuracy and fairness metrics to evaluate our models. For AUC/accuracy metrics, we leverage accuracy and area under curve (AUC). For fairness metrics, we use *Demographic Parity* (DP) [21] and *Equalized Odds* (EO) [20]. Specifically, DP determines the dependency of model predictions on sensitive attributes. EO evaluates whether the subjects have the same true positive rates and false positive rates across protected and unprotected groups. Additional details of evaluation metrics are illustrated in Appendix B.

Implementation Details. We report the mean and standard deviation of ten independent runs with different data splits and random seeds. We use three experts in each layer and incorporate two G-Fame layers for our model design. In addition, we set learning rate to 0.01, epoches to 1000, and noisy gate rate to 0.01. We use Adam [25] to optimize the model. Both G-Fame and G-Fame++ are implemented in PyTorch and trained on NVIDIA V100 GPUs. Detailed hyperparameters are shown in Table 3 of Appendix A due to the limited space.

5.2 Overall Result Comparison

We conduct link prediction experiments to evaluate the AUC/accuracy and fairness of the proposed methods, which are reported in Table 1). According to the table, we can find that general GNNs (i.e., GCN, GAT) cannot perform well in both standard and fairness settings, with a lower ranking in AUC/accuracy and fairness metrics. GNNs with EdgeDrop have a large improvement in performance across all datasets but still fall behind in fairness metrics. On the other hand, fairness algorithms (e.g., FairAdj and GNNs + FairDrop) generally achieve better fairness than AUC and accuracy results. For example, FairAdj with $T2 = 20$ achieves satisfactory results under DP_m metric. However, the decent fairness obtained by these fairness algorithms comes with a large sacrifice on AUC/accuracy, with poor ranking compared to other baselines. Finally, we observe that G-Fame can outperform other baselines by remarkable margins, which demonstrates the effectiveness of the MoE mechanism. In addition, by incorporating the three proposed regularization strategies, G-Fame++ achieves the best overall AUC/accuracy and fairness under eight evaluation metrics, with average rankings of 1, 1, and 1.5 for *Cora*, *CiteSeer*, and *PubMed*, respectively.

5.3 Ablation Study

Since we propose G-Fame to learn distinguishable representations and present G-Fame++ with different regularization strategies to further improve the diversity, we conduct the ablation study to validate their effectiveness by answering the following questions: 1) Does G-Fame layer learn more fair features than baselines? and 2) Does G-Fame++ benefit from the proposed three regularizations? The associated results are shown in Table 2.

Does G-Fame layer learn more fair features than baselines? To answer this question, we replace all G-Fame layers with standard GCNConv layers taken from the backbone, disabling the usage of the MoE mechanism in our model. As shown in Table 2, the absence of G-Fame layer causes the model to lose a significant amount of accuracy (i.e., 5.5% on *Cora*, 12.5% on *CiteSeer*, and 4% on *PubMed*), which demonstrates the effectiveness of MoE mechanism in our model. In addition, replacing G-Fame layer leads to worse AUC/accuracy and fairness results, which further indicates the importance of G-Fame layer in facilitating the model's ability to diversify learned fair representations.

Table 1: Overall results of our proposed G-Fame, G-Fame++ with a number of baselines. Bold indicates the best performance and underline indicates the runner-up. *Performance* and $P.$ denote the performance of accuracy and AUC. *Fairness* and $F.$ denote the fairness metric of ΔDP_m, ΔEO_m, ΔDP_g, ΔEO_g, ΔDP_s, and ΔEO_s. *Avg.* of *Ranking* denote the average of the overall performance ranking and overall fairness ranking.

Method	*Performance*		*Fairness*						*Ranking*		
	Acc. ↑	AUC ↑	ΔDP_m ↓	ΔEO_m ↓	ΔDP_g ↓	ΔEO_g ↓	ΔDP_s ↓	ΔEO_s ↓	$P.$	$F.$	*Avg.*
Link prediction on *Cora*											
GCN [27]	81.0±1.1	88.0±1.0	53.5±2.4	34.8±5.0	13.6±3.2	17.7±4.1	88.3±3.3	100.0±0.0	6	6	6
GAT [43]	80.2±1.4	88.3±1.1	54.9±2.9	39.6±4.1	12.2±2.5	16.5±3.4	90.9±3.5	100.0±0.0	7	9	8
GCN+EdgeDrop [37]	82.4±0.9	90.1±0.7	56.4±2.4	36.5±4.3	12.3±2.6	15.4±3.3	90.2±2.7	100.0±0.0	3	8	5.5
GAT+EdgeDrop [37]	80.5±1.2	88.3±0.8	53.7±2.5	37.1±3.2	18.8±3.6	22.5±4.2	93.6±2.9	100.0±0.0	5	10	7.5
FairAdj$_{T2=5}$ [30]	75.9±1.6	83.0±2.2	40.7±4.1	20.9±4.3	18.4±2.8	31.9±7.1	83.8±4.9	**98.3**±7.2	9	4	6.5
FairAdj$_{T2=20}$ [30]	71.8±1.6	79.0±1.9	**32.3±2.8**	**15.8±4.3**	23.0±4.2	41.4±5.9	78.3±6.8	**98.3**±7.2	10	3	6.5
GCN+FairDrop [39]	82.4±0.9	90.1±0.7	52.9±2.5	31.0±4.9	11.8±3.2	14.9±3.7	89.4±3.4	100.0±0.0	3	5	4
GAT+FairDrop [39]	79.2±1.2	87.8±1.0	48.9±2.8	31.9±4.3	15.3±3.2	18.1±3.5	94.5±2.0	100.0±0.0	8	7	7.5
G-Fame	82.6±0.7	90.2±1.2	48.8±2.0	19.5±0.5	**10.8±0.7**	13.0±0.8	83.3±2.3	100.0±0.0	2	2	2
G-Fame++	**84.1±2.0**	**93.8**±0.8	44.1±3.6	**15.8±3.7**	12.9±2.8	**7.5±2.3**	**76.7±2.5**	100.0±0.0	1	1	**1**
Link prediction on *CiteSeer*											
GCN [27]	76.7±1.3	86.7±1.3	42.6±3.7	27.9±4.7	20.6±4.1	22.2±4.6	68.1±3.7	71.4±9.1	7	6	6.5
GAT [43]	76.3±1.4	85.6±1.9	42.4±2.8	26.4±4.1	21.1±3.8	25.4±5.6	71.3±5.7	73.4±9.9	8	8	8
GCN+EdgeDrop [37]	78.9±1.3	88.0±1.3	44.9±2.5	27.5±4.1	20.1±2.9	21.6±5.0	71.0±3.4	73.2±9.5	4	7	5.5
GAT+EdgeDrop [37]	76.3±0.9	85.6±1.0	42.6±2.5	28.4±5.0	22.2±5.1	27.6±6.3	76.7±3.0	77.5±8.8	8	10	9
FairAdj$_{T2=5}$ [30]	78.5±2.2	86.7±2.2	39.2±3.2	19.0±3.9	17.3±4.4	18.2±5.8	62.6±4.1	47.6±8.8	6	4	5
FairAdj$_{T2=20}$ [30]	74.4±2.5	82.5±2.7	**31.0**±3.1	15.6±3.0	**8.8**±3.2	19.7±6.9	56.1±3.8	43.1±7.4	10	2	6
GCN+FairDrop [39]	79.2±1.4	88.4±1.4	42.6±2.5	26.5±4.2	18.7±4.0	17.6±5.5	67.7±3.5	64.3±9.5	3	5	3
GAT+FairDrop [39]	78.2±1.1	87.1±1.1	42.9±2.2	28.3±4.3	22.5±3.4	25.9±5.2	75.3±3.2	73.4±9.1	5	9	7
G-Fame	79.8±2.3	89.4±1.7	38.6±3.1	13.5±2.4	11.6±2.2	9.4±2.7	59.1±0.7	47.8±8.6	2	3	2.5
G-Fame++	**81.5±0.6**	**91.9**±0.1	38.6±0.5	**13.0±1.7**	13.6±0.2	**8.5**±1.1	**55.1±3.4**	**42.0±1.1**	1	1	**1**
Link prediction on *PubMed*											
GCN [27]	88.0±0.4	94.5±0.2	43.9±1.2	13.2±1.4	5.0±1.7	4.9±1.7	57.3±2.0	26.2±3.6	5	4	4.5
GAT [43]	80.8±0.4	89.4±0.3	42.3±1.7	23.2±1.9	2.3±1.2	5.3±1.2	59.0±1.7	49.7±3.4	6	9	7.5
GCN+EdgeDrop [37]	88.0±0.5	94.6±0.3	43.7±1.0	12.8±0.8	6.3±0.7	6.0±1.1	57.5±1.4	26.3±2.3	4	5	4.5
GAT+EdgeDrop [37]	80.6±0.9	88.8±0.7	43.5±1.1	24.5±1.9	4.8±1.6	7.5±1.5	60.1±1.9	49.3±3.6	7	10	8.5
FairAdj$_{T2=5}$ [30]	75.5±2.5	84.1±2.2	32.3±4.7	15.9±4.7	7.3±3.0	13.8±6.2	53.4±9.9	43.2±9.5	9	7	8
FairAdj$_{T2=20}$ [30]	73.8±2.4	82.1±2.0	**28.9**±4.2	14.0±4.0	7.8±4.0	16.5±6.7	52.5±9.7	43.5±9.8	10	6	8
GCN+FairDrop [39]	88.4±0.4	94.8±0.2	42.5±0.5	12.2±0.7	5.6±1.8	5.1±0.9	55.7±1.5	26.6±2.6	3	3	3
GAT+FairDrop [39]	79.0±0.8	87.6±0.7	37.4±0.9	19.7±1.1	**2.0**±1.0	6.4±1.4	56.8±2.1	47.3±4.1	8	8	8
G-Fame	**89.4±0.7**	**95.9±0.2**	40.7±0.3	11.7±0.6	4.8±0.9	4.2±1.2	53.0±1.2	26.1±1.0	1	2	**1.5**
G-Fame++	89.2±0.4	95.6±0.07	35.9±0.03	**11.0±0.6**	2.3±0.2	**1.5**±0.04	**51.0±0.6**	**25.8±0.2**	2	1	**1.5**

Does G-Fame++ benefit from the proposed three regularizations? Since G-Fame++ contains different regularization strategies, we analyze their effectiveness by removing each of them individually and then comparing the results. From Table 2, first, we discover that the removal of node diversity regularization negatively impacts not only the AUC/accuracy but also the fairness of the G-Fame++ model, causing a 0.9% loss of accuracy and 2.4 loss of AUC. In addition, from the fairness perspective, this removal drops the average ranking of the model from 1 to 4 when applied to a variety of datasets. This indicates the effectiveness of our node diversity regularization strategy. Second, removing the layer diversity regularization results in a high similarity between different layer outputs. As a consequence, it degrades the ranking of the model from 1, 1, 1.5 to ranking 3, 2, and 3.5 on the datasets *Cora*, *CiteSeer*, and *PubMed*, respectively. This demonstrates the effectiveness of layer diversity regularization in facilitating distinct layers to produce diversified outputs. Third, the removal of expert diversity regularization results in expert weight parameter redundancy, which inhibits the experts from capturing different aspects of knowledge as well as generating fair predictions. As a result, the G-Fame model's fairness drops from ranking 1, 1, 1.5 to 3.5, 3.5, 4 on the datasets *Cora*, *CiteSeer*, and *PubMed*, respectively. This shows the efficacy of expert diversity regularization in enabling the model for learning fair representations. In general, the above comparisons about all three

Table 2: Ablation study results on Graph Fairness Learning Benchmark (i.e *Cora*, *CiteSeer*, and *PubMed*). For each dataset, we iteratively remove the three novel components contained in G-Fame and G-Fame++. Bolden represents the best performance and underline indicates the runner-up. *Performance* and *P.* denote the performance of accuracy and AUC. *Fairness* and *F.* denote the fairness metric of ΔDP_m, ΔEO_m, ΔDP_g, ΔEO_g, ΔDP_s, and ΔEO_s. *Avg.* of *Ranking* denotes the average of the overall performance ranking and overall fairness ranking.

Method	*Performance*		*Fairness*						*Ranking*		
	Acc. ↑	AUC ↑	ΔDP_m ↓	ΔEO_m ↓	ΔDP_g ↓	ΔEO_g ↓	ΔDP_s ↓	ΔEO_s ↓	*P.*	*F.*	*Avg.*
Link prediction on *Cora*											
G-Fame++	**84.1**±2.0	**93.8**±0.8	44.1±3.6	**15.8**±3.7	12.9±2.8	**7.5**±2.3	**76.7**±2.5	100.0±0.0	1	1	**1**
-w/o node diversity	83.2±1.2	91.4±0.7	52.0±2.5	20.0±5.3	12.2±2.1	12.4±0.6	86.4±0.6	100.0±0.0	3	4	3.5
-w/o layer diversity	83.2±1.0	91.3±0.8	**43.0**±1.5	16.0±3.9	**10.4**±2.5	**7.5**±1.0	80.3±0.8	100.0±0.0	4	2	3
-w/o expert diversity	83.5±0.4	93.4±0.4	50.0±1.0	19.3±2.9	13.4±1.0	14.1±1.5	87.7±1.7	100.0±0.0	2	5	3.5
G-Fame	82.6±0.7	90.2±1.2	48.8±2.0	19.5±0.5	10.8±0.7	13.0±0.8	83.3±2.3	100.0 ±0.0	5	3	4
-w/o G-Fame layer	81.0±1.1	88.0±1.0	53.5±2.4	34.8±5.0	13.6±3.2	17.7±4.1	88.3±3.3	100.0±0.0	6	6	6
Link prediction on *CiteSeer*											
G-Fame++	**81.5**±0.6	**91.9**±0.1	38.6±0.5	**13.0**±1.7	13.6±0.2	**8.5**±1.1	**55.1**±3.4	**42.0**±1.1	1	1	**1**
-w/o node diversity	80.2±0.2	90.3±0.3	40.2±1.0	13.9±0.6	15.8±1.8	10.5±1.5	62.8±2.0	45.2±15.8	3	4	3.5
-w/o layer diversity	80.0±0.4	89.6±0.4	**37.9**±0.7	14.2±0.8	13.7±1.1	8.7±0.6	55.6±2.7	45.4±4.4	4	2	2
-w/o expert diversity	81.0±0.7	91.1±0.3	39.9±0.2	18.5±2.7	14.3±1.4	11.0±0.6	62.1±2.1	49.0±0.0	2	5	3.5
G-Fame	79.8±2.3	89.4±1.7	38.6±3.1	13.5±2.4	**11.6**±2.2	9.4±2.7	59.1±0.7	47.8±8.6	5	3	4
-w/o G-Fame layer	76.7±1.3	86.7±1.3	42.6±3.7	27.9±4.7	20.6±4.1	22.2±4.6	68.1±3.7	71.4±9.1	6	6	6
Link prediction on *PubMed*											
G-Fame++	89.2±0.4	95.6±0.07	**35.9**±0.03	**11.0**±0.6	**2.3**±0.2	**1.5**±0.04	**51.0**±0.6	**25.8**±0.2	2	1	**1.5**
-w/o node diversity	87.4±0.1	94.2±0.1	50.1±0.4	16.1±0.4	8.6±0.1	7.5±0.3	63.0±0.3	37.2±2.5	5	4	4.5
-w/o layer diversity	87.5±0.4	94.1±0.1	41.3±0.6	12.4±0.3	4.6±0.3	4.8±0.3	55.1±1.2	28.3±0.4	5	2	3.5
-w/o expert diversity	88.8±0.03	95.3±0.1	45.5±0.2	15.7±0.9	6.3±0.9	6.4±0.4	60.2±1.4	36.6±3.5	3	5	4
G-Fame	**89.4**±0.7	**95.9**±0.2	40.7±0.3	11.7±0.6	4.8±0.9	4.2±1.2	53.0±1.2	26.1±1.0	1	3	**1.5**
-w/o G-Fame layer	88.0±0.4	94.5±0.2	43.9±1.2	13.2±1.4	5.0±1.7	4.9±1.7	57.3±2.0	26.2±3.6	4	6	5

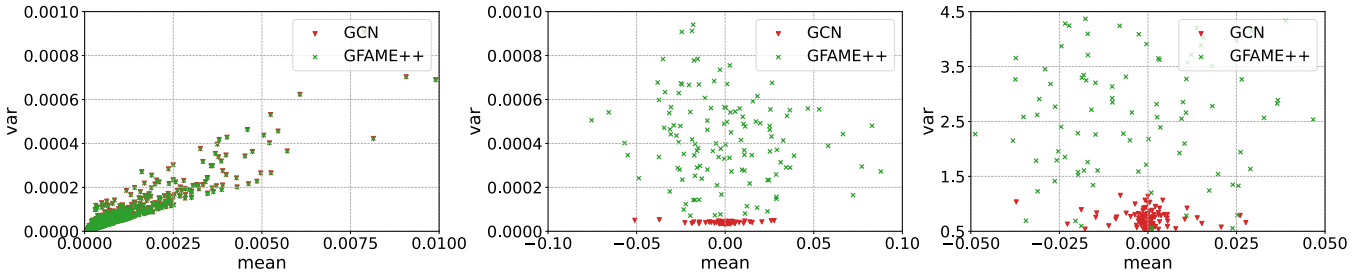
regularizations for different perspectives show that G-Fame++ improves the accuracy and fairness metrics of GNNs against limited learnable information by generating diverse yet useful representations. This is also compatible with the subsequent additional studies in Section 5.4-5.6.

5.4 Expressivity Analysis w.r.t. Fairness

We compare the expressivity of G-Fame++ and vanilla GCN by showing the distributions of node representations under fairness setting in Figure 3. Specifically, we visualize the distributions of node representations from the input, the first Batch Normalization (BN) layer, and the last BN layer. The subfigure 3a shows the distributions of the same input data for both GCN and G-Fame++, with uniform distributions to ensure fair comparison. According to subfigure 3b and subfigure 3c, we observe that GCN and G-Fame++ learn divergent node representation distributions during training. Specifically, in the first BN layer, GCN learns grouped node representation distribution, while the distribution is more scattered and spread out for G-Fame++. As the layer grows deeper, GCN generates a more diversified representation distribution in the last BN layer, but still performs much worse compared to G-Fame++. Generally, G-Fame++ can well capture the limited knowledge and obtain distinguishable representation distributions across layers. This phenomenon is further validated by comparing GCN with G-Fame, as shown in Figure 6 in Appendix A.1.

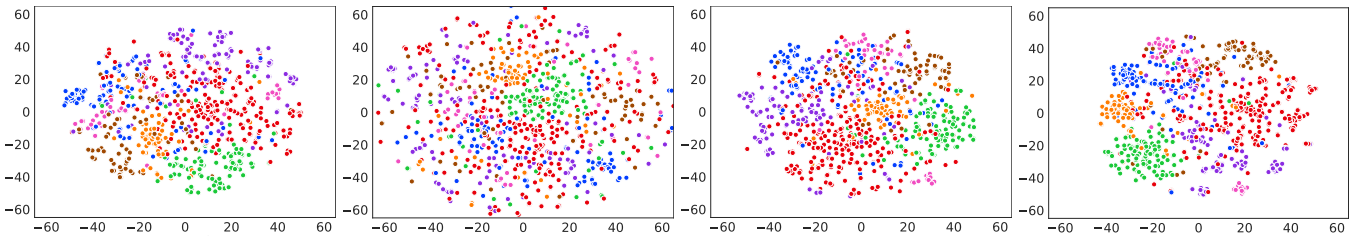
5.5 Representation Diversity Analysis

For a better understanding and comparison, we visualize the learned node representations of GCN under standard setting, GCN under fairness setting, G-Fame, and G-Fame++ via t-SNE [42]. As can be seen in Figure 4, GCN under standard setting can roughly cluster nodes from different categories while the boundaries between categories are vague. Compared to the standard setting, GCN under the fair setting shows a poor performance on node clustering (i.e., nodes are mixed and unorganized). This demonstrates that GCN cannot fully capture the knowledge from the fairness-aware augmented graph and obtain distinguishable node representations. However, both of our models G-Fame and G-Fame++ are able to distinguish and separate nodes of different categories as well as maintain a clear boundary between each category. Furthermore, nodes within the same category can form a condensed cluster instead of splitting into different small groups. This again shows the effectiveness of G-Fame and G-Fame++ on learning distinguishable node representations under the fairness setting. This observation across different layers is further discussed in Appendix A.2.



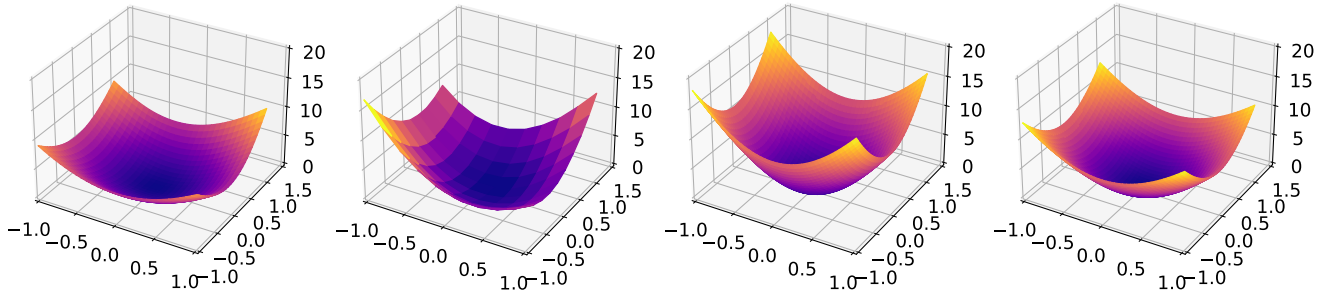
(a) Input (b) First BN layer (c) Last BN layer

Figure 3: The distributions of node representations in both GCN and our G-Fame++: input layer (*figure (a)*), after first layer (*figure (b)*), and after the last layer (*figure (c)*). Both models have the same input. Red color indicates the GCN under *fairness* setting, while green color indicates the GCN with G-Fame++ setting. The x and y axes represent the running mean and running variance of a channel, respectively.



(a) GCN (standard setting) (b) GCN (fairness setting) (c) G-Fame (d) G-Fame++

Figure 4: The t-SNE visualization of node embeddings in the final GNN layer on *Cora* dataset. Different colors denote different node class labels.



(a) GCN (standard setting) (b) GCN (fairness setting) (c) G-Fame (d) G-Fame++

Figure 5: Loss landscapes of GCN under standard setting (a), fairness setting (b) and G-Fame (c), G-Fame++ (d). Under both settings, we visualize the same set of nodes randomly sampled from the test set of *cora* dataset.

5.6 Optimization Landscape Visualization

To further show the effectiveness of our models, we visualize the 3D loss landscapes [29] of G-Fame, G-Fame++, and GCN under two settings (i.e., standard and fairness). According to Figure 5, we observe that the loss landscape for GCN under the standard setting is more smooth than it under the fairness setting, which shows the difficulty of optimizing the model under the fairness setting. On the other hand, the loss landscape for G-Fame is steeper than GCN under both settings. We ascribe the reason to the large volume of parameters contained in each expert, which significantly increases the difficulty of optimizing the model. However, G-Fame++ can learn a much more smooth landscape compared to G-Fame, thanks to the proposed three regularization strategies. This reveals that our regularizations can alleviate the optimization difficulty induced by fairness-aware augmented graphs, which further demonstrates the efficacy of our overall framework.

6 CONCLUSION

In this paper, we identify the problem of limited learnable information problem in graph fairness learning. To address this problem, we present G-Fame, a novel plug-and-play method for assisting any GNNs to learn distinguishable representations. Specifically, G-Fame introduces an MoE mechanism that utilizes multiple expert neural networks to capture different aspects of knowledge in the fairness setting. In addition, we propose G-Fame++ with three innovative regularization strategies to further increase the diversity from perspectives of node representations, layers, and experts. Extensive experiments and in-depth studies demonstrate the superiority of

G-Fame and G-Fame++ across a variety of accuracy and fairness metrics on multiple benchmark datasets.

ACKNOWLEDGMENTS

This work is partially supported by the NSF under grants CMMI-2146076 and Brandeis University. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

REFERENCES

[1] Yahav Bechavod, Christopher Jung, and Steven Z Wu. 2020. Metric-free individual fairness in online learning. In *NeurIPS*.
[2] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075* (2017).
[3] Reuben Binns. 2020. On the apparent conflict between individual and group fairness. In *FAccT*.
[4] Robin Burke. 2017. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093* (2017).
[5] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. [n. d.]. Simple and deep graph convolutional networks. In *ICML*.
[6] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. 2020. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *CVPR*. 9962–9971.
[7] Sean Current, Yuntian He, Saket Gurukar, and Srinivasan Parthasarathy. 2022. FairMod: Fair Link Prediction and Recommendation via Graph Modification. *arXiv preprint arXiv:2201.11596* (2022).
[8] Enyan Dai and Suhang Wang. 2021. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *WSDM*.
[9] Yushun Dong, Song Wang, Yu Wang, Tyler Derr, and Jundong Li. 2022. On Structural Explanation of Bias in Graph Neural Networks. In *KDD*.
[10] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *ITCS*.
[11] Harrison Edwards and Amos Storkey. 2015. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897* (2015).
[12] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*.
[13] Yujie Fan, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. 2022. Heterogeneous temporal graph neural network. In *SDM*.
[14] Will Fleisher. 2021. What's Fair about Individual Fairness?. In *AAAI*.
[15] Arpita Ghosh and Robert Kleinberg. 2016. Inferential privacy guarantees for differentially private mechanisms. *arXiv preprint arXiv:1603.01508* (2016).
[16] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *ICML*.
[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* (2020).
[18] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.
[19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*.
[20] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *NeurIPS*.
[21] Zhimeng Jiang, Xiaotian Han, Chao Fan, Fan Yang, Ali Mostafavi, and Xia Hu. 2022. Generalized Demographic Parity for Group Fairness. In *ICLR*.
[22] Jian Kang and Hanghang Tong. 2021. Fair graph mining. In *ICDM*.
[23] Jian Kang, Tiankai Xie, Xintao Wu, Ross Maciejewski, and Hanghang Tong. 2021. MultiFair: Multi-Group Fairness in Machine Learning. *arXiv preprint arXiv:2105.11069* (2021).
[24] Shima Khoshraftar and Aijun An. 2022. A Survey on Graph Representation Learning Methods. *arXiv preprint arXiv:2204.01855* (2022).
[25] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
[26] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
[27] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
[28] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019. Deepgcns: Can gcns go as deep as cnns?. In *ICCV*.
[29] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *NeurIPS*.
[30] Peizhao Li, Yifei Wang, Han Zhao, Pengyu Hong, and Hongfu Liu. 2021. On dyadic fairness: Exploring and mitigating bias in graph connections. In *ICLR*.
[31] Peiyuan Liao, Han Zhao, Keyulu Xu, Tommi Jaakkola, Geoffrey J Gordon, Stefanie Jegelka, and Ruslan Salakhutdinov. 2021. Information obfuscation of graph neural networks. In *ICML*.
[32] Weiyang Liu, Rongmei Lin, Zhen Liu, Li Xiong, Bernhard Schölkopf, and Adrian Weller. 2021. Learning with hyperspherical uniformity. In *AISTATS*.
[33] Sepideh Mahabadi and Ali Vakilian. 2020. Individual fairness for k-clustering. In *ICML*.
[34] Farzan Masrour, Tyler Wilson, Heng Yan, Pang-Ning Tan, and Abdol Esfahanian. 2020. Bursting the filter bubble: Fairness-aware network link prediction. In *AAAI*.
[35] Dana Pessach and Erez Shmueli. 2022. A Review on Fairness in Machine Learning. *Comput. Surveys* (2022).
[36] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards fair graph embedding. In *IJCAI*.
[37] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *ICLR*.
[38] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *ICLR*.
[39] Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. 2021. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Transactions on Artificial Intelligence* (2021).
[40] Yijun Tian, Kaiwen Dong, Chunhui Zhang, Chuxu Zhang, and Nitesh V Chawla. 2023. Heterogeneous Graph Masked Autoencoders. In *AAAI*.
[41] Yijun Tian, Chuxu Zhang, Zhichun Guo, Xiangliang Zhang, and Nitesh Chawla. 2023. Learning MLPs on Graphs: A Unified View of Effectiveness, Robustness, and Efficiency. In *ICLR*.
[42] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* (2008).
[43] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.
[44] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR*.
[45] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. 2021. Learning fair representations for recommendation: A graph-based perspective. In *WWW*.
[46] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2020. Graph neural networks in recommender systems: a survey. *Comput. Surveys* (2020).
[47] Lianghao Xia, Chao Huang, and Chuxu Zhang. 2022. Self-supervised hypergraph transformer for recommender systems. In *KDD*.
[48] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*.
[49] Moyi Yang, Junjie Sheng, Xiangfeng Wang, Wenyan Liu, Bo Jin, Jun Wang, and Hongyuan Zha. 2022. Obtaining Dyadic Fairness by Optimal Transport. *arXiv preprint arXiv:2202.04520* (2022).
[50] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. In *NeurIPS*.
[51] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*.
[52] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. 2019. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research* (2019).
[53] Chunhui Zhang, Chao Huang, Youhuan Li, Xiangliang Zhang, Yanfang Ye, and Chuxu Zhang. 2022. Look Twice as Much as You Say: Scene Graph Contrastive Learning for Self-Supervised Image Caption Generation. In *CIKM*.
[54] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *KDD*.
[55] Chunhui Zhang, Yijun Tian, Mingxuan Ju, Zheyuan Liu, Yanfang Ye, Nitesh Chawla, and Chuxu Zhang. 2023. Chasing All-Round Graph Representation Robustness: Model, Training, and Optimization. In *ICLR*.
[56] Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. 2020. Few-shot knowledge graph completion. In *AAAI*.
[57] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *NeurIPS*.

Appendix A ADDITIONAL EXPERIMENTS

A.1 Expressivity of G-Fame w.r.t Fairness

The statistics of the batch normalization layer for G-Fame are shown in Figure 6. Here we compare the representations of G-Fame and vanilla GCN under the fairness setting as a complementary for experiments. Specifically, we visualize the distributions of node representations from the input, the first Batch Normalization (BN) layer, and the last BN layer. Subfigure 6a demonstrates the distributions of the uniform input data for both GCN and G-Fame, with same distributions to guarantee the fair comparison. Figure 6 shows a divergent distribution between G-Fame, and GCN under fairness setting. According to Subfigure 6b and Subfigure 6c, we find out that GCN and G-Fame learn disparate node representation distributions during training. In particular, in the first BN layer, GCN learns mixed node representation distribution, while the BN distribution is more scattered and spread out for G-Fame. As the layer depth increases, GCN produces a more diversified representation distribution in the last BN layer, but still performs not as good as G-Fame. Generally, G-Fame can still maintain an outstanding ability to produce distinguishable representations across layers.

A.2 Representation Diversity across Layers

We visualize the performance of G-Fame, G-Fame++, and vanilla GCN under the fairness setting via the t-SNE visualization [42], which is demonstrated in Figure 7. In particular, we show the visualizations of two layers for each model, where layer 1 is the beginning input layer and layer 2 indicates the next proceeding layer after layer 1. According to Figure 7, vanilla GCN under fairness setting illustrates a poor performance on node classification tasks due to its mixed and clustered nodes. In addition, GCN under the fairness setting is having a hard time congregating nodes from different categories while the boundary between each category is pretty vague. This phenomenon is alleviated as the layer depth increases but still not satisfying enough. On the other hand, both of our models G-Fame and G-Fame++ are able to distinguish and separate nodes of different categories while maintaining a clear boundary between each category as layer depth increases, which can also be validated in previous experiment 4. Furthermore, nodes under the same category can form a condensed group instead of splitting into different small clusters. This again shows the effectiveness of G-Fame and G-Fame++ on learning distinguishable node representations under the fairness setting.

Appendix B EVALUATION METRICS

In section 5.1, we briefly demonstrate the general fairness metrics we used in this paper to evaluate and compare the performance of G-Fame and G-Fame++ with other baselines. Let us denote $Y \in [0, 1]$ as a binary target variable and $\hat{Y} = f(x)$ as a predictor. Next, we pair each x with a categorical sensitive attribute A. Two often used metrics under a such case are *Demographic Parity* (DP) [21] and *Equalized Odds* (EO) [20].

Demographic Parity (DP): $\hat{Y}$ satisfies DP if the positive outcome is independent of the value of the sensitive attribute A, such that:

$$P(\hat{Y}|A = 0) = P(\hat{Y}|A = 1). \tag{16}$$

If this is shown on a confusion matrix, it requires the positive rate of every part of the protected group to be the same.

Equalized Odds (EO): $\hat{Y}$ satisfies EO if the true positive rates and false positive rates between two groups match with each other with different values of sensitive attribute A:

$$P(\hat{Y} = 1|A = 0, Y = y) = P(\hat{Y} = 1|A = 1, Y = y). \tag{17}$$

For the link prediction task, we focus more on the dyadic fairness metric such that it requires model predictions to be statistically independent of sensitive attributes corresponding to the edges. In [34, 39], the authors proposed three dyadic criteria: mixed dyadic-level protection, group dyadic-level protection, and subgroup dyadic-level protection. Specifically, the fairness in mixed dyadic is determined based on the homophily of the nodes interconnected by each link; the fairness in subgroup dyadic ensures that no subgroups gain unfair advantages in the formation of links. Group dyadic ensures that every node is involved in link creation regardless of the value of their sensitive attributes. Here, we provide more detailed explanations for these metrics:

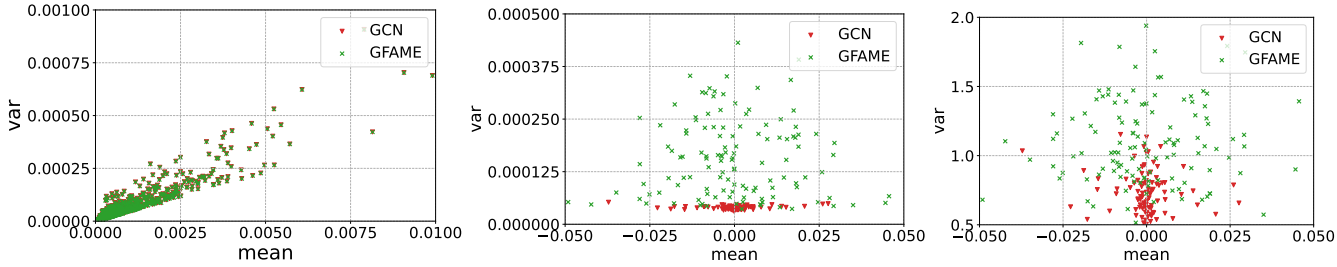
- **Mixed dyadic** [34]: the fairness is evaluated based on the homogeneity of nodes involved in each edge. In particular, the edge is considered to be an intra-group link if it interconnects a pair of nodes with the same sensitive attribute. Otherwise, it is regarded as an inter-group link. This evaluation metric usually appears in the recommender system to prevent segregation of the users.
- **Sub-group dyadic** [34]: the fairness is evaluated based on how representative a subgroup is in the creation of the links (i.e., intra-group and inter-group). In other words, the subgroup dyadic fairness metric aims to protect the balance between all possible intra-group links and inter-group links. It makes sure that no certain subgroup is favored over other subgroups.
- **Group dyadic** [39]: there is an injective mapping between the node-level and dyadic groups. The group dyadic metric ensures that each node gets involved in the links' creation process whether the value of their sensitive attributes.

Appendix C IMPLEMENTATION DETAILS

C.1 Baseline Descriptions

To evaluate the performance of our pipelines on the link prediction task, we compare our models with multiple general GNNs and fairness algorithms to further display their effectiveness:

- **GCN** [27]: a neural network model that implements with layer-wise propagation rule based on a first-order approximation of spectral graph convolutions. (code).
- **GAT** [43]: a convolution-style neural network that leverages masked self-attentional layers to assign importance to different nodes within a neighborhood without depending on the entire graph structure (code).
- **FairAdj** [30]: learns a fair adjacency matrix during the link prediction task. The algorithm implements a graph variational autoencoder [26] and two distinct optimization methods to reach a more favorable fairness-utility tradeoff. In particular, one optimization process is for obtaining a fair version of the adjacency matrix while the other one is for an end-to-end link prediction task (code.



(a) Input (b) First BN Layer (c) Last BN Layer

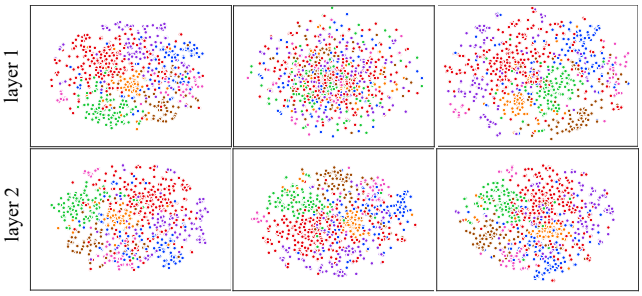
Figure 6: The statistics of the channel-based batch normalization (BN) layer: before transformed by the very first GCN as shown in Figure (a), after the first layer as in Figure (b) shown, and after the last layer as Figure (c) shown. Red color indicates the GCN under *fairness* setting, while green color indicates the GCN with G-Fame setting. The x and y axes represent the running mean and running variance of a channel, respectively.

Table 3: Hyper-parameters of G-Fame and G-Fame++ for *cora*, *citeseer* and *pubmed* datasets. The n and k indicate the number of total experts and activated experts in each layer, respectively. The noisy rate controls the randomness when some expert is activated by the gate module. δ regulates the level of fairness in our model during the training process.

Model	G-Fame			G-Fame++		
Dataset	Cora	CiteSeer	PubMed	Cora	CiteSeer	PubMed
Iteration	500	500	200	500	500	200
Learning Rate	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
Output Dimension	256	256	128	256	256	128
Hiddent Dimension	256	256	128	256	256	128
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
n	3	3	2	3	3	2
k	1	1	1	1	1	1
Dropout	0.5	0.5	0.5	0.5	0.5	0.5
Noisy Rate	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2
δ	0.46	0.50	0.46	0.46	0.46	0.46

Table 4: Statistics of three academic network datasets

Dataset	#Nodes	#Edges	#Feat.	#Classes	#Avg. Degree	Feat. Range (original)	Download links
Cora	2,708	10,556	1,433	7	3.88	[-2.30, 2.40]	https://shorturl.at/bhoY4
Citeseer	3,327	9,104	3,703	6	2.84	[-4.55, 1.67]	https://shorturl.at/bGTZ6
PubMed	19,717	88,648	500	3	4.50	[-4.55, 1.67]	https://shorturl.at/cnEN8



(a) GCN (b) G-FAME (c) G-FAME++

Figure 7: The t-SNE visualization of node embeddings in layer 1 and 2 on fairness-aware augmented *Cora*. Different colors denotes different node category labels.

- **DropEdge** [37]: a data-augmented technique that alleviates over-fitting problems and reduces the information loss caused by over-smoothing during the training process. It randomly removes a number of edges from the input graph during each training epoch (code).
- **FairDrop** [39]: improve fairness in graph representation learning via dropping biased edges. It can also be considered a biased data augmentation technique that can be applied to various datasets and models. The fairness of the algorithm is evaluated based on two tasks: the end-to-end link prediction task and the capability of removing the effect of sensitive attributes from node representations (code).

C.2 Hyperparameters

The hyper-parameters of G-Fame and G-Fame++ are listed in Table 3. Due to the limitation of GPU memory, the output and hidden dimensions for both models on the PubMed dataset are restricted to 128. Similarly, the gating module assigns no more than two experts ($n \leq 2$) in each layer.

C.3 Dataset Details

We evaluate our proposed G-Fame++ and G-Fame++ under graph fairness learning settings on three real-world citation networks. The data statistics are displayed in Table 4.

C.4 Efficiency Analysis

The efficiency of G-FAME++ is comparable to regular GNN. The complexity of G-FAME++ relies on the activated part, which is a relatively small number, thus the computation complexity of G-FAME++ will not grow rapidly like regular GNN as the hidden dimension increases. For example, on Cora, the training time of GCN (dim=32 | dim=256) is (5.2 | 8.5) minutes; meanwhile, G-FAME++ (dim=32 and expertnum=4 | dim=32 and expertnum=8) costs (5.5 | 6) minutes.

C.5 Parameter Comparison

The parameters amount for G-FAME++ (expertnum=4) and existing GCN baseline is 12.0M and 2.2M, respectively. However, during the real training and inference phase, G-FAME++ activates the same number of parameters as the GCN baseline.