EDITS: Modeling and Mitigating Data Bias for Graph Neural Networks

Yushun Dong[1], Ninghao Liu[2], Brian Jalaian[3], Jundong Li[1]
[1]University of Virginia, [2]University of Georgia, [3]U.S. Army Research Laboratory
{yd6eb,jundong}@virginia.edu,ninghao.liu@uga.edu,brian.a.jalaian.civ@mail.mil

ABSTRACT

Graph Neural Networks (GNNs) have shown superior performance in analyzing attributed networks in various applications. Nevertheless, in high-stake decision-making scenarios such as online fraud detection, there is an increasing societal concern that GNNs could make discriminatory decisions towards certain demographic groups. Despite recent explorations on fair GNNs, these works are tailored for a specific GNN model. However, myriads of GNN variants have been proposed for different applications, and it is costly to fine-tune existing debiasing algorithms for each specific GNN architecture. Different from existing works that debias GNN models, we aim to debias the input attributed network to achieve fairer GNNs through feeding GNNs with less biased data. Specifically, we propose novel definitions and metrics to measure the bias in an attributed network, which leads to the optimization objective to mitigate bias. We then develop a framework EDITS to mitigate the bias in attributed networks while maintaining the performance of GNNs in downstream tasks. EDITS works in a model-agnostic manner, i.e., it is independent of any specific GNN. Experiments demonstrate the validity of the proposed bias metrics and the superiority of EDITS on both bias mitigation and utility maintenance.

CCS CONCEPTS

• **Computing methodologies → Machine learning**.

KEYWORDS

graph neural networks, algorithmic fairness, data bias

ACM Reference Format:
Yushun Dong[1], Ninghao Liu[2], Brian Jalaian[3], Jundong Li[1]. 2022. EDITS: Modeling and Mitigating Data Bias for Graph Neural Networks. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3512173

1 INTRODUCTION

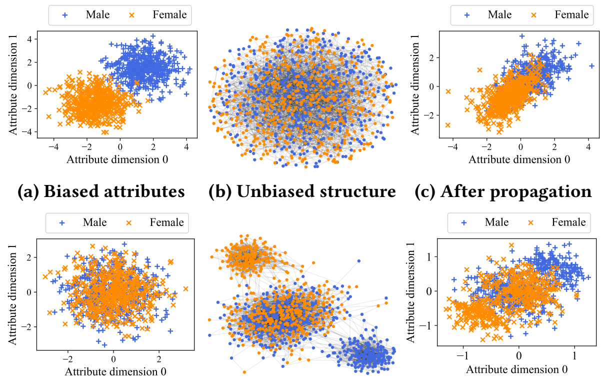
Attributed networks are ubiquitous in a plethora of web-related applications including online social networking [51], web advertising [61], and news recommendation [45]. To better understand these networks, various graph mining algorithms have been proposed. In particular, the recently emerged Graph Neural Networks (GNNs) have demonstrated superior capability of analyzing attributed networks in various tasks, such as node classification [28, 55] and link prediction [29, 63]. Despite the superior performance of GNNs, they usually do not consider fairness issues in the learning process [10]. Extensive research efforts have shown that many recently proposed GNNs [10, 49, 59] could make biased decisions towards certain demographic groups determined by sensitive attributes such as gender [16] and political ideology [42]. For example, e-commerce platforms generate a huge amount of user activity data, and such data is often constructed as a large attributed network in which entities (e.g., buyers, sellers, and products) are nodes while activities between entities (e.g.., purchasing and reviewing) are edges. To prevent potential losses, fraud entities (e.g., manipulated reviews and fake buyers) need to be identified on these platforms, and GNNs have become the prevalent solution to achieve such goal [12, 37]. Nevertheless, GNNs may have the risk of using sensitive information (e.g., race and gender) to identify fraud entities, yielding inevitable discrimination. Therefore, it is a crucial problem to mitigate bias in these network-based applications.

Various efforts have been made to mitigate the bias exhibited in graph mining algorithms. For example, in online social networks, random walk algorithms can be modified via improving the appearance rate of minorities [7, 47]; adversarial learning is another popular approach, which aims to learn node embeddings that are not distinguishable on sensitive attributes [6, 40]. Some recent efforts have also been made to mitigate bias in the outcome of GNNs. For example, adversarial learning can also be adapted to GNNs for outcome bias mitigation [10]. Nevertheless, existing approaches to debias GNN outcomes are tailored for a specific GNN model on a certain downstream task. In practical scenarios, different applications could adopt different GNN variants [19, 28], and it is costly to train and fine-tune the debiasing approaches based on diverse GNN backbones. As a consequence, to mitigate bias more efficiently for different GNNs and tasks, developing a one-size-fits-all approach becomes highly desired. Then the question is: how can we perform debiasing regardless of specific GNNs and downstream tasks? Considering that a model trained on biased datasets also tends to be biased [5, 10, 62], directly debiasing the dataset itself can be a straightforward solution. There are already debiasing approaches modifying original datasets via perturbing data distributions or reweighting the data points in the dataset [8, 25, 57]. These approaches obtain less biased datasets, which help to mitigate bias in learning algorithms. In this regard, considering that debiasing for different GNNs is costly, it is also desired to mitigate the bias in attributed networks before they are fed into GNNs.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512173

In this paper, we make an initial investigation on debiasing attributed networks towards more fair GNNs. Specifically, we tackle the following challenges. (1) ***Data Bias Modeling***. Traditionally, bias modeling is coupled with the outcome of a specific GNN [10]. Based on the GNN outcome, bias can be modeled via different fairness notions, e.g., *Statistical Parity* [14] and *Equality of Opportunity* [20], to determine whether the outcome is discriminatory towards some specific demographic groups. Nevertheless, if debiasing is carried out directly based on the input attributed networks instead of the GNN outcome, the first and foremost challenge is how to appropriately model such data bias. (2) ***Multi-Modality Debiasing***. In fact, attributed networks contain both graph structure and node attribute information. Correspondingly, bias may exist with diverse formats across different data modalities. In this regard, how to debias attributed networks that have different data modalities is the second challenge that needs to be tackled. (3) ***Model-Agnostic Debiasing***. Existing GNN debiasing approaches require the outcome of a specific GNN for objective function optimization during training. Different from these approaches, model-agnostic debiasing for GNNs should not rely on any specific GNN, as our goal is to develop a one-size-fits-all data debiasing approach to benefit various GNNs. Clearly, such model-agnostic debiasing could have better generalization capability but becomes much more difficult compared with the model-oriented GNN debiasing approaches. Nevertheless, the ultimate goal of debiasing is still to ensure the GNN outcome does not exhibit any discrimination. Such a contradiction poses the challenge of how to properly formulate a debiasing objective that can be universally applied to different GNNs in downstream tasks.

To tackle the challenges above, we present novel data bias modeling approaches and a principled debiasing framework named EDITS (modE̲ling anD̲ mI̲tigating daT̲a biaS̲) to achieve model-agnostic attributed network debiasing for GNNs. Specifically, we first carry out preliminary analysis to illustrate how bias exists in the two data modalities of an attributed network (i.e., node attributes and network structure) and affects each other in the information propagation of GNNs. Then, we formally define *attribute bias* and *structural bias*, together with the corresponding metrics for data bias modeling. Besides, we formulate the problem of debiasing attributed networks for GNNs, and propose a novel framework named EDITS for bias mitigation. It is worth mentioning that EDITS is model-agnostic for GNNs. In other words, our goal is to obtain less biased attributed networks for the input of any GNNs. Finally, empirical evaluations on both synthetic and real-world datasets corroborate the validity of the proposed bias metrics and the effectiveness of EDITS. Our contributions are summarized as: (1) **Problem Formulation.** We formulate and make an initial investigation on a novel research problem: debiasing attributed networks for GNNs based on the analysis of the information propagation mechanism; (2) **Metric and Algorithm Design.** We design novel bias metrics for attributed networks, and propose a model-agnostic debiasing framework named EDITS to mitigate the bias in attributed networks before they are fed into GNNs; (3) **Experimental Evaluation.** We conduct comprehensive experiments on both synthetic and real-world datasets to verify the validity of the proposed bias metrics and the effectiveness of the proposed framework.



(a) Biased attributes **(b) Unbiased structure** **(c) After propagation**

(d) Unbiased attributes **(e) Biased structure** **(f) After propagation**

Figure 1: Two exemplary cases illustrating how bias in the two data modalities of an attributed network introduce bias in GNN information propagation. Here (c) is the node attribute distribution after propagation with biased node attributes (a) and unbiased network structure (b); while (f) is the attribute distribution after propagation with unbiased node attributes (d) and biased network structure (e).

2 PRELIMINARY ANALYSIS

We provide two cases to show how the two data modalities of an attributed network (i.e., node attribute and network structure) introduce bias in information propagation – the most common operation in GNNs. These two cases also bring insights on tackling the three challenges mentioned in Sec. 1. Specifically, two synthetic datasets are generated with either biased node attribute or network structure, and then attributes are propagated across the network structure to show how bias is introduced in GNNs. Here we consider the attribute distribution difference between different demographic groups as the bias in attribute, while the group membership distribution difference of the neighbors for nodes between different demographic groups is regarded as the bias in network structure. Such bias in attribute and structure can be regarded as the bias that existed in two data modalities in an attributed network. It should be noted that using distribution difference to define the level of bias is consistent with many algorithmic fairness studies [14, 62], Now we explain how the synthetic datasets are generated. We assume the *sensitive attribute* is gender, and 1,000 nodes are generated with half males (blue) and half females (orange) for both cases. In addition to the sensitive attribute, each node is with an extra two-dimensional attribute vector, which will be initialized and fed as input for information propagation. To introduce bias to either of the data modalities, different strategies are adopted to generate the attribute vector and the network structure. To study how the two data modalities introduce bias in information propagation, we compare the distribution difference of attributes between groups before and after the propagation in GCN [28].

Case 1: Biased attributes and unbiased structure. In this case, we generate biased two-dimensional attribute vectors for nodes from the two groups (i.e., males and females) and unbiased network structure. Specifically, biased attributes at each dimension is generated independently with Gaussian distribution $\mathcal{N}$(-1.5, 1^2) for female and $\mathcal{N}$(1.5, 1^2) for male. The distributions are shown in

Fig. (1a). We then introduce how an unbiased network structure is generated. For each node in an unbiased network structure, the expected membership ratio of any group in its neighbor node set should be independent of the membership of the node itself. In this regard, we generate unbiased network structure via *random graph* model with edge formation probability as 2×10^{-3}. The visualization of the network is presented in Fig. (1b). The attribute distribution after information propagation according to the network structure is shown in Fig. (1c). Comparing Fig. (1a) (attribute distribution before propagation) with (1c) (attribute distribution after propagation), we can see the unbiased structure helps mitigate the original attribute bias after attributes are propagated according to the network structure. This not only implies that the attribute distribution difference between groups is a vital source of bias, but also demonstrates that unbiased structure helps mitigate bias in attributes after the information propagation process.

Case 2: Unbiased attributes and biased structure. In this case, unbiased attributes are generated independently at each dimension with $\mathcal{N}(0, 1^2)$ for both males and females. The distributions are shown in Fig. (1d). The biased network structure is generated as follows. For each node, we sum up its attribute values. Then, we rank all nodes in descending order according to the summation of attribute values. After that, given a threshold integer t, for the top-ranked t males and bottom-ranked t females, we assume that they form two separated communities. The two communities are shown as the bottom right community (males) and the upper left community (females) in Fig. (1e). We generate edges via *random graph* model with edge formation probability as 5×10^{-2} within each community. Similarly, the rest nodes form the third community via *random graph* model with edge formation probability as 1×10^{-2}. We also generate edges between nodes from the male (or female) community and the third community with the probability of 2×10^{-4}. In this way, we introduce bias in network structure. The final network is presented in Fig. (1e). The attribute distribution after propagation according to the network structure is shown in Fig. (1f). Comparing Fig. (1d) with (1f), we find that even if the original attributes are unbiased, the biased structure still turns the attributes into biased ones after information propagation. Hence the bias in the network structure is also a source of bias.

Here we draw three preliminary conclusions to help us tackle the challenges in Sec. 1. (1) For ***Data Bias Modeling***, bias in attributes can be modeled based on the difference of attribute distribution between two groups. Also, bias in network structure can be modeled based on the difference of attribute distribution between two groups after information propagation. (2) For ***Multi-Modality Debiasing*** in an attributed network, at least two debiasing processes should be carried out targeting the two data modalities (i.e., attributes and structure). (3) For ***Model-Agnostic Debiasing***, if the attribute distributions between groups can be less biased both before and after information propagation, the learned node representations tend to be indistinguishable between groups. Then GNNs trained on such data could also be less biased.

3 MODELING DATA BIAS FOR GNNS

In this section, we define *attribute bias* and *structural bias* in attributed networks together with their metrics.

3.1 Preliminaries

In this paper, without further specification, bold uppercase letters (e.g., $\mathbf{X}$), bold lowercase letters (e.g., $\mathbf{x}$), and normal lowercase letters (e.g., x) represent matrices, vectors, and scalars, respectively. For any matrix, e.g., $\mathbf{X}$, we use $\mathbf{X}_i$ denote its i-th row.

Let $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ be an undirected attributed network. Here $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, and $\mathbf{X} \in \mathbb{R}^{N \times M}$ is the node attribute matrix, where N is the number of nodes and M is the attribute dimension. Let a diagonal matrix $\mathbf{D}$ be the degree matrix of $\mathbf{A}$, where its (i,i)-th entry $\mathbf{D}_{i,i} = \sum_j \mathbf{A}_{i,j}$, and $\mathbf{D}_{i,j} = 0$ $(i \neq j)$. $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the graph Laplacian matrix. Denote the normalized adjacency matrix and the normalized Laplacian matrix as $\mathbf{A}_{\text{norm}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ and $\mathbf{L}_{\text{norm}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$. $|.|$ is the absolute value operator.

3.2 Definitions of Bias

We consider two types of bias on attributed networks, i.e., attribute bias and structural bias. We first define attribute bias as follows.

Definition 1. ***Attribute bias.*** *Given an undirected attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ and the group indicator (w.r.t. the sensitive attribute) for each node $\mathbf{s} = [s_1, s_2, ..., s_N]$, where $s_i \in \{0, 1\}$ $(1 \leq i \leq N)$. For any attribute, if its value distributions between different demographic groups are different, then attribute bias exists in $\mathcal{G}$.*

Besides, as shown in the second example in Sec. 2, bias can also emerge after attributes are propagated in the network even when original attributes are unbiased. Therefore, an intuitive idea to identify structural bias is to check whether information propagation in the network introduces or exacerbates bias [22]. Formally, we define structural bias on attributed networks as follows.

Definition 2. ***Structural bias.*** *Given an undirected attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ and the corresponding group indicator (w.r.t. sensitive attribute) for each node $\mathbf{s} = [s_1, s_2, ..., s_N]$, where $s_i \in \{0, 1\}$ $(1 \leq i \leq N)$. For the attribute values propagated w.r.t. $\mathbf{A}$, if their distributions between different demographic groups are different at any attribute dimension, then structural bias exists in $\mathcal{G}$.*

Apart from these definitions, it is also necessary to quantitatively measure the attribute bias and structural bias. In the sequel, we introduce our proposed metrics for the two types of bias.

3.3 Bias Metrics

Here we take the first step to define metrics for both *attribute bias* and *structural bias* for an undirected attributed network $\mathcal{G}$.

Attribute bias metric. Let $\mathbf{X}_{\text{norm}} \in \mathbb{R}^{N \times M}$ be the normalized attribute matrix. For the m-th dimension $(1 \leq m \leq M)$ of $\mathbf{X}_{\text{norm}}$, we use $\mathcal{X}_m^0$ and $\mathcal{X}_m^1$ to denote attribute value set for nodes with $s_i = 0$ and $s_i = 1$ $(1 \leq i \leq N)$. Then, attributes of all nodes can be divided into tuples: $\mathcal{X}_{total} = \{(\mathcal{X}_1^0, \mathcal{X}_1^1), (\mathcal{X}_2^0, \mathcal{X}_2^1), ..., (\mathcal{X}_M^0, \mathcal{X}_M^1)\}$. We measure attribute bias with Wasserstein-1 distance [54] between the distributions of the two groups:

$$b_{\text{attr}} = \frac{1}{M} \sum_m W(pdf(\mathcal{X}_m^0), pdf(\mathcal{X}_m^1)). \tag{1}$$

Here $pdf(\cdot)$ is the probability density function for a set of values, and $W(.,.)$ is the Wasserstein distance between two distributions. Intuitively, b_{attr} describes the average Wasserstein-1 distance between attribute distributions of different groups across all dimensions. It should be noted that taking the distribution difference between demographic groups as the indication of bias is in align with many existing algorithmic fairness studies [6, 10, 62].

Structural bias metric. As illustrated in Sec. 2, the key mechanism of GNNs is information propagation, during which the structural bias could be introduced. Let $\mathbf{P}_{\text{norm}} = \alpha\mathbf{A}_{\text{norm}} + (1-\alpha)\mathbf{I}$. Here $\mathbf{P}_{\text{norm}}$ can be regarded as a normalized adjacency matrix with re-weighted self-loops, where $\alpha \in [0, 1]$ is a hyper-parameter. Before measuring structural bias, we define the *propagation matrix* $\mathbf{M}_H \in \mathbb{R}^{N\times N}$ as:

$$\mathbf{M}_H = \beta_1\mathbf{P}_{\text{norm}} + \beta_2\mathbf{P}^2_{\text{norm}} + ... + \beta_H\mathbf{P}^H_{\text{norm}}, \tag{2}$$

where β_h $(1 \leq h \leq H)$ is re-weighting parameters. The rationale behind the formulation above is to measure the aggregated reaching likelihood from each node to other nodes within a distance of H. To achieve localized effect for each node, a desired choice is to let $\beta_1 \geq \beta_2 \geq ... \geq \beta_H$, i.e., emphasizing short-distance terms and reducing the weights of long-distance terms. For example, assume $H = 3$, then the value $(\mathbf{M}_3)_{i,j}$ is the aggregated reaching likelihood from node i to node j within 3 hops with re-weighting parameters being β_1, β_2 and β_3. Also, given attributes $\mathbf{X}_{\text{norm}}$, we define the *reachability matrix* $\mathbf{R} \in \mathbb{R}^{N\times M}$ as $\mathbf{R} = \mathbf{M}_H\mathbf{X}_{\text{norm}}$. Intuitively, $\mathbf{R}_{i,m}$ is the aggregated reachable attribute value for attribute m of node i. We utilize $\mathcal{R}^0_m$ and $\mathcal{R}^1_m$ to represent the set of values of the m-th dimension in $\mathbf{R}$ for nodes with $s_i = 0$ and $s_i = 1$ $(1 \leq i \leq N)$. The entries in $\mathbf{R}$ can also be divided into tuples according to attribute dimensions: $\mathcal{R}_{total} = \{(\mathcal{R}^0_1, \mathcal{R}^1_1), (\mathcal{R}^0_2, \mathcal{R}^1_2), ..., (\mathcal{R}^0_M, \mathcal{R}^1_M)\}$. We define structural bias as:

$$b_{\text{stru}} = \frac{1}{M}\sum_m W(pdf(\mathcal{R}^0_m), pdf(\mathcal{R}^1_m)). \tag{3}$$

Here b_{stru} is defined in a similar way as b_{attr}, except that the former uses $\mathcal{R}^0_m$ and $\mathcal{R}^1_m$ instead of $\mathcal{X}^0_m$ and $\mathcal{X}^1_m$. In this way, structural bias b_{stru} describes the average difference between aggregated attribute distributions of different groups after rounds of propagation.

3.4 Problem Statement

Based on the definitions and metrics in Sec. 3.2 and 3.3, we argue that if both b_{attr} and b_{stru} are reduced, bias in an attributed network can be mitigated. As a result, if GNNs are trained on such data, the bias issues in downstream tasks could also be alleviated. Formally, we define the debiasing problem as follows.

Problem 1. ***Debiasing attributed networks for GNNs.*** *Given an attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, our goal is to debias $\mathcal{G}$ by reducing b_{attr} and b_{stru} to obtain $\tilde{\mathcal{G}} = (\tilde{\mathbf{A}}, \tilde{\mathbf{X}})$, so that the bias of GNNs trained on $\tilde{\mathcal{G}}$ is mitigated. The debiasing is independent of any specific GNNs.*

4 MITIGATING DATA BIAS FOR GNNS

In this section, we discuss how to tackle Problem 1 with our proposed framework EDITS. We focus on the binary sensitive attribute for the sake of simplicity and discuss the extension later. We first present an overview of EDITS, followed by the formulation of the objective function. Finally, we present the optimization process.

4.1 Framework Overview

An overview of the proposed framework EDITS is shown in Fig. (2). Specifically, EDITS consists of three modules: (1) ***Attribute Debiasing.*** This module learns a debiasing function $g_{\boldsymbol{\theta}}$ with learnable parameter $\boldsymbol{\theta} \in \mathbb{R}^M$. The debiased version of $\mathbf{X}$ is obtained as output where $\tilde{\mathbf{X}} = g_{\boldsymbol{\theta}}(\mathbf{X})$; (2) ***Structural Debiasing.*** This module outputs $\tilde{\mathbf{A}}$ as the debiased $\mathbf{A}$. Specifically, $\tilde{\mathbf{A}}$ is initialized with $\mathbf{A}$ at the beginning of the optimization process. The entries in $\tilde{\mathbf{A}}$ are optimized via gradient descent with binarization; (3) ***Wasserstein Distance Approximator.*** This module learns an f for each attribute dimension. f is utilized to estimate the Wasserstein distance between the attribute distributions of different groups.

4.2 Objective Function

In this subsection, we introduce the details of our framework. Following the Definition 1 and Definition 2, our goal is to reduce b_{attr} and b_{stru} simultaneously. For the ease of understanding, we first consider the m-th attribute dimension as an example, and then extend it to all M dimensions to obtain our objective function.

Let $P_{0,m}$ and $P_{1,m}$ be the value distribution at the m-th attribute dimension in $\mathbf{X}$ for nodes with sensitive attribute $s = 0$ and $s = 1$, respectively. Denote $x_{0,m} \sim P^{(h)}_{0,m}$ and $x_{1,m} \sim P^{(h)}_{1,m}$ as two random variables drawn from the two distributions. Assume that we have a function $g_{\theta_m} : \mathbb{R} \rightarrow \mathbb{R}$ to mitigate attribute bias, where $1 \leq m \leq M$. For the m-th dimension, we denote $x^{(0)}_{0,m} = g_{\theta_m}(x_{0,m}) \sim P^{(0)}_{0,m}$ and $x^{(0)}_{1,m} = g_{\theta_m}(x_{1,m}) \sim P^{(0)}_{1,m}$ as the debiasing results for $x_{0,m}$ and $x_{1,m}$, respectively. Here the superscript (0) indicates that no information propagation is performed in the debaising process. Correspondingly, when such operation is extended to all M dimensions, we will have the debiased attribute matrix $\tilde{\mathbf{X}}$. Apart from the goal of mitigating attribute bias, we also want to mitigate structural bias. Let $\tilde{\mathbf{A}}$ be the adjacency matrix from the debiased network structure, and $\tilde{\mathbf{P}}_{\text{norm}}$ denotes the normalized $\tilde{\mathbf{A}}$ with re-weighted self-loops. Information propagation with h hops using the debiased adjacency matrix could be expressed as $\tilde{\mathbf{P}}^h_{\text{norm}}\tilde{\mathbf{X}}$, where $1 \leq h \leq H$. Let $P^{(h)}_{0,m}$ and $P^{(h)}_{1,m}$ be the value distribution at the m-th column of $\tilde{\mathbf{P}}^h_{\text{norm}}\tilde{\mathbf{X}}$ for nodes with sensitive attribute $s = 0$ and $s = 1$, respectively. Denote $x^{(h)}_{0,m} \sim P^{(h)}_{0,m}$ and $x^{(h)}_{1,m} \sim P^{(h)}_{1,m}$ as two random variables drawn from the two distributions. We hope that $\tilde{\mathbf{A}}$ could mitigate structural bias. We combine attribute and structural debiasing as below.

Based on the random variables $x^{(0)}_{0,m}$ to $x^{(H)}_{0,m}$ and $x^{(0)}_{1,m}$ to $x^{(H)}_{1,m}$, we have $(H+1)$-dimensional vectors $\mathbf{x}_{0,m} = [x^{(0)}_{0,m}, x^{(1)}_{0,m}, ..., x^{(H)}_{0,m}]$ and $\mathbf{x}_{1,m} = [x^{(0)}_{1,m}, x^{(1)}_{1,m}, ..., x^{(H)}_{1,m}]$ following the joint distribution $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$, respectively. To reduce both b_{attr} and b_{stru} at the m-th dimension, our goal is to minimize the Wasserstein distance between $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$, i.e., $\min_{\theta_m, \tilde{\mathbf{A}}} W(P^{Joint}_{0,m}, P^{Joint}_{1,m})$. $W(P^{Joint}_{0,m}, P^{Joint}_{1,m})$ can be expressed as

$$W(P^{Joint}_{0,m}, P^{Joint}_{1,m}) = \inf_{\gamma \in \Pi(P^{Joint}_{0,m}, P^{Joint}_{1,m})} \mathbb{E}_{(\mathbf{x}_{0,m}, \mathbf{x}_{1,m})\sim\gamma}[||\mathbf{x}_{0,m} - \mathbf{x}_{1,m}||_1]. \tag{4}$$

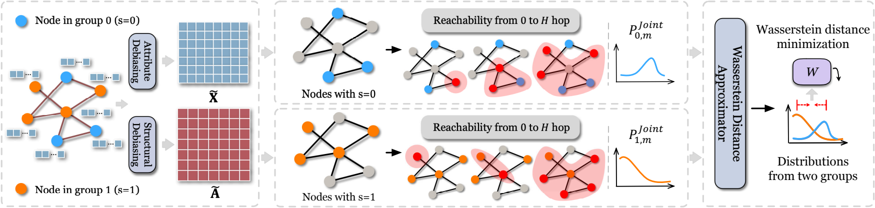


Figure 2: An illustration of EDITS with $H = 2$: Wasserstein Distance Approximator yields the approximated Wasserstein distance between $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$; Attribute Debiasing and Structural Debiasing are optimized towards less biased $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{A}}$.

Here $\Pi(P^{Joint}_{0,m}, P^{Joint}_{1,m})$ represents the set of all joint distributions $\gamma(\mathbf{x}_{0,m}, \mathbf{x}_{1,m})$ whose marginals are $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$, respectively. After considering all the M dimensions, the overall objective is

$$\min_{\theta, \tilde{\mathbf{A}}} \frac{1}{M} \sum_{1 \le m \le M} W(P^{Joint}_{0,m}, P^{Joint}_{1,m}). \tag{5}$$

It is non-trivial to optimize Eq. (5) as the infimum is intractable. Therefore, in the next subsection, we show how to convert it into a tractable optimization problem through approximation, which enables end-to-end gradient-based optimization.

4.3 Framework Optimization

In this subsection, we introduce our optimization algorithm. For simplicity, first we still use the m-th attribute dimension in $\mathbf{X}$ to illustrate the idea. Considering the infimum in Wasserstein distance computation is intractable, we apply the Kantorovich-Rubinstein duality [56] to convert the problem of Eq. (4) as:

$$\begin{aligned} W(P^{Joint}_{0,m}, P^{Joint}_{1,m}) &= \\ &\sup_{\|f\|_L \le 1} \mathbb{E}_{\mathbf{x}_{0,m} \sim P^{Joint}_{0,m}}[f(\mathbf{x}_{0,m})] - \mathbb{E}_{\mathbf{x}_{1,m} \sim P^{Joint}_{1,m}}[f(\mathbf{x}_{1,m})]. \end{aligned} \tag{6}$$

Here $\|f\|_L \le 1$ denotes that the supremum is taken over all 1-Lipschitz functions $f : \mathbb{R}^{H+1} \rightarrow \mathbb{R}$. The problem can be solved by learning a neural network as f. Nevertheless, it is worth noting that the 1-Lipschitz function is difficult to obtain during optimization. Therefore, here we relax $\|f\|_L \le 1$ to $\|f\|_L \le k$ (k is a constant). In this case, the left side of Eq. (6) also changes to $kW(P^{Joint}_{0,m}, P^{Joint}_{1,m})$. Then, the Wasserstein distance between $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$ up to a multiplicative constant can be attained via:

$$\max_{f_m \in \mathcal{F}} \mathbb{E}_{\mathbf{x}_{0,m} \sim P^{Joint}_{0,m}}[f_m(\mathbf{x}_{0,m})] - \mathbb{E}_{\mathbf{x}_{1,m} \sim P^{Joint}_{1,m}}[f_m(\mathbf{x}_{1,m})], \tag{7}$$

where $\mathcal{F}$ denotes the set of all k-Lipschitz functions (i.e., $\|f_m\|_L \le k$, $f_m \in \mathcal{F}$). Then, extending Eq. (7) to all M dimensions leads to our final objective function as:

$$\mathscr{L}_1 = \sum_{1 \le m \le M} \{\mathbb{E}_{\mathbf{x}_{0,m} \sim P^{Joint}_{0,m}}[f_m(\mathbf{x}_{0,m})] - \mathbb{E}_{\mathbf{x}_{1,m} \sim P^{Joint}_{1,m}}[f_m(\mathbf{x}_{1,m})]\}, \tag{8}$$

where $\{f_m : 1 \le m \le M\} \subset \mathcal{F}$. To model the function f in Eq. (8), a single-layered neural network serves as the *Wasserstein Distance Approximators* in Fig. (2) to approximate each f_m ($1 \le m \le M$), where the objective can be formulated as:

$$\max_{\{f_m : 1 \le m \le M\} \subset \mathcal{F}} \mathscr{L}_1 \ . \tag{9}$$

The weights of neural networks are clipped within $[-c, c]$ (c is a pre-defined constant), which has been proved to be a simple but effective way to enforce the Lipschitz constraint for every f_m [3]. For the *Attribute Debiasing* module in Fig. (2), we choose a linear function, i.e., $g_{\theta_m}(x_{s,m}) = \theta_m x_{s,m}$ ($s \in \{0, 1\}$). One advantage is that it acts as the role of feature re-weighting by assigning a feature weight for each attribute, which enables better interpretability for the debiased result. In matrix form, assume $\mathbf{\Theta}$ is a diagonal matrix with the m-th diagonal entry being θ_m, we have $\tilde{\mathbf{X}} = g_{\theta}(\mathbf{X}) = \mathbf{X}\mathbf{\Theta}$. Then the optimization goal for *attribute debiasing* is:

$$\min_{\mathbf{\Theta}} \mathscr{L}_1 + \mu_1 \|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2 + \mu_2 \|\mathbf{\Theta}\|_1, \tag{10}$$

where μ_1 and μ_2 are hyper-parameters. The second term ensures that the debiased attributes after feature re-weighting are close to the original ones (i.e., preserve as much information as possible). The third term controls the sparsity of re-weighting parameters. For the *Structural Debiasing* module in Fig. (2), $\tilde{\mathbf{A}}$ is optimized through:

$$\min_{\tilde{\mathbf{A}}} \mathscr{L}_1 + \mu_3 \|\tilde{\mathbf{A}} - \mathbf{A}\|_F^2 + \mu_4 \|\tilde{\mathbf{A}}\|_1 \quad s.t., \tilde{\mathbf{A}} = \tilde{\mathbf{A}}^\top. \tag{11}$$

where μ_3 and μ_4 are hyper-parameters. The second term ensures the debiased result $\tilde{\mathbf{A}}$ is close to the original structure $\mathbf{A}$. The third term enforces the debiased network structure is also sparse, which is aligned with the characteristics of real-world networks [23].

Optimization Strategy. To optimize function f, parameter $\mathbf{\Theta}$, and $\tilde{\mathbf{A}}$, we propose a gradient-based optimization approach for alternatively training as Algorithm 1 in Appendix. First, for the optimization of f w.r.t. Eq. (9), we directly utilize Stochastic Gradient Descent (SGD). Second, for the optimization of parameter $\mathbf{\Theta}$ w.r.t. Eq. (10), we adopt Proximal Gradient Descent (PGD). In the projection operation in PGD, we clip the parameters in $\mathbf{\Theta}$ within $[0, 1]$. Finally, to remove the most biased attribute channels, the z smallest weights in the diagonal of $\mathbf{\Theta}$ are masked with 0, where z is a pre-assigned hyper-parameter for attribute debiasing. Third, for the optimization of parameter $\tilde{\mathbf{A}}$ w.r.t. Eq. (11), we also adopt PGD with similar clipping strategy as the optimization of $\mathbf{\Theta}$. Finally, Algorithm 1 outputs $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{A}}$ after multiple epochs of optimization.

Edge Binarization. Here we introduce how we binarize the elements in $\tilde{\mathbf{A}}$ to indicate existence of edges. The basic intuition is to set a numerical threshold to determine the edge existence based on the entry-wise value change between $\tilde{\mathbf{A}}$ and $\mathbf{A}$. Specifically, for the "0" entries in $\mathbf{A}$, if the corresponding weight of any entry in $\tilde{\mathbf{A}}$ exceeds $r \cdot \max(\tilde{\mathbf{A}} - \mathbf{A})$, then we flip such entry from 0 to 1. Here r is a pre-set threshold for binarization, and $\max(\cdot)$ outputs the largest entry of a matrix. Similarly, for the "1" entries in $\mathbf{A}$, if the corresponding weight of any entry in $\tilde{\mathbf{A}}$ is reduced by a number exceeding $r \cdot |\min(\tilde{\mathbf{A}} - \mathbf{A})|$, then such entry should be flipped as 0. Here $\min(\cdot)$ gives the smallest entry of a matrix. To summarize, this operation aims to flip the entries with significant changes in value directly, and maintain other entries as their original values. Finally, the binarized matrix is assigned to $\tilde{\mathbf{A}}$ as the final outcome.

5 EXPERIMENTAL EVALUATIONS

In this section, we aim to answer the following research questions. **RQ1:** How well can EDITS mitigate the bias in attributed networks together with the outcome of different GNN variants for the downstream task? **RQ2:** How well can EDITS balance utility maximization and bias mitigation compared with other debiasing baselines tailored for a specific GNN?

5.1 Downstream Task and Datasets

Downstream Task. We choose the widely adopted *node classification* task to assess the effectiveness of our proposed framework.
Datasets. We use two types of datasets in our experiments, including six real-world datasets and two synthetic datasets. Statistics of the real-world datasets can be found in Table 3 of Appendix. We elaborate more details as follows: (1) *Real-world Datasets.* We use six real-world datasets, namely Pokec-z, Pokec-n [10, 50], UCSD34 [53], German Credit, Credit Defaulter, and Recidivism [2]. We first introduce the three web-related networks. *Pokec-z* and *Pokec-n* are collected from a popular social network in Slovakia. Here a node represents a user, and an edge denotes the friendship relation between two users [50]. We take "region" as the sensitive attribute, and the task is to predict the user working field. UCSD34 is a Facebook friendship network of the University of California San Diego [53]. Each node denotes a user, and edges represent the friendship relations between nodes. We take "gender" as the sensitive attribute, and the task is to predict whether a user belongs to a specific major. Users with incomplete information (e.g., missing attribute values) are filtered out from the three web networks above. Besides, we also adopt three networks beyond web-related data. In *German Credit*, nodes represent clients in a German bank, and edges are formed between clients if their credit accounts are similar. With "gender" being the sensitive attribute, the task is to classify the credit risk of the clients as high or low. In *Recidivism*, nodes are defendants released on bail during 1990-2009. Nodes are connected based on the similarity of past criminal records and demographics. The task is to classify defendants into bail vs. no bail, with "race" being the sensitive attribute. In the *Credit Defaulter*, nodes are credit card users, and they are connected based on the pattern similarity of their purchases and payments. Here "age" is the sensitive attribute, and the task is to predict whether a user will default on credit card payment. (2) *Synthetic Datasets.* For the ablation study of EDITS, we use the two datasets generated in Sec. 2. One network has biased attributes and an unbiased structure, while the other network is on the opposite. We add eight extra attribute dimensions besides the two attribute dimensions for both datasets. The attribute values in the extra attribute dimensions are generated uniformly between 0 and 1. For labels, we compute the sum of the first two extra attribute dimensions. Then, we add Gaussian noise to the sum values, and rank them by the values in descending order. Labels of the top-ranked 50% individuals are set as 1, while the labels of the other 50% are set as 0. The task is to predict the labels.

Table 1: Attribute and structural bias comparison between original networks and debiased ones from EDITS (in scale of $\times 10^{-3}$). The lower, the better. Best ones are marked in bold.

	Attribute Bias		Structural Bias	
	Vanilla	EDITS	Vanilla	EDITS
Pokec-z	0.43	**0.33** (−23.3%)	0.83	**0.75** (−9.64%)
Pokec-n	0.54	**0.42** (−22.2%)	1.03	**0.89** (−13.6%)
UCSD34	0.53	**0.48** (−9.43%)	0.68	**0.63** (−7.35%)
German	6.33	**2.38** (−62.4%)	10.4	**3.54** (−66.0%)
Credit	2.46	**0.56** (−77.2%)	4.45	**2.36** (−47.0%)
Recidivism	0.95	**0.39** (−58.9%)	1.10	**0.52** (−52.7%)

5.2 Experimental settings

GNN Models. Here we adopt three popular GNN variants in our experiments: GCN [28], GraphSAGE [19], and GIN [60].
Baselines. Since there is no existing work directly debiasing network data for GNNs, here we choose two state-of-the-art GNN-based debiasing approaches for comparison, namely FairGNN [10] and NIFTY [2]. (1) *FairGNN.* It is a debiasing method based on adversarial training. A discriminator is trained to distinguish the representations between different demographic groups. The goal of FairGNN is to train a GNN that fools the discriminator for bias mitigation. (2) *NIFTY.* It is a recently proposed GNN-based debiasing framework. With counterfactual perturbation on the sensitive attribute, bias is mitigated via learning node representations that are invariant to the sensitive attribute. It should be noted that both of them take GNNs as their backbones in the downstream task. While on the other hand, EDITS attempts at debiasing attributed networks without referring to the output of downstream GNN models (i.e., EDITS is model-agnostic). The hyper-parameters of EDITS are tuned only based on our proposed bias metrics. Obviously, the debiasing performed by EDITS generalizes better but is more difficult compared with the model-oriented baselines.
Evaluation Metrics. We evaluate model performance from two perspectives: model utility and bias mitigation. Good performance means low bias and high model utility. We introduce the adopted metrics for model utility and bias mitigation: (1) *Model Utility Metrics.* For node classification, we use the area under the receiver operating characteristic curve (AUC) and F1 score as the indicator of model utility; (2) *Bias Mitigation Metrics.* We use two widely-adopted metrics Δ_{SP} and Δ_{EO} to show to what extent the bias in the output of different GNNs are mitigated [5, 10, 38]. For both metrics, a lower value means better bias mitigation performance.

5.3 Debiasing Attributed Network for GNNs

To answer **RQ1**, we first evaluate the effectiveness of EDITS in reducing the bias measured by the two proposed metrics and traditional bias metrics with different GNN backbones. The attribute and structural bias of the six real-world datasets before and after being debiased by EDITS are shown in Table 1. The comparison on Δ_{SP} and Δ_{EO} between GNNs trained on debiased networks from EDITS and original networks is presented in Table 2. We make the following observations: (1) From the perspective of bias mitigation in the attributed network, EDITS demonstrates significant advantages over the vanilla approach as indicated by Table 1. This verifies the effectiveness of EDITS in reducing the bias existing

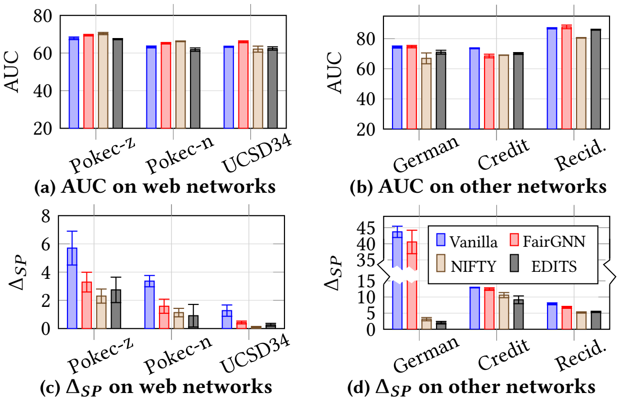
Table 2: Comparison on utility and bias mitigation between GNNs with original networks (denoted as Vanilla) and debiased networks (denoted as EDITS) as input. ↑ denotes the larger, the better; ↓ denotes the opposite. Best ones are in bold.

		GCN		GraphSAGE		GIN	
		Vanilla	EDITS	Vanilla	EDITS	Vanilla	EDITS
Pokec-z	**AUC ↑**	**67.83 ± 0.7%**	67.38 ± 0.3%	**68.00 ± 0.3%**	66.37 ± 0.7%	**66.74 ± 0.8%**	65.64 ± 0.5%
	F1 ↑	**61.95 ± 0.6%**	61.91 ± 0.1%	**61.58 ± 1.3%**	60.62 ± 0.6%	**61.55 ± 0.5%**	60.65 ± 1.2%
	Δ_{SP} ↓	5.70 ± 1.2%	**2.74 ± 0.9%**	7.10 ± 1.2%	**2.89 ± 0.4%**	5.20 ± 1.0%	**1.90 ± 1.3%**
	Δ_{EO} ↓	4.88 ± 1.3%	**2.87 ± 1.0%**	6.37 ± 0.8%	**2.54 ± 0.7%**	4.65 ± 1.1%	**2.09 ± 1.1%**
Pokec-n	**AUC ↑**	**63.24 ± 0.5%**	61.82 ± 0.9%	**64.07 ± 0.4%**	62.05 ± 0.6%	**62.53 ± 1.4%**	61.60 ± 1.4%
	F1 ↑	**54.32 ± 0.4%**	52.84 ± 0.3%	**53.45 ± 1.2%**	52.53 ± 0.1%	**52.62 ± 1.2%**	52.56 ± 1.0%
	Δ_{SP} ↓	3.36 ± 0.4%	**0.91 ± 0.87%**	3.85 ± 0.2%	**2.08 ± 1.2%**	5.90 ± 2.5%	**0.96 ± 0.5%**
	Δ_{EO} ↓	3.97 ± 1.6%	**1.10 ± 1.0%**	2.64 ± 0.3%	**1.82 ± 0.9%**	4.47 ± 3.7%	**0.47 ± 0.4%**
UCSD34	**AUC ↑**	**63.33 ± 0.3%**	62.43 ± 0.9%	62.62 ± 1.0%	**62.82 ± 2.4%**	62.57 ± 0.7%	**64.50 ± 0.9%**
	F1 ↑	94.16 ± 0.3%	**94.69 ± 0.1%**	94.00 ± 0.2%	**94.55 ± 0.1%**	92.24 ± 1.6%	**92.48 ± 0.5%**
	Δ_{SP} ↓	1.27 ± 0.4%	**0.27 ± 0.1%**	1.27 ± 0.5%	**0.35 ± 0.3%**	2.11 ± 1.3%	**0.36 ± 0.1%**
	Δ_{EO} ↓	1.40 ± 0.4%	**0.39 ± 0.1%**	1.40 ± 0.4%	**0.25 ± 0.3%**	2.32 ± 1.6%	**0.47 ± 0.4%**
German	**AUC ↑**	**74.46 ± 0.7%**	71.01 ± 1.3%	**75.28 ± 2.1%**	73.21 ± 0.5%	71.35 ± 1.7%	**71.51 ± 0.6%**
	F1 ↑	81.54 ± 0.9%	**82.43 ± 0.69%**	**81.52 ± 1.0%**	80.62 ± 1.5%	83.08 ± 0.9%	**83.78 ± 0.4%**
	Δ_{SP} ↓	43.14 ± 2.5%	**2.04 ± 1.3%**	26.83 ± 0.5%	**8.30 ± 3.1%**	18.55 ± 2.0%	**1.26 ± 0.7%**
	Δ_{EO} ↓	33.75 ± 0.4%	**0.63 ± 0.39%**	20.66 ± 3.0%	**3.75 ± 3.3%**	11.27 ± 3.5%	**2.87 ± 1.4%**
Credit	**AUC ↑**	**73.62 ± 0.3%**	70.16 ± 0.6%	74.99 ± 0.2%	**75.28 ± 0.5%**	**73.82 ± 0.4%**	72.06 ± 0.9%
	F1 ↑	**81.86 ± 0.1%**	81.44 ± 0.2%	82.31 ± 0.7%	**83.39 ± 0.3%**	82.11 ± 0.1%	**85.10 ± 0.7%**
	Δ_{SP} ↓	12.93 ± 0.1%	**9.13 ± 1.2%**	17.03 ± 3.3%	**12.25 ± 0.2%**	12.18 ± 0.3%	**8.79 ± 5.6%**
	Δ_{EO} ↓	10.65 ± 0.0%	**7.88 ± 1.0%**	15.31 ± 4.0%	**9.58 ± 0.1%**	9.48 ± 0.3%	**7.19 ± 3.8%**
Recidivism	**AUC ↑**	**86.91 ± 0.4%**	85.96 ± 0.3%	88.12 ± 1.4%	**88.15 ± 0.9%**	**82.40 ± 0.8%**	81.55 ± 1.5%
	F1 ↑	**78.30 ± 1.0%**	75.80 ± 0.5%	76.23 ± 2.8%	**76.30 ± 1.4%**	70.36 ± 1.9%	**71.09 ± 2.3%**
	Δ_{SP} ↓	7.89 ± 0.3%	**5.39 ± 0.2%**	2.42 ± 1.2%	**0.79 ± 0.5%**	9.97 ± 0.7%	**4.98 ± 0.9%**
	Δ_{EO} ↓	5.58 ± 0.2%	**3.36 ± 0.3%**	2.98 ± 2.2%	**1.01 ± 0.5%**	6.10 ± 1.2%	**5.47 ± 0.7%**

in the attributed network data. (2) From the perspective of bias mitigation in the downstream task, we observe from Table 2 that EDITS achieves desirable bias mitigation performance with little utility sacrifice in all cases compared with GNNs with the original network as input (i.e., the vanilla one). This verifies that attributed networks debiased by EDITS can generally mitigate the bias in the outcome of different GNNs. (3) When comparing bias mitigation performance indicated by Table 1 and Table 2, we can find that the bias in the outcome of GNNs is also mitigated after EDITS mitigates attribute bias and structural bias in the attributed networks. Such consistency verifies the validity of our proposed metrics on measuring the bias that existed in the attributed networks.

5.4 Comparison with Other Debiasing Models

To answer **RQ2**, we then compare the balance between model utility and bias mitigation with other baselines based on a given GNN. Here we present the comparison of AUC and Δ_{SP} based on GCN in Fig. (3). Similar results can be obtained for other GNNs, which are omitted due to space limit. Experimental results include the performance of baselines and EDITS on the six real-world datasets. The following observations can be made: (1) From the perspective of model utility (indicated by Fig. (3a) and Fig. (3b)), EDITS and baselines achieve comparable results with the vanilla GCN. This implies that the debiasing process of EDITS preserves as much useful information for the downstream task as the original attributed network. (2) From the perspective of bias mitigation (indicated by Fig. (3c) and Fig. (3d)), all baselines achieve effective bias mitigation. Compared with debiasing in downstream tasks, debiasing the attributed network is more difficult due to the lack of supervision signals from GNN prediction. Observation can be drawn that the debiasing performance of EDITS is similar to or even better than that of the adopted baselines. This verifies the superior performance of EDITS on debiasing attributed networks for more fair GNNs. (3) From the perspective of balancing the model utility and bias mitigation, EDITS achieves comparable model utility with alternatives but exhibits better bias mitigation performance. Consequently, we argue that EDITS achieves superior performance on balancing the model utility and bias mitigation over other baselines.



(a) AUC on web networks

(b) AUC on other networks

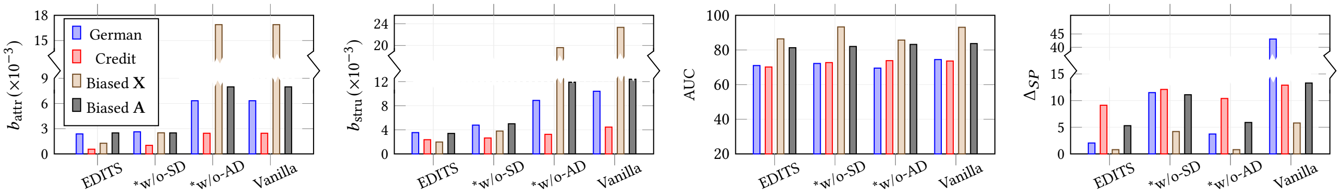
(c) Δ_{SP} on web networks

(d) Δ_{SP} on other networks

Figure 3: Performance comparison between EDITS and baselines on utility (AUC) and bias mitigation (Δ_{SP}).

5.5 Ablation Study

To evaluate the effectiveness of the two debiasing modules (i.e., attribute debiasing module and structural debiasing module) in EDITS, here we investigate how each of them individually contributes to bias mitigation under our proposed bias metrics and the traditional bias metrics in the downstream task. We choose GCN as the GNN model in our downstream task. For better visualization purposes, the two datasets showing large attribute bias and structural bias (i.e., *German* and *Credit*) are selected for experiments. Besides, to better demonstrate the functionality of the two debiasing modules, we also adopt the two synthetic datasets we mentioned in Sec. 2 (i.e., the network with only biased attributes and the network with only biased structure), which are further modified according to



(a) Attribute bias **(b) Structural bias** **(c) Model utility performance** **(d) Fairness performance**

Figure 4: Performance EDITS and its variants on two real-world datasets and two synthetic datasets. EDITS denotes that both debiasing modules are included; *w/o-SD means EDITS without structural debiasing module; *w/o-AD means EDITS is without attribute debiasing module; Vanilla means applying GNN with the original attributed network as input.

Sec. 5.1. Based on the four selected datasets, four different variants of EDITS are tested, namely EDITS with both debiasing modules, EDITS without the structural debiasing module (i.e., *w/o-SD), EDITS without the attribute debiasing module (i.e., *w/o-AD), vanilla GCN model without debiased input (i.e., Vanilla). We present their performance of attribute bias, structural bias, AUC, and Δ_{SP} on the four datasets in Fig. (4). We make the following observations: (1) The value of *attribute bias* can be reduced with the attribute debiasing module of EDITS, which maintains the model utility (i.e., AUC) but reduces Δ_{SP} in the downstream task. (2) The value of *structural bias* can be reduced with both attribute debiasing and structural debiasing modules. With only structural debiasing, EDITS still maintains comparable model utility but reduces Δ_{SP} in the downstream task. (3) Although both attribute debiasing and structural debiasing module help mitigate *structural bias*, only debiasing the network structure achieves better bias mitigation performance on all four datasets compared with only debiasing the attributes as implied by Fig. (4d). This demonstrates the indispensability of the structural debiasing module in EDITS.

6 RELATED WORK

Mitigating Bias in Machine Learning. Bias can be defined from a variety of perspectives in machine learning algorithms [4, 13, 20, 35, 39, 58]. Commonly used algorithmic bias notions can be broadly categorized into *group fairness* and *individual fairness* [15]. Group fairness emphasizes that algorithms should not yield discriminatory outcomes for any specific demographic groups [15]. Such groups are usually determined by sensitive attributes, e.g., gender or race [25]. Existing debiasing approaches work in one of the three data flow stages, i.e., pre-processing, processing and post-processing stage. In pre-processing stage, a common method is to re-weight training samples from different groups to mitigate bias before model training [25]. Perturbing data distributions between groups is another popular approach to debias the data in the pre-processing stage [57]. In processing stage, a popular method is to add regularization terms to disentangle the outcome from sensitive attribute [36, 48] or minimize the outcome difference between groups [1]. Besides, utilizing adversarial learning to remove sensitive information from representations is also widely adopted [17]. In post-processing stage, bias in outcomes is usually mitigated by constraining the outcome to follow a less biased distribution [20, 30, 33, 44, 64]. Usually, all above-mentioned approaches are evaluated via measuring how much certain fairness notion is violated. *Statistical Parity* [15], *Equality of Opportunity*, *Equality of Odds* [20] and *Counterfactual Fairness* [31] are commonly studied fairness notions. Different from group fairness, individual fairness focuses on treating similar individuals similarly [15, 62]. The similarity can be given by oracle similarity scores from domain experts [32]. Most existing debiasing methods based on individual fairness work in the processing stage. For example, constraints can enforce similar predictions between similar instances [24, 32]. *Consistency* is a popular metric for individual fairness evaluation [32, 34].

Mitigating Bias in Graph Mining. Efforts have been made to mitigate bias in graph mining algorithms, where these works can be broadly categorized into either focusing on *group fairness* or *individual fairness*. For group fairness, adversarial learning can be adopted to learn less biased node representations that fool the discriminator [6, 10]. Rebalancing between groups is also a popular approach to mitigate bias [7, 18, 35, 46, 52]. For example, Rahman et al. mitigate bias via rebalancing the appearance rate of minority groups in random walks [47]. Projecting the embeddings onto a hyperplane orthogonal to the hyperplane of sensitive attributes is another approach for bias mitigation [41]. Compared with the vast amount of works on group fairness, only few works promote individual fairness in graphs. To the best of our knowledge, Kang et al. [26] first propose to systematically debias multiple graph mining algorithms based on individual fairness. Dong et al. [11] argue that for each individual, the similarity ranking of others in the GNN outcome should follow the same order of an oracle ranking from domain experts. Different from these approaches, this paper proposes to debias attributed networks in a model-agnostic way.

7 CONCLUSION

GNNs are increasingly critical in various applications. Nevertheless, there is an increasing societal concern that GNNs could yield discriminatory decisions towards certain demographic groups. Existing debiasing approaches are mainly tailored for a specific GNN. Adapting these methods to different GNNs can be costly, as they need to be fine-tuned. Different from them, in this paper, we propose to debias the attributed network for GNNs. With analysis of the source of bias existing in different data modalities, we define two kinds of bias with corresponding metrics, and formulate a novel problem of debiasing attributed networks for GNNs. To tackle this problem, we propose a principled framework EDITS for model-agnostic debiasing. Experiments demonstrate the effectiveness of EDITS in mitigating bias and maintaining model utility.

8 ACKNOWLEDGMENTS

This material is supported by the National Science Foundation (NSF) under grants #2006844 and the Cisco Faculty Research Award.

REFERENCES

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*. PMLR, 60–69.

[2] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. 2021. Towards a Unified Framework for Fair and Stable Graph Representation Learning. *arXiv preprint arXiv:2102.13186* (2021).

[3] Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. *CoRR* abs/1701.07875 (2017). arXiv:1701.07875 http://arxiv.org/abs/1701.07875

[4] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2017. Fairness in machine learning. *NeurIPS Tutorial* 1 (2017).

[5] 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).

[6] Avishek Joey Bose and William L. Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *Proceedings of the 36th ICML (ICML '19)*.

[7] R. Burke, N. Sonboli, M. Mansoury, and A. Ordoñez-Gauger. 2017. Balanced neighborhoods for fairness-aware collaborative recommendation. (2017).

[8] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. 2009. Building classifiers with independency constraints. In *2009 IEEE ICDM Workshops*. IEEE, 13–18.

[9] F. Chung and Fan C. Graham. 1997. *Spectral graph theory*. Number 92. American Mathematical Soc.

[10] Enyan D. and Suhang W. 2020. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. arXiv:2009.01454

[11] Y. Dong, J. Kang, H. Tong, and J. Li. 2021. Individual Fairness for Graph Neural Networks: A Ranking based Approach. In *Proceedings of the 27th SIGKDD (KDD '21)*. 300–310.

[12] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing Graph Neural Network-based Fraud Detectors Against Camouflaged Fraudsters. In *Proceedings of the 29th CIKM (CIKM '20)*. 315–324.

[13] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. 2019. Fairness in deep learning: A computational perspective. *arXiv preprint arXiv:1908.08843* (2019).

[14] C. Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd ITCS (ITCS '12)*.

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*.

[16] Michael D Ekstrand and Daniel Kluver. 2021. Exploring author gender in book rating and recommendation. *User Modeling and User-Adapted Interaction* (2021).

[17] Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. *arXiv preprint arXiv:1808.06640* (2018).

[18] Golnoosh F., Pigi K., Spencer K. T., Sriram S., and Lise G. 2018. A Fairness-aware Hybrid Recommender System. *CoRR* abs/1809.09030 (2018).

[19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st NeurIPS (NIPS '17)*.

[20] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Proceedings of the 30th NeurIPS (NeurIPS '16)*.

[21] G. Hinton, N. Srivastava, and K. Swersky. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. (2012).

[22] Zeinab S Jalali, Weixiang Wang, Myunghwan Kim, Hema Raghavan, and Sucheta Soundarajan. 2020. On the Information Unfairness of Social Networks. In *Proceedings of the 20th ICDM (ICDM '20)*. SIAM, 613–521.

[23] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang. 2020. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th SIGKDD (KDD '20)*. 66–74.

[24] Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, Logan Stapleton, and Zhiwei Steven Wu. 2019. Eliciting and enforcing subjective individual fairness. *arXiv preprint arXiv:1905.10660* (2019).

[25] F. Kamiran and T. Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2012), 1–33.

[26] J. Kang, J. He, R. Maciejewski, and H. Tong. 2020. InFoRM: Individual Fairness on Graph Mining. In *Proceedings of the 26th SIGKDD (KDD '20)*.

[27] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd ICLR (ICLR '15)*.

[28] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[29] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[30] E. Krasanakis, S. Papadopoulos, and I. Kompatsiaris. 2020. Applying Fairness Constraints on Graph Node Ranks Under Personalization Bias. In *International Conference on Complex Networks and Their Applications*. Springer.

[31] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *Advances in Neural Information Processing Systems*. 4066–4076.

[32] Preethi L., K. P. Gummadi, and G. Weikum. 2019. Operationalizing Individual Fairness with Pairwise Fair Representations. *Proc. VLDB Endow.* (2019).

[33] Charlotte Laclau, Ievgen Redko, Manvi Choudhary, and Christine Largeron. 2021. All of the Fairness for Edge Prediction with Optimal Transport. In *Proceedings of the 24th AISTATS (AISTATS '21)*. PMLR, 1774–1782.

[34] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. 2019. ifair: Learning individually fair data representations for algorithmic decision making. In *Proceedings of the 35th ICDE (ICDE '19)*. IEEE.

[35] Peizhao Li, Yifei Wang, Han Zhao, Pengyu Hong, and Hongfu Liu. 2021. On dyadic fairness: Exploring and mitigating bias in graph connections. In *Proceedings of International Conference on Learning Representations*.

[36] Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. *arXiv preprint arXiv:1906.08286* (2019).

[37] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection. In *Proceedings of the 31th WWW (WWW '21)*. 3168–3177.

[38] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. 2015. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830* (2015).

[39] Ninareh M., Fred M., Nripsuta S., Kristina L., and Aram G. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* (2019).

[40] Farzan Masrour, Tyler Wilson, Heng Yan, Pang-Ning Tan, and Abdol Esfahanian. 2020. Bursting the Filter Bubble: Fairness-Aware Network Link Prediction. In *AAAI Conference on Artificial Intelligence*. 841–848.

[41] John Palowitch and Bryan Perozzi. 2019. MONET: Debiasing Graph Embeddings via the Metadata-Orthogonal Training Unit. *CoRR* abs/1909.11793 (2019).

[42] Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin.

[43] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[44] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. *arXiv preprint arXiv:1709.02012* (2017).

[45] Y. Qian, P. Zhao, Z. Li, J. Fang, L. Zhao, V. Sheng, and Z. Cui. 2020. Interaction graph neural network for news recommendation. In *International Conference on Web Information Systems Engineering*. Springer, 599–614.

[46] Tahleen A. R., Bartlomiej S., Michael B., and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *Proceedings of the 28th IJCAI (IJCAI '19)*.

[47] T. A Rahman, B. Surma, M. Backes, and Y. Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *Proceedings of the 28th IJCAI (IJCAI '19)*.

[48] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717* (2017).

[49] V. Shumovskaia, K. Fedyanin, I. Sukharev, D. Berestnev, and M. Panov. 2021. Linking bank clients using graph neural networks powered by rich transactional data. *International Journal of Data Science and Analytics* (2021), 1–11.

[50] L. Takac and M. Zabovsky. 2012. Data analysis in public social networks. In *International scientific conference and international workshop*, Vol. 1.

[51] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th SIGKDD (KDD '09)*.

[52] X. Tang, H. Yao, Y. Sun, Y. Wang, J. Tang, C. Aggarwal, P. Mitra, and S. Wang. 2020. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In *Proceedings of the 29th CIKM (CIKM '20)*.

[53] Amanda L Traud, Peter J Mucha, and Mason A Porter. 2012. Social structure of Facebook networks. *Physica A: Statistical Mechanics and its Applications* 391, 16 (2012), 4165–4180.

[54] C. V. 2021. *Topics in optimal transportation*. American Mathematical Soc.

[55] Petar V., Guillem C., Arantxa C., Adriana R., Pietro L., and Yoshua B. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[56] Cédric Villani. 2008. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

[57] Hao Wang, Berk Ustun, and Flavio Calmon. 2019. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *International Conference on Machine Learning*. PMLR, 6618–6627.

[58] Y. Wu, L. Zhang, and X. Wu. 2019. Counterfactual Fairness: Unidentification, Bound and Algorithm. In *International Joint Conference on Artificial Intelligence*.

[59] Bingbing Xu, Huawei Shen, Bingjie Sun, Rong An, Qi Cao, and Xueqi Cheng. 2021. Towards Consumer Loan Fraud Detection: Graph Neural Networks with Role-Constrained Conditional Random Field. (2021).

[60] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[61] Zhiju Yang, Weiping Pei, Monchu Chen, and Chuan Yue. 2022. WTAGRAPH: Web Tracking and Advertising Detection using Graph Neural Networks. In *IEEE Symposium on Security and Privacy*.

[62] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *Proceedings of the 30th ICML (ICML '13)*.

[63] Muhan Zhang and Yixin Chen. 2018. Link Prediction Based on Graph Neural Networks. In *Proceedings of the 32nd NeurIPS (NIPS '18)*. 5171–5181.

[64] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457* (2017).

A APPENDIX

A.1 Datasets Statistics

The detailed statistics of six real-world datasets (i.e., German Credit, Recidivism and Credit Defaulter) can be found in Table 3.

A.2 Algorithm

We present the optimization algorithm for EDITS in Algorithm 1.

Algorithm 1 The Optimization Algorithm for EDITS

```
Input:
    A: Adjacency matrix; X: Attribute matrix; $\alpha$, $\mu_1$ to $\mu_4$: Hyper-parameters in objectives; $c$: Threshold enforcing Lipschitz; $z$: Threshold for attribute masking; $r$: Threshold factor for adjacency matrix binarization;
Output:
    Debiased adjacency matrix $\tilde{\mathbf{A}}$ and attribute matrix $\tilde{\mathbf{X}}$;
$\tilde{\mathbf{A}} \leftarrow \mathbf{A}$; $\Theta \leftarrow \mathbf{I}$;
while $epoch \leq epoch\_max$ do
    Compute $\mathscr{L}_1$ following Eq. (8);
    Update the weights of $f$ by SGD following Eq. (9);
    Clip the weights of $f$ within [-$c$, $c$];
    Update $\Theta$ by PGD following Eq. (10), $\tilde{\mathbf{X}} \leftarrow \mathbf{X}\Theta$;
    Update $\tilde{\mathbf{A}}$ by PGD following Eq. (11), $\tilde{\mathbf{A}} \leftarrow \frac{1}{2}(\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^\top)$;
end while
Mask the $z$ smallest entries with 0 in $diag(\Theta)$, $\tilde{\mathbf{X}} \leftarrow \mathbf{X}\Theta$;
Binarize $\tilde{\mathbf{A}}$ w.r.t. the threshold $r$;
return $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{X}}$;
```

A.3 Theoretical Analysis

Here we present theoretical analysis for the two proposed metrics to gain a deeper understanding of debiasing attributed networks for GNNs. For attribute bias, it is straightforward that if the Wasserstein distance of the attribute value distribution between the two groups is zero for every dimension, then there would be no clue to distinguish between the two groups. Consequently, here we mainly focus on the theoretical analysis of the structural bias metric. Specifically, we perform theoretical analysis from the perspective of Spectral Graph Theory [9]. Usually, an undirected attributed network is regarded as a signal composed of different frequency components in Graph Signal Processing (GSP). If an operation preserves lower frequency components more than higher ones of a graph signal, this operation low-pass filters the input graph signal.

Theorem 1. *Let λ_{max} be the largest eigenvalue of $\mathbf{L}_{norm}$. Multiplying $\mathbf{X}$ by the propagation matrix $\mathbf{M}_H$ can be regarded as low-pass filtering $\mathbf{X}$ when $\alpha = \frac{1}{\lambda_{max}}$ and $\beta_i > 0$ ($1 \leq i \leq H$).*

Proof. We present the proof based on Laplacian graph spectrum. By replacing α with $\frac{1}{\lambda_{\max}}$, we have

$$\mathbf{P}_{\text{norm}} = \frac{1}{\lambda_{\max}}\mathbf{A}_{\text{norm}} + (1 - \frac{1}{\lambda_{\max}})\mathbf{I} = \mathbf{I} - \frac{\mathbf{L}_{\text{norm}}}{\lambda_{\max}}. \tag{12}$$

Then, by combining Eq. (2) and Eq. (12), we get

$$\mathbf{M}_H = \beta_1(\mathbf{I} - \frac{\mathbf{L}_{\text{norm}}}{\lambda_{\max}}) + \beta_2(\mathbf{I} - \frac{\mathbf{L}_{\text{norm}}}{\lambda_{\max}})^2 + ... + \beta_H(\mathbf{I} - \frac{\mathbf{L}_{\text{norm}}}{\lambda_{\max}})^H. \tag{13}$$

Considering that $\mathbf{L}_{\text{norm}}$ is a symmetric real matrix, it can be decomposed as $\mathbf{L}_{\text{norm}} = \mathbf{U}\Lambda\mathbf{U}^\top$, then Eq. (13) can be rewritten as

$$\mathbf{M}_H = \mathbf{U}(\beta_1(\mathbf{I} - \frac{\Lambda}{\lambda_{\max}}) + \beta_2(\mathbf{I} - \frac{\Lambda}{\lambda_{\max}})^2 + ... + \beta_H(\mathbf{I} - \frac{\Lambda}{\lambda_{\max}})^H)\mathbf{U}^\top. \tag{14}$$

Here Λ is the diagonal eigenvalue matrix of $\mathbf{L}_{\text{norm}}$, and the h-th term ($1 \leq h \leq H$) in Eq. (14) indicates a frequency response function of $(1 - \frac{\lambda_{\text{i}}}{\lambda_{\max}})^h$. For any λ_i ($1 \leq i \leq N$), $\frac{\lambda_i}{\lambda_{\max}} \leq 1$ holds. Consequently, the frequency response of each term in Eq. (14) monotonically decreases w.r.t. λ_i. This indicates that, for each term, when it is multiplied by a graph signal, the higher frequency components of the graph signal are more weakened compared with the lower frequency components. Therefore, according to Eq. (14), $\mathbf{M}_H$ can be regarded as a graph filter whose frequency response is composed of H low-pass filters. In conclusion, multiplying the propagation matrix $\mathbf{M}_H$ with any graph signal equals to the operation of low-pass filtering when $\alpha = \frac{1}{\lambda_{\max}}$ and all $\beta_i > 0$. The graph signal is the attribute matrix $\mathbf{X}$ in the proposed structural bias metric. □

Based on Theorem 1, we propose the corollary below to build connections between attribute bias and structural bias.

Corollary 1. *The attribute bias contained in the low frequency components of an attributed network is equivalent to structural bias.*

From the proof of Theorem 1, we can observe that $\mathbf{M}_H\mathbf{X}_{\text{norm}}$ is equivalent to low-pass filtering the attribute matrix $\mathbf{X}_{\text{norm}}$. Then Corollary 1 is self-evident based on Definition 2. At the same time, considering that the frequencies and the corresponding basis of a network data changes when $\mathbf{A}$ is optimized to be $\tilde{\mathbf{A}}$. The basic goal of EDITS can also be interpreted as: *debiasing the full spectrum of a graph signal, and learning better frequencies together with the corresponding basis to further mitigate the bias existed in the lower frequency components of the graph signal.*

A.4 Implementation Details

EDITS is implemented using Pytorch [43] and optimized via RMSprop optimizer [21]. In the training of EDITS, we set the training epochs as 100 for Recidivism and 500 for other datasets. The learning rate is set as 3×10^{-3} for epochs under 400 and 1×10^{-3} for those above. α is set as 0.5 considering that $\lambda_{\max} = 2$ [9]. To train GNNs, we fix the training epochs to be 1,000 based on Adam optimizer [27], with the learning rate of 1×10^{-3}.

A.5 Extension to Non-Binary Sensitive Attributes

Here, we show how our proposed framework EDITS can be generalized to handle non-binary sensitive attributes. More specifically, we use a synthetic dataset to showcase the extension.

Synthetic Dataset Generation. Our goal here is to generate a synthetic attributed network with both biased node attributes and network structure, where nodes should come from at least three different groups based on the sensitive attribute. We elaborate more details from three perspectives: biased network structure generation, biased node attribute generation, and node label generation. (1) *Biased Network Structure Generation.* We adopt a similar approach as presented in Fig. (1) to generate three communities with dense intra-community links but sparse inter-community links. (2) *Biased Node Attributes Generation.* We generate a ten-dimensional attribute vector for each node. The values at the first two dimensions are generated independently with Gaussian distribution $\mathcal{N}$(-1, 1^2), $\mathcal{N}$(0, 1^2), and $\mathcal{N}$(1, 1^2) for the nodes in the three communities, respectively. The attribute values for all other dimensions are generated with independent Gaussian Distribution $\mathcal{N}$(0, 1^2). Besides, We generate a ternary variable $s \in \{0, 1, 2\}$ based on the node community membership for all nodes as an extra attribute dimension. Here the

Table 3: The statistics and basic information about the six real-world datasets adopted for experimental evaluation. Sens. represents the semantic meaning of sensitive attribute.

Dataset	Pokec-z	Pokec-n	UCSD34	German Credit	Recidivism	Credit Defaulter
# Nodes	7,659	6,185	4,132	1,000	18,876	30,000
# Edges	29,476	21,844	108,383	22,242	321,308	1,436,858
# Attributes	59	59	7	27	18	13
Avg. degree	7.70	7.06	52.5	44.5	34.0	95.8
Sens.	Region	Region	Gender	Gender	Race	Age
Label	Working field	Working field	Student major	Credit status	Bail decision	Future default

community membership is regarded as the sensitive attribute of nodes in this network. (3) *Node Label Generation.* We sum up the values at the first two unbiased attribute dimensions for all nodes, and then add Gaussian noise to the summation. The summation values with noise are ranked in descending order. Labels of the top-ranked 50% nodes are set as 1, while the labels of the other 50% nodes are set as 0. The task is to predict the labels.

Framework Extension. To extend the proposed framework EDITS to handle non-binary sensitive attributes, the basic rationale is to encourage the function f_m introduced in Section 4.3 to help approximate the squared Wasserstein distance sum between all group pairs based on ternary sensitive attribute. Therefore, we modify the $\mathscr{L}_1$ in Eq. (8) as

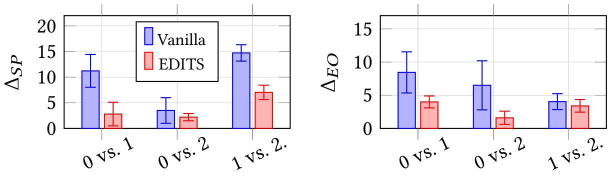
$$\tilde{\mathscr{L}}_1 = \sum_{i,j}\sum_{m}\{\mathbb{E}_{\mathbf{x}_{i,m}}[f_m(\mathbf{x}_{i,m})] - \mathbb{E}_{\mathbf{x}_{j,m}}[f_m(\mathbf{x}_{j,m})]\}^2. \tag{15}$$

Here $1 \leq m \leq M$, and $i, j \in \{0, 1, 2\}$ $(i < j)$. $\mathbf{x}_{i,m}$ and $\mathbf{x}_{j,m}$ follows $P^{Joint}_{i,m}$ and $P^{Joint}_{j,m}$, respectively. The $\mathscr{L}_1$ in Eq. (9), (10), and (11) are repalced with $\tilde{\mathscr{L}}_1$. This enables EDITS to minimize the squared Wasserstein distance sum between all group pairs.

Research Questions. Here we aim to answer two research questions. **RQ1:** Can EDITS mitigate the bias in the network dataset with ternary sensitive attributes? **RQ2:** Can EDITS achieve a good balance between mitigating bias and maintaining utility for GNN predictions with ternary sensitive attributes?

Evaluation Metrics. We introduce the metrics following the two research questions above. (1) For RQ1, to measure the bias in the network dataset, we adopt the b_{attr} and b_{stru} introduced in Sec. 3.3. (2) For RQ2, to measure the bias exhibited in GNN predictions, we adopt two traditional fairness metrics: Δ_{SP} and Δ_{EO}. Considering that these two metrics are designed only for binary sensitive attributes, Δ_{SP} and Δ_{EO} for each pair of groups are utilized to evaluate the fairness level of GNN predictions. Besides, AUC and F1 are adopted to evaluate the utility of GNN predictions.

Results Analysis. Results based on GCN are presented in Fig. (5) and Table 5, and similar observations can also be found on other GNN backbones. We evaluate the performance of EDITS from two perspectives. (1) *RQ1: the fairness level of the network dataset.* As presented in Table 5, b_{attr} and b_{stru} of the dataset are clearly reduced with EDITS. This verifies the effectiveness of EDITS on debiasing the attributed network data. (2) *RQ2: the balance between fairness and utility for GNN predictions.* As presented in Fig. (5), Δ_{SP} and Δ_{EO} for every group pair are reduced. This corroborates the effectiveness of EDITS on achieving more fair GNN predictions. At the same time, Table 5 indicates that the GNN with debiased input data still maintains similar utility performance compared with the GNN with vanilla input. This indicates that EDITS achieves a good balance between fairness and utility for GNN predictions.



(a) Comparison of Δ_{SP} **(b) Comparison of Δ_{EO}**

Figure 5: Comparison of Δ_{SP} and Δ_{EO} between vanilla and EDITS based on GCN for ternary sensitive attributes.

Table 4: Parameter study for μ_1 and μ_3. The values of b_{attr} and b_{stru} are in scale of $\times 10^{-3}$.

μ_1	b_{attr}	F1(%)	Δ_{SP}(%)	μ_3	b_{stru}	F1(%)	Δ_{SP}(%)
1e2	6.33	81.69	35.3	**1e2**	10.2	82.26	34.2
1e1	5.02	80.69	19.9	**1e1**	9.97	80.89	25.0
1e0	3.74	80.28	7.76	**1e0**	9.81	79.77	14.1
1e-1	2.38	80.00	4.58	**1e-1**	4.89	79.46	3.96
1e-2	2.34	79.95	4.08	**1e-2**	3.53	78.93	3.26
1e-3	2.35	79.46	3.96	**1e-3**	3.34	78.89	2.76
1e-4	2.34	79.03	3.29	**1e-4**	3.29	78.37	2.06
1e-5	2.34	76.22	2.86	**1e-5**	3.22	78.06	2.00

Table 5: Comparison of fairness level and utility between the original synthetic network and the debiased one based on the ternary sensitive attributes. The values of b_{attr} and b_{stru} are in scale of $\times 10^{-3}$. Best ones are marked in bold.

	Attribute Bias & Structural Bias Comparison					
	Group 0 v.s. 1		Group 0 v.s. 2		Group 1 v.s. 2	
	b_{attr}	b_{stru}	b_{attr}	b_{stru}	b_{attr}	b_{stru}
Vanilla	13.7	25.5	26.5	48.8	11.0	20.4
EDITS	**5.33**	**9.63**	**13.4**	**24.1**	**4.73**	**8.73**
	Utility Comparison					
	AUC			F1		
Vanilla	**67.09 ± 0.3%**			**64.50 ± 0.6%**		
EDITS	67.05 ± 0.2%			62.91 ± 0.8%		

A.6 Parameter Study

Here we aim to study the sensitivity of EDITS. Specifically, we show the parameter study of μ_1 and μ_3 on German dataset, but similar observations can also be found on other datasets. Here μ_1 and μ_3 control how much original information should be preserved from the original attributes and graph structure, respectively. We first vary μ_1 in the range of {1e2, 1e1, 1e0, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5} while fix other parameters as μ_2=1e-4, μ_3=1e-1, μ_4=1e-4; then we vary μ_1 in the same range with μ_1=1e-3, μ_2=1e-4, μ_4=1e-4. The results in Table 4 indicate that the trade-off between debiasing and utility performance is stable when μ_1 and μ_3 are in a wide range between 1e-3 and 1e-1. Therefore, it is safe to say that we can tune these parameters in a wide range without greatly affecting the fairness and model utility.