Graph Fairness Learning under Distribution Shifts

Yibo Li
Beijing University of Posts and Telecommunications
Beijing, China
yiboL@bupt.edu.cn

Xiao Wang[†]
Beihang University
Beijing, China
xiao_wang@buaa.edu.cn

Yujie Xing
Beijing University of Posts and Telecommunications
Beijing, China
yujie-xing@bupt.edu.cn

Shaohua Fan
Tsinghua University
Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education(Beijing Jiaotong University)
Beijing, China
fanshaohua@bupt.cn

Ruijia Wang
Beijing University of Posts and Telecommunications
Beijing, China
wangruijia@bupt.edu.cn

Yaoqi Liu
Beijing University of Posts and Telecommunications
Beijing, China
yaoqiliu@bupt.edu.cn

Chuan Shi[†]
Beijing University of Posts and Telecommunicationsy
Beijing, China
shichuan@bupt.edu.cn

ABSTRACT

Graph neural networks (GNNs) have achieved remarkable performance on graph-structured data. However, GNNs may inherit prejudice from the training data and make discriminatory predictions based on sensitive attributes, such as gender and race. Recently, there has been an increasing interest in ensuring fairness on GNNs, but all of them are under the assumption that the training and testing data are under the same distribution, i.e., training data and testing data are from the same graph. *Will graph fairness performance decrease under distribution shifts? How does distribution shifts affect graph fairness learning?* All these open questions are largely unexplored from a theoretical perspective. To answer these questions, we first theoretically identify the factors that determine bias on a graph. Subsequently, we explore the factors influencing fairness on testing graphs, with a noteworthy factor being the representation distances of certain groups between the training and testing graph. Motivated by our theoretical analysis, we propose our framework FatraGNN. Specifically, to guarantee fairness performance on unknown testing graphs, we propose a graph generator to produce numerous graphs with significant bias and under different distributions. Then we minimize the representation distances for each certain group between the training graph and generated graphs. This empowers our model to achieve high classification and fairness performance even on generated graphs with significant bias, thereby effectively handling unknown testing graphs. Experiments on real-world and semi-synthetic datasets demonstrate the effectiveness of our model in terms of both accuracy and fairness.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Networks → Network algorithms**.

KEYWORDS

Graph Neural Networks; Fairness; Distribution Shifts

ACM Reference Format:
Yibo Li, Xiao Wang[†], Yujie Xing, Shaohua Fan, Ruijia Wang, Yaoqi Liu, and Chuan Shi[†]. 2024. Graph Fairness Learning under Distribution Shifts. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3589334.3645508

1 INTRODUCTION

Graph Neural Networks (GNNs) are powerful deep learning algorithms that can be used to model graph-structured data. In recent years, there have been enormous successful applications of GNNs on various areas such as social media mining [18, 27, 39, 45, 46],

[†]Corresponding authors.

This work is supported in part by the National Natural Science Foundation of China (No.U20B2045, 62192784, U22B2038, 62002029, 62172052, 62322203, 62172052), Foundation of Key Laboratory of Big Data Artificial Intelligence in Transportation(Beijing Jiaotong University), Ministry of Education (No.BATLAB202301), China Postdoctoral Science Foundation (No.2023M741946) and China Postdoctoral Researcher Program (No.GZB20230345).

WWW '24, May 13–17, 2024, Singapore, Singapore.

ACM ISBN 979-8-4007-0171-9/24/05
https://doi.org/10.1145/3589334.3645508

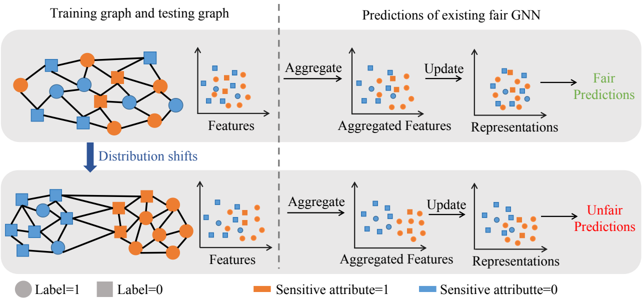

Figure 1: Toy example of fairness under distribution shifts on graphs.

drug discovery [21], and recommender system [5, 44]. However, despite their success, there is a growing concern that GNNs may inherit or even amplify discrimination and social bias from the training data, leading to unfair treatment of sensitive groups with sensitive attributes such as gender, age, region, and race. This may result in social and ethical issues, thus limiting the application of GNNs in critical areas such as job marketing [22], criminal justice [35], and credit scoring [15].

To mitigate this issue, many fair GNNs [6, 8, 10, 28, 42, 47] have been proposed. They improve graph fairness by adding a fairness-related regularization term to the optimization objective [42, 47], adopting adversarial learning to learn fairer node representations [6, 8], debiasing the graph itself [10, 28], etc. Despite the success of fair GNNs, they are all proposed under the common hypothesis that the training and testing data are under identical distribution, which does not always hold in reality.

In real-world contexts, distribution shifts frequently occurs [2, 4, 26, 41] and can adversely affect the fairness performance of existing fair GNNs. This is exemplified in Figure 1, where a fair GNN designed for job recommendation is trained on a social network from one state and subsequently applied to a network from another state. In both social networks, race serves as the sensitive attribute, and the label is whether to recommend the job. However, the two graphs are under different distributions. Specifically in the testing graph, there are larger feature differences between different sensitive groups, and nodes within the same sensitive group are more likely to be connected. After the feature aggregation step, the aggregated features of nodes within the same sensitive group will be more homophilous, and nodes in different sensitive groups will be even more distinguishable. As it is easy to recognize the sensitive attributes of the nodes, the fair GNN relies more on this information to make predictions on the testing graph, resulting in discrimination such as disproportionately recommending low-payment jobs to certain sensitive groups identified by race.

Although distribution shifts can lead to unfairness, previous studies [12–14, 41] mainly aim at keeping stable classification performance of GNNs under distribution shifts, while largely ignoring the fairness issue. *Why might graph fairness deteriorate under distribution shifts? How does distribution shifts affect the fairness of GNNs?* The answers from a theoretical and methodological perspective remain largely unknown.

Recently, the topic of fairness learning under distribution shifts has received considerable attention [3, 20, 33]. However, all these works focus on Euclidean data, overlooking the vital structural information in graphs. Such information helps in making accurate predictions but runs the risk of amplifying the data bias, therefore requiring additional careful consideration. In this work, we first theoretically analyze the relationship between graph data distribution and graph fairness (Theorem 3.6), and conclude that graph fairness is determined by a sensitive structure-property and the feature difference between different sensitive groups. This insight sheds light on the potential deterioration of graph fairness due to distribution shifts. Then we prove that fairness on the testing graph depends on two key factors: fairness on the training graph and the representation distances of certain groups (nodes with the same label and sensitive attribute) between the training graph and the testing graph (Theorem 3.8). These findings well deepen our understanding of graph fairness learning under distribution shifts.

Motivated by our theoretical insights, we further propose a novel model called FatraGNN to handle this issue. Our model employs an adversarial module to ensure fairness on the training graph. As the testing graphs are unknown, we draw inspiration from previous research [41] that generates graphs under various distributions, and subsequently trains GNN on them to bolster the classification performance on unknown testing graphs. Similarly, we also utilize a graph generation module to generate graphs with significant bias and under different distributions. Then we utilize an alignment module to minimize the representation distances of each certain group between the training graph and the generated graphs. If our model can learn fair representations for these generated graphs with large bias, it will be more robust to distribution shifts and effectively deal with specific testing graphs which usually have smaller bias. In summary, our contributions are three-fold:

- To the best of our knowledge, this is the first attempt to study graph fairness learning under distribution shifts from a theoretical perspective. We theoretically analyze the relationship between graph fairness and graph data distribution and discover the key factors that affect fairness learning under distribution shifts.
- Based on the theoretical insights, we propose our FatraGNN, which consists of an adversarial debiasing module, a graph generation module, and an alignment module, to ensure fairness on the unknown testing graphs.
- Extensive experiments show that our FatraGNN outperforms state-of-the-art baselines under distribution shifts in terms of both classification and fairness performance on real-world and semi-synthetic datasets.

2 PRELIMINARIES AND NOTATIONS

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be a graph with n nodes, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. $\mathbf{X} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{n \times \zeta}$ represents the node feature matrix, where x_i is the feature vector of node v_i and ζ is the dimension of node features. Graph structure of $\mathcal{G}$ can be described by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, and $\mathbf{A}_{ij} = 1$ iff there exists an edge between nodes v_i and v_j. The diagonal degree matrix is denoted as $\mathbf{D} = \mathrm{diag}\,(d_1, \cdots, d_n)$, where $d_i = \sum_j \mathbf{A}_{ij}$. Node sensitive attributes are specified by the t-th channel of $\mathbf{X}$, i.e., $\mathbf{F} = \mathbf{X}_{:,t} = [f_1, f_2, \dots, f_n] \in \{0, 1\}^n$, where f_i is the sensitive attribute of node i. Here we focus on binary classification tasks, and the binary labels of the nodes are denoted by $\mathbf{Y} \in \{0, 1\}^n$.

Fairness Metric There exist several different definitions of fairness, such as group fairness [29], individual fairness [11], counterfactual fairness [25], and degree-related fairness [38]. In this work, we focus on group fairness, and use two commonly used metrics to measure it: equalized oods [19] and demographic parity [7]. Equalized odds seeks to achieve the same true positive rate and true negative rate between two sensitive groups, which is defined as $\Delta_{EO} = \frac{1}{2}\sum_{y=0}^{1} |\mathbb{E}_{v_i\in\mathcal{V}}(\hat{y}_i = y|y_i = y, f_i = 1) - \mathbb{E}_{v_j\in\mathcal{V}}(\hat{y}_j = y|y_j = y, f_j = 0)|$, where $\hat{y}_i$ is the predicted label of v_i, and y_i is the ground-truth label of v_i. Demographic parity measures the acceptance rate difference between two sensitive groups. For example, in binary classification tasks such as deciding whether a student should be admitted into a university or not, demographic parity is considered to be achieved if the model yields the same acceptance rate for individuals in both sensitive groups. It is defined as $\Delta_{DP} = |\mathbb{E}_{v_i\in\mathcal{V}}(\hat{y}_i = 1 \mid f_i = 1) - \mathbb{E}_{v_j\in\mathcal{V}}(\hat{y}_j = 1 \mid f_j = 0)|$.

Fairness on Graph Distribution Shifts Following the definition of previous study [41], we characterize the data generation process as $\mathbb{P}(\mathbf{A}, \mathbf{X}, \mathbf{Y}|\mathbf{e}) = \mathbb{P}(\mathbf{A}, \mathbf{X}|\mathbf{e})\mathbb{P}(\mathbf{Y}|\mathbf{A}, \mathbf{X}, \mathbf{e})$, where $\mathbf{e}$ represents a random variable denoting the latent environmental factors that influence the data distribution. First, the graph is generated via $\mathbb{P}(\mathbf{A}, \mathbf{X}|\mathbf{e})$. Then the labels are generated via $\mathbb{P}(\mathbf{Y}|\mathbf{A}, \mathbf{X}, \mathbf{e})$. We assume that $\mathbb{P}(\mathbf{Y}|\mathbf{A}, \mathbf{X}, \mathbf{e})$ is invariant under different environments. Our aim is to achieve a scenario where the generation of $\mathbf{Y}$ is not influenced by the features related to sensitive attributes $\mathbf{F}$ to ensure fairness. We consider training graph from data distribution $\mathbb{P}(\mathbf{A}_{\mathcal{J}}, \mathbf{X}_{\mathcal{J}}|\mathbf{e} = \mathcal{J})$, testing graphs from data distribution $\mathbb{P}(\mathbf{A}_{\mathcal{K}}, \mathbf{X}_{\mathcal{K}}|\mathbf{e} = \mathcal{K})$. This work intends to ensure fairness when $\mathcal{J} \neq \mathcal{K}$.

3 GRAPH FAIRNESS UNDER DISTRIBUTION SHIFTS

In this section, we first establish a relationship between graph fairness and graph data distribution $\mathbb{P}(\mathbf{A}, \mathbf{X}|\mathbf{e})$. Then we gain insight into why distribution shifts may lead to fairness degradation.

3.1 Relationship between Data Distribution and Graph Fairness

We first use aggregated feature distance to establish the connection between data distribution and graph fairness. Referring to commonly used GNNs, we define the aggregated features as $\mathbf{H} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}\mathbf{X}$, where $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$. For the convenience of expression, we define sensitive group, EO group, and aggregated feature distance between EO groups as follows:

Definition 3.1. (Sensitive group) The sensitive group of nodes with sensitive attribute f is defined as:

$$\mathcal{V}_f = \{v_i \in \mathcal{V}|f_i = f\}. \quad (1)$$

Definition 3.2. (EO group) The EO group of nodes with label y and sensitive attribute f is defined as:

$$\mathcal{V}_f^y = \{v_i \in \mathcal{V}|(f_i = f) \cap (y_i = y)\}. \quad (2)$$

Definition 3.3. (Aggregated feature distance between sensitive groups with the same label) The aggregated feature distance between sensitive groups with label y is defined as:

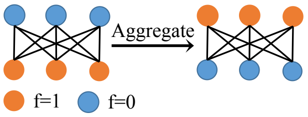

Figure 2: Example of low sensitive homophily.

$$\eta_y = \max_{v_a\in\mathcal{V}_1^y} \min_{v_b\in\mathcal{V}_0^y} ||h_a - h_b||_2, \quad (3)$$

where h_a and h_b are the aggregated features of nodes v_a and v_b, respectively.

The aggregated feature distance η_y defines the maximum shortest path from a node in $\mathcal{V}_1^y$ to a node in $\mathcal{V}_0^y$, thus can measure the aggregated feature difference between the two sensitive groups with label y. Large η_y implies that the aggregated features of different sensitive groups are easy to distinguish, and GNNs may make predictions based on this sensitive information, resulting in unfairness.

We then show that η_y is mainly affected by two factors determined by $\mathbb{P}(\mathbf{A}, \mathbf{X}|\mathbf{e})$. The first factor is the sensitive structure-property of the graph. Previous fairness studies [28, 40] focus on the sensitive homophily of the graph structure, defined as $\alpha = \mathbb{E}_{v_i\in\mathcal{V}} \frac{\sum_{j\in N_i\cup\{v_i\}} \mathbf{1}_{(f_i=f_j)}}{d_i + 1}$, where N_i is the neighbors of node v_i, $\mathbf{1}_{(f_i=f_j)}$ is the indicator function evaluating to 1 if and only if $f_i = f_j$. They believe that higher sensitive homophily will make the aggregated features of two sensitive groups more distinguishable, resulting in unfairness. However, we find that lower sensitive homophily will also make aggregated features of sensitive groups distinguishable. For example, the graph in Figure 2 has very low sensitive homophily according to α. After the aggregation step of GNN, different sensitive groups may change their features but are still distinguishable. We further point out that η_y is determined by whether the nodes tend to have balanced neighborhoods, i.e., the number of neighbors belonging to different sensitive groups is nearly the same. To theoretically analyze the relationship between balanced neighborhoods and η_y, we define a new sensitive balance degree to quantify the structure property:

Definition 3.4. (Sensitive balance degree) The sensitive balance degree of node v_i with sensitive attribute f_i is:

$$u_i = |p_i - q_i|, \quad (4)$$

where $p_i = \frac{\sum_{j\in N_i\cup\{v_i\}} \mathbf{1}_{(f_i=f_j)}}{d_i + 1}$ and $q_i = \frac{\sum_{j\in N_i\cup\{v_i\}} \mathbf{1}_{(f_i\neq f_j)}}{d_i + 1}$ represent the proportions of neighbors with the same and different sensitive attribute, respectively. The average sensitive balance degree on a graph is :

$$u = \mathbb{E}_{i\in\mathcal{V}}(u_i). \quad (5)$$

The sensitive balance degree reflects the difference in the number of neighbors around node v_i belonging to different sensitive groups. If a node has nearly the same number of neighbors with different sensitive attributes, then it has a more balanced neighborhood and smaller u_i, and vice versa.

The second factor that affects η_y is the feature difference between different sensitive groups. We assume features of nodes belonging to two sensitive groups follow Gaussian distribution, i.e., $\mathbb{P}(x_a \mid v_a \in \mathcal{V}_1) \sim \mathcal{N}(\mu_{\mathcal{V}_1}\mathbf{I}_\zeta, \sigma^2_{\mathcal{V}_1}\mathbf{I}_\zeta)$ and $\mathbb{P}(x_b \mid v_b \in \mathcal{V}_0) \sim \mathcal{N}(\mu_{\mathcal{V}_0}\mathbf{I}_\zeta, \sigma^2_{\mathcal{V}_0}\mathbf{I}_\zeta)$.

With the feature distribution of two sensitive groups and the graph structure property u, we can bound η_y with the following theorem:

Theorem 3.5. *For any $\delta \in (0, 1)$, with probability greater than $1 - \delta$ and large enough feature dimension ζ, we have:*

$$\eta_y^2 \geq (\sigma_{\mathcal{V}_1}^2 + \sigma_{\mathcal{V}_0}^2)\zeta(1 - 2\sqrt{\frac{log(2/\delta)}{\zeta}}) + \zeta u^2(\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0})^2,$$
$$\eta_y^2 \leq (\sigma_{\mathcal{V}_1}^2 + \sigma_{\mathcal{V}_0}^2)\zeta(1 + 4\sqrt{\frac{log(2/\delta)}{\zeta}}) + \zeta u^2(\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0})^2. \quad (6)$$

From the above theorem, we can find out that the upper bound and lower bound of η_y are determined by $(\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0})$ and u. Then we show that the fairness metric Δ_{EO} is actually bounded by η_y as the following theorem.

Theorem 3.6. *Consider an encoder $g : H \rightarrow Z \in \mathbb{R}^{n \times \zeta'}$ extracting ζ'-dimensional representations Z and an classifier $\omega : Z \rightarrow C \in \mathbb{R}^{n \times 2}$ predicting the binary labels of the nodes. Assume that g and ω have L_1-Lipschitz and L_2-Lipschitz continuity, respectively, then equalized odds is bounded by:*

$$\Delta_{EO} \leq L_1 L_2 \frac{\sum_{y=0}^{1} \eta_y}{2}. \quad (7)$$

Combining Theorem 3.6 and Theorem 3.5, we find out that Δ_{EO} is mainly affected by two key factors determined by $\mathbb{P}(\mathbf{A}, \mathbf{X}|\mathbf{e})$: 1) The feature difference between sensitive groups $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$. Larger $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$ indicates that the features of the two sensitive groups are easier to distinguish, resulting in larger Δ_{EO}. 2) The average sensitive balance degree of the graph u. Larger u implies the nodes in the graph tend to have unbalanced neighbors, resulting in larger Δ_{EO}.

We direct the readers to Appendix **??** for proofs of all the above theorems.

3.2 Bounds on Fairness on the Testing Graph

Given the factors that affect graph fairness, we can gain insight into the reason why distribution shifts may lead to unfairness.

As Δ_{EO} is determined by two factors affected by $\mathbb{P}(\mathbf{A}, \mathbf{X}|\mathbf{e})$, suppose the data distribution differs between the training graph and testing graphs, i.e., $\mathbb{P}(\mathbf{A}_{\mathcal{J}}, \mathbf{X}_{\mathcal{J}}|\mathbf{e} = \mathcal{J}) \neq \mathbb{P}_{\mathcal{K}}(\mathbf{A}_{\mathcal{K}}, \mathbf{X}_{\mathcal{K}}|\mathbf{e} = \mathcal{K})$, then the two factors including u and $(\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0})$ will also change, resulting in fairness deterioration in some cases. For example, if $(\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0})$ and u are small in the training graph but large in the testing graph, then the model is highly fair on the training graph but highly unfair on the testing graph.

Then we characterize the difference of Δ_{EO} between the training graph and the testing graph, denoted as $\Delta_{EO}^{\mathcal{J}} - \Delta_{EO}^{\mathcal{K}}$, by analyzing the accuracy difference between the training graph and the testing graph of each EO group. The EO group in the testing graph with label y and sensitive attribute f is denoted as $\mathcal{K}_f^y = \{v_i \in \mathcal{V}_{\mathcal{K}} | (f_i = f) \cap (y_i = y)\}$, where $\mathcal{V}_{\mathcal{K}}$ is the node set in the testing grah, and we define the prediction accuracy on $\mathcal{K}_f^y$ as $\mathbb{E}_{\mathcal{K}_f^y} = \mathbb{E}_{v_i \in \mathcal{K}_f^y}(\hat{y}_i = y)$. Similarly, on training graph we have $\mathbb{E}_{\mathcal{J}_f^y} = \mathbb{E}_{v_i \in \mathcal{J}_f^y}(\hat{y}_i = y)$. Then we bound the equalized odds difference for data in the training graph and testing graph as:

$$\Delta_{EO}^{\mathcal{K}} - \Delta_{EO}^{\mathcal{J}} \leq \sum_{y,f} |\mathbb{E}_{\mathcal{K}_f^y} - \mathbb{E}_{\mathcal{J}_f^y}|. \quad (8)$$

We then define EO group representation distance between the training graph and the testing graph in Definition 3.7, and build a relationship between the representation distance and $|\mathbb{E}_{\mathcal{K}_f^y} - \mathbb{E}_{\mathcal{J}_f^y}|$ in Theorem 3.8.

Definition 3.7. (EO group representation distance between the training graph and the testing graph) For EO group with label y and sensitive attribute f, we define the representation distance between the training graph and the testing graph as:

$$\epsilon_f^y = \max_{v_j \in \mathcal{K}_f^y} \min_{v_i \in \mathcal{J}_f^y} ||z_i - z_j||_2, \quad (9)$$

where z_i is the representation of node v_i learned by the encoder g.

Theorem 3.8. *Assume that the nonlinear transformation $\omega(Z) = RELU(ZW_\omega)$ has L_2-Lipschitz continuity, we have:*

$$|\mathbb{E}_{\mathcal{K}_f^y} - \mathbb{E}_{\mathcal{J}_f^y}| \leq L_2 \epsilon_f^y. \quad (10)$$

Then equalized odds difference between the training graph and the testing graph can be bounded as:

$$\Delta_{EO}^{\mathcal{K}} - \Delta_{EO}^{\mathcal{J}} \leq L_2 \sum_{f,y} \epsilon_f^y. \quad (11)$$

Based on Theorem 3.8, we can see that $\Delta_{EO}^{\mathcal{K}}$ relies on both $\Delta_{EO}^{\mathcal{J}}$ and EO group representation distance, which is determined by how much $\mathbb{P}(\mathbf{A}_{\mathcal{K}}, \mathbf{X}_{\mathcal{K}}|\mathbf{e} = \mathcal{K})$ differs from $\mathbb{P}(\mathbf{A}_{\mathcal{J}}, \mathbf{X}_{\mathcal{J}}|\mathbf{e} = \mathcal{J})$. Please note that our objective is not to achieve a tight bound for equalized odds on the testing graphs. Instead, our focus is on identifying the sufficient conditions that ensure fair performance on the testing graphs, and Theorem 3.8 actually reveals the sufficient conditions.

To alleviate the unfairness issue on the testing graph, i.e., minimize $\Delta_{EO}^{\mathcal{K}}$, we not only have to minimize $\Delta_{EO}^{\mathcal{J}}$, but also have to minimize the EO group representation distance. Also, minimizing the EO group representation distance leads to the minimization of $\Delta_{DP}^{\mathcal{K}}$. Detailed proofs of minimization of $\Delta_{DP}^{\mathcal{K}}$, Theorem 3.8, and Eq. (8) are deferred to Appendix **??**.

4 METHODOLOGY

In order to solve the unfairness under distribution shifts problem, motivated by the findings in Section 3, we present our framework (shown in Figure 3), which mainly includes three parts: (a) the generative adversarial debiasing module to get smaller $\Delta_{EO}^{\mathcal{J}}$ on the training graph, (b) the graph generation module to generate graphs with large bias and are under different distributions, (c) the EO group alignment module to minimize the EO group representation distance.

4.1 Adversarial Debiasing on Training Graph

As suggested in Theorem 3.8, to improve fairness on the testing graph, we have to ensure fairness on the training graph. Combining the aggregation step and the encoder g discussed in Section 3.1, we use a GNN-based encoder $\rho_{\theta_\rho} : (\mathbf{A}, \mathbf{X}) \rightarrow Z \in \mathbb{R}^{n \times \zeta'}$ with parameters θ_ρ to extract the ζ'-dimensional representations of the nodes. If the representations of different sensitive groups are distinguishable, then the classifier may make predictions based on this information, resulting in unfairness. In order to make the representations undistinguishable, we use a sensitive discriminator $\xi_{\theta_\xi} : Z \rightarrow F \in \{0, 1\}^n$ with parameters θ_ξ to predict the sensitive attributes of the nodes

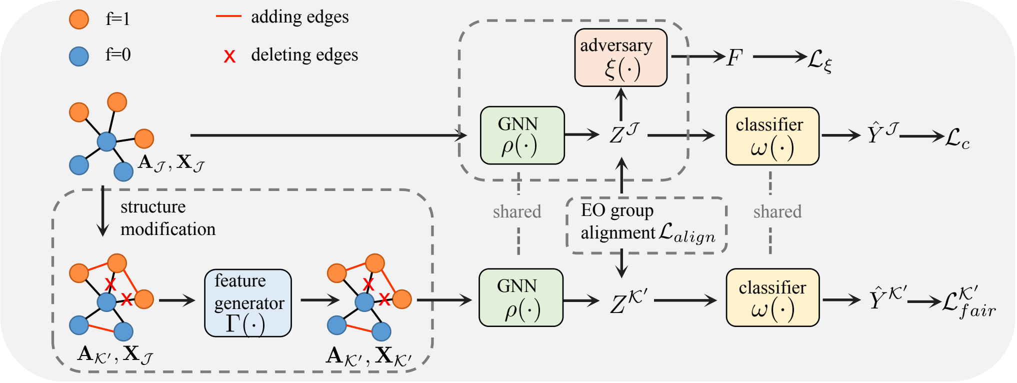

Figure 3: An overview of FatraGNN.

given their representations Z. And the encoder ρ_{θ_ρ} is trained to learn similar representations between sensitive groups, thus can fool the discriminator. Leveraging adversarial training, we compute the loss of the discriminator and encoder as:

$$\begin{aligned}\min_{\theta_\rho} \max_{\theta_\xi} \mathcal{L}_{\xi} = \mathbb{E}_{v_i \in \mathcal{J}} (f_i \log(\xi_{\theta_\xi}(\rho_{\theta_\rho}(x_i))) \\ +(1-f_i)\log(1-\xi_{\theta_\xi}(\rho_{\theta_\rho}(x_i)))).\end{aligned} \tag{12}$$

Besides, we also train the encoder together with an MLP-based classifier ω_{θ_ω} to minimize the classification loss to ensure accuracy:

$$\min_{\theta_\rho, \theta_\omega} \mathcal{L}_c = -\mathbb{E}_{v_i \in \mathcal{J}} \left(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right). \tag{13}$$

4.2 Graph Generation Module

To address the unfairness issue caused by data distribution shifts, we should also get similar representations between the training graph and the testing graph for each EO group, as suggested in Theorem 3.8. During the training process, we consider training ρ_{θ_ρ} to learn similar representations for graphs under the distribution of $\mathbb{P}(\mathbf{A}_{\mathcal{K}}, \mathbf{X}_{\mathcal{K}} | \mathbf{e} = \mathcal{K})$ and $\mathbb{P}(\mathbf{A}_{\mathcal{J}}, \mathbf{X}_{\mathcal{J}} | \mathbf{e} = \mathcal{J})$. As $\mathbb{P}(\mathbf{A}_{\mathcal{K}}, \mathbf{X}_{\mathcal{K}} | \mathbf{e} = \mathcal{K})$ is unknown during the training process, it is challenging to generate the exact testing graphs, so we generate graphs with significant bias and are under different distributions. If our model can handle graphs that are more likely to cause unfairness, it will be able to address unfairness issues under distribution shifts more effectively. The generated graphs follows distribution $\mathbb{P}(\mathbf{A}_{\mathcal{K}'}, \mathbf{X}_{\mathcal{K}'} | \mathbf{e} = \mathcal{K}')$. As demonstrated in Theorem 3.6, larger $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$ and u will lead to poor fairness performance. So we propose a graph generation module, including a structure modification step to generate $\mathbf{A}_{\mathcal{T}'}$ by modifying $\mathbf{A}_{\mathcal{J}}$, and a feature generator to generate $\mathbf{X}_{\mathcal{T}'}$ based on $\mathbf{X}_{\mathcal{J}}$. Thus we can generate graphs that will lead to unfairness and are under different distributions.

For the structure modification step, we generate graphs with larger u, i.e., graphs with significant unbalanced neighborhoods. Two strategies can be employed: One is randomly adding edges between nodes with the same sensitive attribute and removing edges between nodes with different sensitive attribute. The other is the inverse. Both are used to get a bunch of generated $\mathbf{A}_{\mathcal{T}'}$ with larger u before training.

To make our model adapt to various structures, we feed one of the generated graphs into training during every certain number of epochs, then we use an MLP-based feature generator $\Gamma_{\theta_\Gamma} : \mathbf{X}_{\mathcal{J}} \rightarrow \mathbf{X}_{\mathcal{T}'}$ to generate features with larger $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$. The generated graph with $\mathbf{A}_{\mathcal{T}'}$ and $\mathbf{X}_{\mathcal{T}'}$ is then feed into ρ and ω to make predictions. We also include a regularization term to ensure that the feature generator does not produce features that significantly stray from the features of the training graph. The feature generator is trained to maximize the fairness loss:

$$\begin{aligned}\max_{\theta_\Gamma} \mathcal{L}^{\mathcal{K}'}_{fair} = \frac{1}{2}\sum_{y=0}^{1} |\mathbb{E}_{v_i \in \mathcal{J}_1^y} (\hat{y}_i = y | \Gamma(x_i), \mathbf{A}_{\mathcal{T}'}) \\ -\mathbb{E}_{v_j \in \mathcal{J}_0^y} (\hat{y}_j = y | \Gamma(x_j), \mathbf{A}_{\mathcal{T}'})| - \tau ||\mathbf{X}_{\mathcal{T}'} - \mathbf{X}_{\mathcal{J}}||_F^2.\end{aligned} \tag{14}$$

where $|| \cdot ||_F^2$ is the Frobenius norm of matrix, τ is the coefficient.

Thus the feature generator can be trained to explore the features that lead to poor fairness performance but not deviate too much from the training graph. After the structure modification step and the feature generation step, we can generate graphs that lead to unfairness and under different distributions.

4.3 EO Group Alignment Module

We can learn from Theorem 3.8 that the unfairness issue on the testing graph can be alleviated by minimizing the EO group representation distance ϵ_f^y. Thus we aim to minimize EO group representation distance between the training graph and the generated graph.

We utilize a similarity score $\lambda_f^y = \mathbb{E}_{i \in \mathcal{J}_f^y, j \in \mathcal{K}'^y_f} \frac{(z_i z_j)}{||z_i|| \cdot ||z_j||}$ to measure the alignment of EO group representation. Higher λ_f^y implies better alignment and lower ϵ_f^y. Then we maximize the similarity score of all EO groups:

$$\max_{\theta_\rho} \mathcal{L}_{align} = \sum_{f,y} \lambda_f^y. \tag{15}$$

In this way, we can get smaller ϵ_f^y, implying better fairness performance on the testing graph. Furthermore, the alignment module improves the classification accuracy on generated graphs by guiding the GNN-based encoder to acquire similar representations for both the training graph and the generated graphs.

Meanwhile, the alignment module implicitly aids in preserving the causal features from disruption. The module forces the shared encoder to learn similar representations for the generated graphs and the training graph. As the encoder is shared, it is not possible to learn similar representations for features with significant differences. This implicitly forces the generator to learn features that are not significantly different from the original graph but may cause unfairness. Experimental analysis can be found in Section 5.4.

5 EXPERIMENT

Datasets We use five datasets to evaluate the performance of our model under distribution shifts. Each dataset comprises at least two graphs: one for training and validation, and others for testing. The five datasets comprise three real-world datasets and two semi-synthetic datasets summarized as follows. 1) **Pokecs** includes Pokec-z and Pokec-n, which are drawn from the popular social network in Slovakia [9] based on the provinces that users belong to. Both Pokec-z and Pokec-n consist of users belonging to two major regions of the corresponding provinces. We use Pokec-z for training and validation, and Pokec-n for testing. We treat "region" as the sensitive attribute, and the task is to predict the working field of the users. 2) **Bail-Bs** is obtained from the commonly used fairness-related graph Bail [23], where nodes are defendants released on bail. Utilizing the modularity-based community detection method [30], we partition Bail into communities and find that they exhibit different data distributions. Then we retain five large communities and name them from B0 to B4. We use B0 for training and validation, and the remaining graphs B1 to B4 for testing. The task is to decide whether to bail the defendants with "race" being the sensitive attribute. 3) **Credit-Cs** is partitioned the same way as Bail-Bs from Credit [43], where nodes represent credit card users. We get five communities named from C0 to C4. C0 is used for training and validation, while C1 to C4 are used for testing. The task is to classify the credit risk of the clients as high or low with "age" being the sensitive attribute. 4) **sync-B1s** comprises testing graphs with different u from 0 to 0.6 obtained by modifying the structure of B1, and a training graph B0. 5) **sync-B2s** consisits of testing graphs obtained by modifying B2 the same way as sync-B1s and a training graph B0. More details such as dataset statistics and the data distribution of the training graph and testing graphs can be found in Appendix **??**.

Baselines We compare our model with nine baselines: 1) Traditional learning methods: MLP [31], GCN [24]. 2) Fair GNNs: FairVGNN [40], NIFTY [1], EDITS [10], CAF [17]. 3) Out-of-distribution (OOD) GNN: EERM [41]. 4) Model-agnostic OOD method: SAGM [37]. 5) Fairness under distribution shifts methods: RFR [20].

Performance Evaluation We use accuracy (ACC) and ROC-AUC to evaluate the predictive performance of the node classification task. To measure fairness, we use Δ_{DP} and Δ_{EO} introduced in Section 2. Note that a model with lower Δ_{DP} and Δ_{EO} implies better fairness performance. To comprehensively assess the classification and fairness performance of a model across various testing graphs, we introduce a metric denoted as $s = \text{ACC}+\text{ROC-AUC}-\Delta_{DP}-\Delta_{EO}$, where greater values of this metric indicate superior model performance. We calculate the total score for each method by summing up their scores on all testing graphs, and then provide the overall rankings for each method.

Experimental Setting We perform a hyperparameter search for our model on all dataset groups. For other baseline models: GCN, MLP, FairVGNN, NIFTY, EDITS, and EERM, we carefully fine-tune them to get optimal performance on all the dataset groups. Note that EDITS and EERM have higher complexity and are hard to be trained on Pokec-z , so we only report the results of other baselines on Pokecs. For all methods, we randomly run 5 times and report the mean and variance of each metric. More details such as the hyperparameter setting can be found in Appendix **??**.

5.1 Evaluation on Real-world Datasets

We use three real-world datasets for evaluation: Pokecs, Bail-Bs, and Credit-Cs.

Results Table 1 and Table 2 show the effectiveness of FatraGNN in terms of classification and fairness performance on all testing graphs in Bail-Bs and Pokecs. Due to space limitations, we defer the results on Credit-Cs to Appendix **??**.

We observe that the proposed FatraGNN outperforms all baselines in most cases. Additionally, we find that while fairness baselines aim to improve fairness performance, they cannot perform well on testing graphs when distribution shifts. Although the graph OOD model EERM achieves better classification performance than fairness baselines when distribution shifts, it has lower fairness performance on all the testing graphs because it cannot learn fair representations.

We also analyze the relationship between accuracy and Δ_{EO} of the models, because good fairness performance could be a result of poor classification performance. For example, if a model misclassifies all samples, then the accuracy on all EO groups will be 0, resulting in $\Delta_{EO} = 0$, which implies good fairness performance. However, this is not the ideal model. Fairness models may ensure fairness at the cost of accuracy, so we further show the Pareto front curves [34], which are generated by a grid search of hyperparameters, to show this trade-off between accuracy and Δ_{EO}. As shown in Figure 4, the horizontal axis represents Δ_{EO} and the vertical axis represents accuracy. Curves closer to the upper-left corner imply higher accuracy and lower Δ_{EO}, indicating better trade-off performance. We can see that FatraGNN achieves better performance than fairness baselines in terms of this trade-off.

5.2 Evaluation on Semi-synthetic Datasets

We further use sync-B1s and sync-B2s to test the performance of each method on testing graphs with different u. Testing graphs with higher u have less balanced neighbors and may result in unfairness.

Results As u is calculated by $|p - q|$, we find that accuracy of the models have different changing trend when $p - q > 0$ and $p - q < 0$. In order to demonstrate the performance of models more clearly, we use $u' = p - q$ instead to reflect the average sensitive balance degree of the graphs.

The classification and fairness performance are shown in Figure 5. Overall, our FatraGNN outperforms other baselines in terms of both accuracy and Δ_{EO} on most testing graphs. Moreover, FatraGNN demonstrates low variance in both classification and fairness performance across different testing graphs with various u', indicating its potential to perform well when distribution shifts. Additionally, we find that most models achieve their optimal fairness performance when u' is close to 0. When $u = |u'|$ increases, Δ_{EO} also increases, verifying our analysis in Section 3.1 that unbalanced neighborhoods will lead to unfairness. Additionally, we find that baselines tend to achieve better accuracy as u' increases. This is because the nodes

Table 1: Classification and fairness performance (%±σ) on Bail-Bs. ↑ denotes the larger, the better; ↓ denotes the opposite. Best ones are in bold.

	metric	MLP	GCN	FairVGNN	NIFTY	EDITS	EERM	CAF	SAGM	RFR	FatraGNN (ours)
B1	ACC↑	70.53±1.01	72.93±4.06	69.76±2.03	69.54±7.26	72.69±1.72	73.25±1.4	69.39±2.30	73.08 ±4.25	71.63 ±1.52	**74.59±0.93**
	ROC-AUC↑	62.76±1.87	59.41±14.42	64.82±4.32	62.65±5.95	59.91±0.31	63.98±1.28	62.84±1.84	62.76±3.45	62.39±1.53	**66.0±0.01**
	Δ_{DP} ↓	4.83±9.38	4.58±0.78	11.05±4.58	7.21±4.54	4.35±1.3	8.85±2.57	4.46±2.03	7.33±4.59	2.57±1.24	**1.14±2.87**
	Δ_{EO} ↓	7.48±7.31	10.19±2.3	8.35±4.82	9.57±2.8	9.22±0.97	10.93±2.38	4.97±2.31	7.35±4.56	2.63±0.87	**2.38±3.19**
B2	ACC↑	64.33±0.63	69.88±0.45	65.03±2.4	69.95±8.3	69.03±0.16	70.2±0.12	69.36±1.37	68.67±3.24	68.62±1.42	**70.46±0.44**
	ROC-AUC↑	59.21±1.18	68.35±10.68	70.21±2.61	65.93±13.46	74.25±0.73	72.23±0.49	71.58±2.03	70.67±2.14	70.25±2.35	**73.27±4.48**
	Δ_{DP} ↓	8.36±1.62	6.91±0.58	5.64±2.78	3.21±4.54	3.2±3.06	8.31±0.5	2.53±3.62	5.78±2.53	2.15±1.94	**0.15±0.79**
	Δ_{EO} ↓	6.51±0.32	8.68±0.2	3.23±3.47	3.57±2.8	2.89±0.54	6.29±0.12	3.81±2.08	6.34±3.56	2.64±1.63	**0.43±1.14**
B3	ACC↑	60.76±0.18	68.56±4.2	70.63±0.61	68.8±9.76	68.56±1.82	70.69±5.42	68.97±2.44	69.50±2.12	68.36±1.89	**71.65±4.65**
	ROC-AUC↑	62.89±2.87	72.99±0.68	80.76±5.01	77.98±5.5	79.28±1.48	79.98±3.61	78.04±2.67	78.43±3.90	78.74±1.95	**82.17±3.63**
	Δ_{DP} ↓	9.8±0.38	12.72±2.44	8.05±0.45	6.21±4.54	5.24±0.03	5.64±3.49	6.32±2.45	6.78±3.23	**4.23±1.72**	5.02±3.54
	Δ_{EO} ↓	6.29±0.36	14.15±3.09	9.18±0.36	5.57±2.8	3.08±0.27	4.65±1.21	4.32±2.67	5.67±2.84	4.72±2.17	**2.43±4.94**
B4	ACC↑	63.13±1.69	69.43±0.48	68.99±2.44	57.96±11.99	68.42±0.14	70.9±1.36	67.33±2.67	70.88±0.98	69.18±2.68	**72.59±3.39**
	ROC-AUC↑	61.57±0.97	76.4±0.78	77.23±1.14	69.21±5.39	69.2±1.41	68.81±2.27	71.93±1.64	69.34±1.89	68.35±2.52	**77.36±3.79**
	Δ_{DP} ↓	4.45±3.15	4.49±1.13	5.21±6.03	3.21±4.54	3.2±9.1	7.23±0.26	3.84±1.41	6.36±6.32	3.43±2.45	**2.48±3.09**
	Δ_{EO} ↓	3.29±3.54	8.74±1.62	5.33±6.18	2.57±2.8	5.6±7.86	9.04±0.86	5.36±2.19	7.34±4.67	3.51±2.39	**2.45±6.67**
	rank	10	9	5	8	2	7	3	6	4	**1**

Table 2: Quantitative results (%±σ) on Pokecs. (bold: best)

	metric	MLP	GCN	FairVGNN	NIFTY	CAF	SAGM	RFR	FatraGNN (ours)
Pokec-n	ACC↑	52.74±3.67	54.83±2.34	60.8±0.54	58.68±5.54	59.37±1.45	58.78±2.33	57.42±3.68	**62.00±0.24**
	ROC-AUC↑	65.38±0.43	63.48±2.34	65.26±1.45	67.09±2.25	66.86±1.32	65.67±2.45	65.29±1.36	**67.82±3.23**
	Δ_{DP} ↓	4.86±1.23	7.38±0.28	5.88±2.34	4.21±3.43	5.49±2.65	5.67±3.22	4.56±2.85	**1.34±0.27**
	Δ_{EO} ↓	4.16±2.34	6.37±0.52	6.26±2.21	3.82±3.88	5.02±0.73	4.19±2.45	3.41±2.37	**1.43±2.68**
	rank	7	8	6	2	3	5	4	**1**

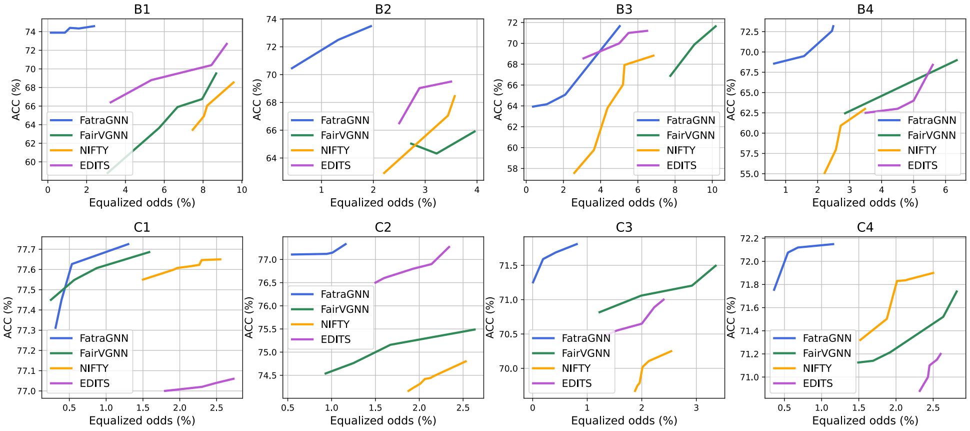

Figure 4: Trade-off of ACC and Δ_{EO} on all testing graphs of Bail-Bs, Credit-Cs. Upper-left corner (high accuracy, low Δ_{EO}) is preferred. The first row shows the results on B1 to B4. The second row shows the results on C1 to C4.

with the same sensitive attribute tend to share the same label, which provides additional information for the classification task.

5.3 Ablation Study

To fully understand the effect of each component of FatraGNN on alleviating unfairness under distribution shifts, we propose several variants of FatraGNN, including **Fatra w/o AD** as removing the adversarial module, **Fatra w/o GE** as removing the graph generation module, **Fatra w/o MD** as removing the structure modification step, and **Fatra w/o AL** as removing the EO group alignment module. Results of the ablation study on sync-B1s and Bail-Bs are shown in Figure 6. We can see that FatraGNN consistently outperforms the other variants. Without the adversarial module, Fatra w/o AD learns distinguishable representations for the two sensitive groups, resulting in poor fairness performance. Without the graph generation module, Fatra w/o GE fails to perform well when distribution shifts. Without the alignment module, Fatra w/o AL only generates graphs but is not trained to learn similar representations between the input graph and the generated graph for each EO group, resulting in similar poor performance as Fatra w/o GE. Without modification of the structure, Fatra w/o MD still performs better than Fatra w/o

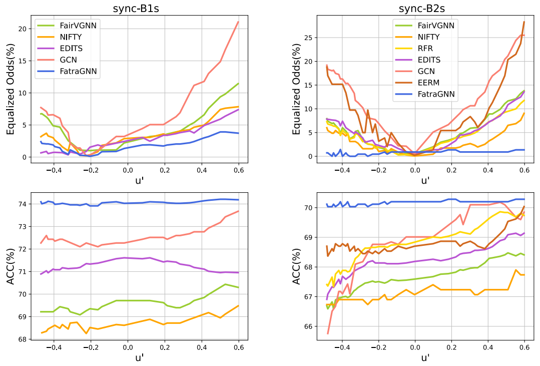

Figure 5: Accuracy and Δ_{EO} on sync-B1s (left) and sync-B2s (right).

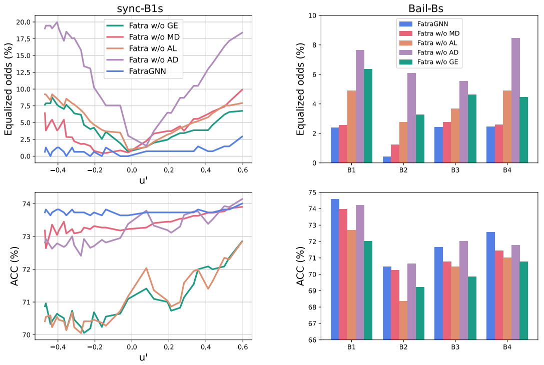

Figure 6: Ablation study on sync-B1s (left) and Bail-Bs (right). Please note that testing graphs of sync-B1s have continuously varying u', so we utilize a line chart to illustrate the performance change of the model on graphs with varying u'.

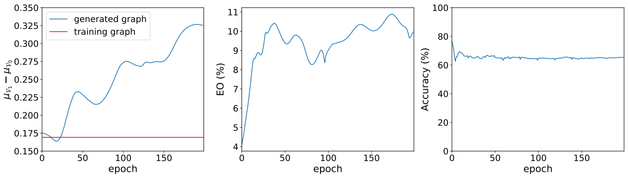

Figure 7: The representation difference between sensitive groups, equalized odds, and accuracy on the generated graph during training.

GE and Fatra w/o AL, since it is trained to adapt to different feature distributions. However, due to the lack of training graph with different u', Fatra w/o MD cannot perform well on testing graphs with different u'.

5.4 Additional Analysises

Analysis of Generated Graphs We also analyze the generated graphs and find that the generated graphs have different distributions from the training graph, and their casual features are maintained.

First, we show that the generated graphs are under different distributions from the training graph. Usually, graphs with different u and $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$ have different distributions because the structures of the graphs and the feature difference between different sensitive groups are different. In the generation module, we modify the structure of the graph to get larger u. Also, as we learn feature representations through an end-to-end method to get graphs that lead to unfairness, $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$ of the generated graph will also change during training. We do experiments Bail-Bs. As shown in Figure 7, as the number of epochs increases, $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$ increases, indicating that the generated graphs are under different distributions from the training graph.

Then we examine if there are potential disruptions in the causal features of the generated graphs. We conduct experiments on Bail-Bs to observe the changes in $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$, ΔEO, and accuracy during the training process. As shown in Figure 7, during training, the generated graphs have larger $\mu_{\mathcal{V}_1} - \mu_{\mathcal{V}_0}$ and poorer fairness (larger ΔEO). This suggests that the generated graphs will lead to unfairness and are under different distributions. Still, the accuracy hardly decreases, indicating that the key features of the graph are not disrupted.

Analysis of Representation Distances To demonstrate that our model can ensure alignment between the representations of the training graph and testing graphs for better fairness and accuracy performance, we plot the representations of the training graph and testing graphs of Bail-Bs using t-SNE [36]. The implementation details and result figures can be found in Appendix **??**. We can see that the representations of nodes belonging to the same EO group on the training graph and testing graphs are close, indicating that by minimizing the representation difference between the training graph and the generated graphs, the alignment module of our model can ensure the proximity of representations of the same EO group between the training graph and testing graphs, thereby guaranteeing fairness and accuracy on the testing graphs.

Analysis of Convergence We notice that during training, it is not difficult to tune the parameters to achieve convergence. Despite that no theory can guarantee convergence to the saddle point, it functions well in our experiments, which has also been observed in many other adversarial methods [16, 32, 40]. Additional experiments are provided in Appendix **??**.

Other additional experiment results such as hyperparameter study can be found in Appendix **??**.

6 CONCLUSION

In this work, we study the unfairness problem under distribution shifts on graphs, which is crucial for the real-world applications of fair GNNs. We theoretically prove that graph fairness is determined by a sensitive structure property and feature difference between sensitive groups of the graph, and explain the reason why distribution shifts will lead to unfairness. We then derive an upper bound for fairness on the testing graph. Based on our analysis, we further propose a novel FatraGNN framework to alleviate this problem. Experimental results demonstrate that FatraGNN consistently outperforms state-of-the-art baselines in terms of fairness-accuracy trade-off performance under distribution shifts.

REFERENCES

[1] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. 2021. Towards a unified framework for fair and stable graph representation learning. In *Uncertainty in Artificial Intelligence*. PMLR, 2114–2124.

[2] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron C. Courville. 2021. Systematic generalisation with group invariant predictions. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=b9PoimzZFJ

[3] Bang An, Zora Che, Mucong Ding, and Furong Huang. 2022. Transferring Fairness under Distribution Shifts via Fair Consistency Regularization. In *NeurIPS*. http://papers.nips.cc/paper_files/paper/2022/hash/d1dbaabf454a479ca86309e66592c7f6-Abstract-Conference.html

[4] Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant Risk Minimization. *CoRR* abs/1907.02893 (2019). arXiv:1907.02893 http://arxiv.org/abs/1907.02893

[5] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[6] Avishek Bose and William Hamilton. 2019. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*. PMLR, 715–724.

[7] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. 2009. Building classifiers with independency constraints. In *2009 IEEE international conference on data mining workshops*. IEEE, 13–18.

[8] Enyan Dai and Suhang Wang. 2021. 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*. 680–688.

[9] Enyan Dai and Suhang Wang. 2022. Learning fair graph neural networks with limited and private sensitive attribute information. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[10] Yushun Dong, Ninghao Liu, Brian Jalaian, and Jundong Li. 2022. Edits: Modeling and mitigating data bias for graph neural networks. In *Proceedings of the ACM Web Conference 2022*. 1259–1269.

[11] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[12] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. 2022. Debiasing graph neural networks via learning disentangled causal substructure. *Advances in Neural Information Processing Systems* 35 (2022), 24934–24946.

[13] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. 2023. Generalizing graph neural networks on out-of-distribution graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[14] Shaohua Fan, Xiao Wang, Chuan Shi, Kun Kuang, Nian Liu, and Bai Wang. 2022. Debiased graph neural networks with agnostic label selection bias. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[15] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 259–268.

[16] 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* 63, 11 (2020), 139–144.

[17] Zhimeng Guo, Jialiang Li, Teng Xiao, Yao Ma, and Suhang Wang. 2023. Towards Fair Graph Neural Networks via Graph Counterfactual. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, 669–678. https://doi.org/10.1145/3583780.3615092

[18] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[19] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[20] Zhimeng Jiang, Xiaotian Han, Hongye Jin, Guanchu Wang, Rui Chen, Na Zou, and Xia Hu. 2023. Chasing Fairness Under Distribution Shift: A Model Weight Perturbation Approach. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[21] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*. PMLR, 2323–2332.

[22] Jeff Johnson, Donald M Truxillo, Berrin Erdogan, Talya N Bauer, and Leslie Hammer. 2009. Perceptions of overall fairness: are effects on job performance moderated by leader-member exchange? *Human Performance* 22, 5 (2009), 432–449.

[23] Kareem L Jordan and Tina L Freiburger. 2015. The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length. *Journal of Ethnicity in Criminal Justice* 13, 3 (2015), 179–196.

[24] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[25] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *Advances in neural information processing systems* 30 (2017).

[26] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[27] Yibo Li, Xiao Wang, Hongrui Liu, and Chuan Shi. 2023. A generalized neural diffusion framework on graphs. *arXiv preprint arXiv:2312.08616* (2023).

[28] Hongyi Ling, Zhimeng Jiang, Youzhi Luo, Shuiwang Ji, and Na Zou. 2023. Learning fair graph representations via automated data augmentations. In *The Eleventh International Conference on Learning Representations*.

[29] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[30] Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103, 23 (2006), 8577–8582.

[31] Sankar K Pal and Sushmita Mitra. 1992. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on neural networks* 3, 5 (1992), 683–697.

[32] Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. 2018. Global versus localized generative adversarial nets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1517–1525.

[33] Candice Schumann, Xuezhi Wang, Alex Beutel, Jilin Chen, Hai Qian, and Ed H Chi. 2019. Transfer of machine learning fairness across domains. *arXiv preprint arXiv:1906.09688* (2019).

[34] Jürgen Teich. 2001. Pareto-front exploration with uncertain objectives. In *Evolutionary Multi-Criterion Optimization: First International Conference, EMO 2001 Zurich, Switzerland, March 7–9, 2001 Proceedings*. Springer, 314–328.

[35] Songül Tolan, Marius Miron, Emilia Gómez, and Carlos Castillo. 2019. Why machine learning may lead to unfairness: Evidence from risk assessment for juvenile justice in catalonia. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 83–92.

[36] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[37] Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. 2023. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3769–3778.

[38] Ruijia Wang, Xiao Wang, Chuan Shi, and Le Song. 2022. Uncovering the Structural Fairness in Graph Contrastive Learning. *Advances in Neural Information Processing Systems* 35 (2022), 32465–32473.

[39] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community preserving network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[40] Yu Wang, Yuying Zhao, Yushun Dong, Huiyuan Chen, Jundong Li, and Tyler Derr. 2022. Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1938–1948.

[41] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. 2022. Handling Distribution Shifts on Graphs: An Invariance Perspective. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=FQOC5u-1egI

[42] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. *Advances in neural information processing systems* 30 (2017).

[43] I-Cheng Yeh and Che-hui Lien. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications* 36, 2 (2009), 2473–2480.

[44] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[45] Xingtong Yu, Zhenghao Liu, Yuan Fang, Zemin Liu, Sihong Chen, and Xinming Zhang. 2023. Generalized Graph Prompt: Toward a Unification of Pre-Training and Downstream Tasks on Graphs. *arXiv preprint arXiv:2311.15317* (2023).

[46] Xingtong Yu, Zemin Liu, Yuan Fang, and Xinming Zhang. 2023. Learning to count isomorphisms with graph neural networks. *arXiv preprint arXiv:2302.03266* (2023).

[47] Ziqian Zeng, Rashidul Islam, Kamrun Naher Keya, James Foulds, Yangqiu Song, and Shimei Pan. 2021. Fair representation learning for heterogeneous information networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 15. 877–887.