Chasing Fairness in Graphs: A GNN Architecture Perspective

Zhimeng Jiang[1], **Xiaotian Han**[1], **Chao Fan**[2], **Zirui Liu**[3], **Na Zou**[4], **Ali Mostafavi**[1], **Xia Hu**[3]

[1]Texas A&M University, [2]Clemson University, [3]Rice University, [4]University of Houston,
{zhimengj,han, amostafavi}@tamu.edu, cfan@g.clemson.edu, {zl105, xia.hu}@rice.edu, nzou2@central.uh.edu

Abstract

There has been significant progress in improving the performance of graph neural networks (GNNs) through enhancements in graph data, model architecture design, and training strategies. For fairness in graphs, recent studies achieve fair representations and predictions through either graph data pre-processing (e.g., node feature masking, and topology rewiring) or fair training strategies (e.g., regularization, adversarial debiasing, and fair contrastive learning). How to achieve fairness in graphs from the model architecture perspective is less explored. More importantly, GNNs exhibit worse fairness performance compared to multilayer perception since their model architecture (i.e., neighbor aggregation) amplifies biases. To this end, we aim to achieve fairness via a new GNN architecture. We propose Fair Message Passing (FMP) designed within a unified optimization framework for GNNs. Notably, FMP *explicitly* renders sensitive attribute usage in *forward propagation* for node classification task using cross-entropy loss without data pre-processing. In FMP, the aggregation is first adopted to utilize neighbors' information and then the bias mitigation step explicitly pushes demographic group node presentation centers together. In this way, FMP scheme can aggregate useful information from neighbors and mitigate bias to achieve better fairness and prediction tradeoff performance. Experiments on node classification tasks demonstrate that the proposed FMP outperforms several baselines in terms of fairness and accuracy on three real-world datasets. The code is available in https://github.com/zhimengj0326/FMP.

Introduction

Graph neural networks (GNNs) (**??????**) are widely adopted in various domains, such as social media mining (**?**), knowledge graph (**?**) and recommender system (**?**), due to remarkable performance in learning representations. Graph learning, a topic with growing popularity, aims to learn node representation containing both topological and attribute information in a given graph. Despite the outstanding performance in various tasks, GNNs often inherit or even amplify societal bias from input graph data (**?**). The biased node representation largely limits the application of GNNs in many high-stake tasks, such as job hunting (**?**) and crime ratio prediction (**?**). Hence, bias mitigation that facilitates the research on fair GNNs is in urgent need and we aim to achieve fair prediction for GNNs.

Data, model architecture, and training strategy are the most popular aspects to improve deep learning performance. For fairness in graphs, many existing works achieving fair prediction in graphs either rely on graph pre-processing (e.g., node feature masking(**?**), and topology rewiring (**?**)) or fair training strategies (e.g., regularization (**?**), adversarial debiasing (**?**), or contrastive learning (**????**)). The GNNs architecture perspective to improve fairness in graphs is less explored. More importantly, GNNs are notorious in terms of fairness since GNN aggregation amplifies bias compared to multilayer perception (MLP) (**?**). From the GNNs architecture perspective, message passing is a critical component to improve fairness in graphs. Therefore, a natural question is raised:

Can we achieve fairness via fair message passing using vanilla training loss [1] *without graph pre-processing?*

In this work, we provide a positive answer by designing a fair message-passing scheme guided by a unified optimization framework [2] for GNNs. The key idea of achieving fair message passing is aggregation first and then conducting bias mitigation via explicitly chasing consistent demographic group representation centers. Specifically, we first formulate an optimization problem that integrates fairness and smoothness objectives for graph data. Then, we solve the formulated problem via Fenchel conjugate and gradient descent to generate fair and informative representations, where the property of softmax function is adopted to accelerate the gradient calculation over primal variables. We also interpret the optimization problem solver as two main steps (e.g., aggregation first and then debiasing). Finally, we integrate FMP in graph neural networks to achieve fair and accurate prediction for node classification tasks. We demonstrate the superiority of FMP by examining its effectiveness and efficiency on various real-world datasets.

[1] The sensitive attributes are not adopted in vanilla training loss. We only consider node classification tasks and vanilla loss is cross-entropy loss in this paper.

[2] Many aggregations in popular GNNs can be interpreted as gradient descent step for specific optimization problem with specific step size and initialization (**??**).

In short, the contributions can be summarized as follows:

- We demonstrate proof-of-concept that a meticulously crafted GNN architecture can improve fairness for graph data. Our work offers a fresh outlook in comparison to conventional approaches that focus on data preprocessing and fair training strategy design.
- We propose FMP to achieve fairness via explicitly incorporating sensitive attribute information in message passing, guided by a unified optimization framework. Additionally, we introduce an acceleration method based on softmax property to reduce gradient computational complexity.
- The effectiveness and efficiency of FMP are experimentally evaluated on three real-world datasets. The results show that compared to the state-of-the-art, our FMP exhibits a comparable or superior trade-off between prediction performance and fairness with negligibly computation overhead.

Preliminaries

Notations

We adopt bold upper-case letters to denote matrix such as $\mathbf{X}$, bold lower-case letters such as $\mathbf{x}$ to denote vectors, and calligraphic font such as $\mathcal{X}$ to denote sets. Given a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, the i-th row and j-th column are denoted as $\mathbf{X}_i$ and $\mathbf{X}_{\cdot,j}$, and the element in i-th row and j-th column is $\mathbf{X}_{i,j}$. We use the Frobenius norm, l_1 norm of matrix $\mathbf{X}$ as $||\mathbf{X}||_F = \sqrt{\sum_{i,j}\mathbf{X}_{i,j}^2}$ and $||\mathbf{X}||_1 = \sum_{ij}|\mathbf{X}_{ij}|$, respectively. Given two matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times d}$, the inner product is defined as $\langle \mathbf{X}, \mathbf{Y}\rangle = tr(\mathbf{X}^\top \mathbf{Y})$, where $tr(\cdot)$ is the trace of a square matrix. $SF(\mathbf{X})$ represents softmax function with a default normalized column dimension. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph with the node set $\mathcal{V} = \{v_1, \cdots, v_n\}$ and the undirected edge set $\mathcal{E} = \{e_1, \cdots, e_m\}$, where n, m represent the number of node and edge, respectively. The graph structure $\mathcal{G}$ can be represented as an adjacent matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, where $\mathbf{A}_{ij} = 1$ if existing edge between node v_i and node v_j. $\mathcal{N}(i)$ denotes the neighbors of node v_i and $\tilde{\mathcal{N}}(i) = \mathcal{N}(i) \cup \{v_i\}$ denotes the self-inclusive neighbors. Suppose that each node is associated with a d-dimensional feature vector and a (binary) sensitive attribute, the feature for all nodes and sensitive attribute is denoted as $\mathbf{X}_{ori} = \mathbb{R}^{n\times d}$ and $\mathbf{s} \in \{-1, 1\}^n$ [3]. Define the sensitive attribute incident vector as $\Delta_{\mathbf{s}} = \frac{\mathbf{1}_{>0}(\mathbf{s})}{||\mathbf{1}_{>0}(\mathbf{s})||_1} - \frac{\mathbf{1}_{>0}(-\mathbf{s})}{||\mathbf{1}_{>0}(-\mathbf{s})||_1}$ to normalize each sensitive attribute group, where $\mathbf{1}_{>0}(\mathbf{s})$ is an element-wise indicator function.

GNNs as Graph Signal Denoising

A GNN model is usually composed of several stacking GNN layers. Given a graph $\mathcal{G}$ with N nodes, a GNN layer typically contains feature transformation $\mathbf{X}_{trans} = f_{trans}(\mathbf{X}_{ori})$ and aggregation $\mathbf{X}_{agg} = f_{agg}(\mathbf{X}_{trans})$, where $\mathbf{X}_{ori} \in \mathbb{R}^{n\times d_{in}}$, $\mathbf{X}_{trans}, \mathbf{X}_{agg} \in \mathbb{R}^{n\times d_{out}}$ represent the input and output features. The feature transformation operation transforms the node feature dimension, and *feature aggregation*, updates node features based on neighbors' features and graph topology. Recent works (**??**) have established the connections between many feature aggregation operations $AGG(\cdot)$ in representative GNNs and a graph signal denoising problem with Laplacian regularization, i.e., recovering a clean signal $\mathbf{F} \in \mathbb{R}^{n\times d_{out}}$ from $\mathbf{X}_{trans}$ with the smooth assumption over graph $\mathcal{G}$. Here, we introduce several popular GNN architectures, including GCN/SGC, GAT, and PPNP/APPNP, as examples to show the connection from the perspective of graph signal denoising.

GCN/SGC. Feature aggregation in Graph Convolutional Network (GCN) or Simplifying Graph Convolutional Network (SGC) is given by $\mathbf{X}_{agg} = \tilde{\mathbf{A}}\mathbf{X}_{trans}$, where $\tilde{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ is a normalized self-loop adjacency matrix $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, and $\tilde{\mathbf{D}}$ is degree matrix of $\tilde{\mathbf{A}}$. Recent works (**??**) provably demonstrate that such feature aggregation can be interpreted as one-step gradient descent to minimize $tr\big(\mathbf{F}^\top\big(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{F}\big)$ with initialization $\mathbf{F} = \mathbf{X}_{trans}$.

GAT. Feature aggregation in GAT applies the normalized attention coefficient to compute a linear combination of neighbor's features as $\mathbf{X}_{agg,i} = \sum_{j\in\mathcal{N}(i)}\alpha_{ij}\mathbf{X}_{trans,j}$, where $\alpha_{ij} = softmax_j(e_{ij})$, $e_{ij} = \text{LeakyReLU}(\mathbf{X}_{trans,i}^\top\mathbf{w}_i + \mathbf{X}_{trans,j}^\top\mathbf{w}_j)$, and $\mathbf{w}_i$ and $\mathbf{w}_j$ are learnable column vectors. Prior study (**?**) demonstrates that one-step gradient descent with adaptive stepsize $\frac{1}{\sum_{j\in\tilde{\mathcal{N}}(i)}(c_i+c_j)}$ for the following objective problem:

$$\min_{\mathbf{F}}\sum_{i\in\mathcal{V}}||\mathbf{F}_i - \mathbf{X}_{trans,i}||_F^2 + \frac{1}{2}\sum_{i\in\mathcal{V}}c_i\sum_{j\in\tilde{\mathcal{N}}(i)}||\mathbf{F}_i - \mathbf{F}_j||_F^2.$$

is actually an attention-based feature aggregation, which is equivalent to GAT if $c_i + c_j$ is equivalent to e_{ij}, where c_i is a node-dependent coefficient that measures the local smoothness.

PPNP / APPNP. Feature aggregation in PPNP and APPNP adopt the aggregation rules as $\mathbf{X}_{agg} = \alpha\Big(\mathbf{I} - (1 - \alpha)\tilde{\mathbf{A}}\Big)^{-1}\mathbf{X}_{trans}$ and $\mathbf{X}_{agg}^{k+1} = (1-\alpha)\tilde{\mathbf{A}}\mathbf{X}_{agg}^k + \alpha\mathbf{X}_{trans}$. It is shown that they are equivalent to the exact solution and one gradient descent step with stepsize $\frac{\alpha}{2}$ to minimize the following objective problem:

$$\min_{\mathbf{F}}||\mathbf{F} - \mathbf{X}_{trans}||_F^2 + (\frac{1}{\alpha} - 1)tr\Big(\mathbf{F}^\top(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{F}\Big).$$

Fair Message Passing

In this section, we propose a new fair message-passing scheme to aggregate useful information from neighbors while debiasing representation bias. In this way, fair prediction can be achieved from a model backbone perspective. Specifically, we formulate fair message passing as an optimization problem to pursue *smoothness* and *fair* node representation simultaneously [4]. Together with an effective and

[3] The sensitive attribute $\mathbf{s}$ is not included in node features matrix $\mathbf{X}_{ori}$.

[4] Fair message passing is an alternative operation to replace GNNs aggregations.

efficient optimization algorithm, we derive the closed-form fair message passing. Finally, the proposed FMP is shown to be integrated into fair GNNs at three stages, including transformation, aggregation, and debiasing step, as shown in Figure 1. These three stages adopted node feature, graph topology, and sensitive attributes respectively.

The Optimization Framework

Most of the existing works rely on hand-craft architecture (e.g., JKNet (?)) design for specific tasks, and thus lack of theoretical understanding how such architecture is designed. In our paper, starting from this unified optimization framework for GNNs, we design a new objective, including smoothness and fairness objective, and then derive the proposed FMP to explicitly chase the new objective via fair message passing.

In previous work (?), a general and universal framework is developed to understand aggregation operations in GNNs. Building on top of this framework, we formulate an optimization problem to achieve fair message passing operation (replace aggregation operations in GNNs). To achieve graph smoothness prior and fairness in the same process, a reasonable message passing should be a good solution for the following optimization problem:

$$\min_{\mathbf{F}} \underbrace{\frac{\lambda_s}{2} tr(\mathbf{F}^T\tilde{\mathbf{L}}\mathbf{F}) + \frac{1}{2}||\mathbf{F}-\mathbf{X}_{trans}||_F^2}_{h_s(\mathbf{F})} + \underbrace{\lambda_f||\mathbf{\Delta}_s SF(\mathbf{F})||_1}_{h_f\left(\mathbf{\Delta}_s SF(\mathbf{F})\right)}. \quad (1)$$

where $\tilde{\mathbf{L}}$ represents normalized Laplacian matrix, $h_s(\cdot)$ and $h_f(\cdot)$ denotes the smoothness and fairness objectives [5], respectively, and $\mathbf{X}_{trans} \in \mathbf{R}^{n\times d_{out}}$ is the transformed d_{out}-dimensional node features and $\mathbf{F} \in \mathbf{R}^{n\times d_{out}}$ is the aggregated node features of the same matrix size. The first two terms preserve the similarity of connected node representation and thus enforce graph smoothness. The last term enforces fair node representation so that the average predicted probability between groups of different sensitive attributes can remain constant. The regularization coefficients λ_s and λ_f adaptively control the trade-off between graph smoothness and fairness.

Smoothness Objective $h_s(\cdot)$. The adjacent matrix in existing graph message passing schemes is normalized for improving numerical stability and achieving superior performance. Similarly, the graph smoothness term requires normalized Laplacian matrix, i.e., $\tilde{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{A}}$, $\tilde{\mathbf{A}} = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}$, and $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$. From an edge-centric view, the smoothness objective enforces connected node representation to be similar since

$$tr(\mathbf{F}^T\tilde{\mathbf{L}}\mathbf{F}) = \sum_{(v_i,v_j)\in\mathcal{E}} ||\frac{\mathbf{F}_i}{\sqrt{d_i+1}} - \frac{\mathbf{F}_j}{\sqrt{d_j+1}}||_F^2, \quad (2)$$

where $d_i = \sum_k A_{ik}$ represents the degree of node v_i.

Fairness Objective $h_f(\cdot)$. The fairness objective measures the bias for node representation after aggregation. Recall sensitive attribute incident vector $\Delta_{\mathbf{s}}$ indicates the sensitive attribute group and group size via the sign and absolute value summation. Recall that the sensitive attribute incident vector as

$$\Delta_{\mathbf{s}} = \frac{\mathbf{1}_{>0}(\mathbf{s})}{||\mathbf{1}_{>0}(\mathbf{s})||_1} - \frac{\mathbf{1}_{>0}(-\mathbf{s})}{||\mathbf{1}_{>0}(-\mathbf{s})||_1}, \quad (3)$$

and $SF(\mathbf{F})$ represents the predicted probability for node classification task, where $SF(\mathbf{F})_{ij} = \hat{P}(y_i = j|\mathbf{X})$. Furthermore, we can show that our fairness objective is actually equivalent to demographic parity, i.e., $\Big(\Delta_s SF(\mathbf{F})\Big)_j = \hat{P}(y_i = j|\mathbf{s}_i = 1, \mathbf{X}) - \hat{P}(y_i = j|\mathbf{s}_i = -1, \mathbf{X})$. Please see proof in Appendix **??**. In other words, our fairness objective, l_1 norm of $\Delta_s SF(\mathbf{F})$ characterizes the predicted probability difference between two groups with different sensitive attributes. Therefore, our proposed optimization framework can pursue graph smoothness and fairness simultaneously.

Optimization Problem Solver

For smoothness objective, many existing popular message passing schemes can be derived based on gradient descent with appropriate step size choice (??). In this paper, we consider smoothness objective $h_s(\mathbf{F})$ and fairness objective $h_f(\Delta SF(\mathbf{F}))$ simultaneously for chasing fair and accurate prediction. However, directly solving the optimization problem (1) is much more challenging due to the nonsmoothness of the fairness objective, and the non-separability of smoothness objective $h_s(\mathbf{F})$ and fairness objective $h_f(\Delta SF(\mathbf{F}))$ due to incident vector Δ_s.

Bi-level Optimization Problem Formulation In the literature, many optimization algorithms are developed for optimization problems with l_1 norm, such as Alternating Direction Method of Multipliers (ADMM) and Newton type algorithms (??). However, these algorithms require non-trivial sub-problem solving for each iteration. Therefore, computation complexity is high and is infeasible to integrate deep learning models. Fortunately, Fenchel conjugate (a.k.a. convex conjugate) (?) can transform the original problem as an equivalent saddle point problem using a primal-dual algorithm (?). In this way, the computation complexity can be reduced and compatible with back-propagation training. Similarly, to solve optimization problem 1 in a more effective and efficient manner, Fenchel conjugate (?) is introduced to transform the original problem into a bi-level optimization problem. For the general convex function $h(\cdot)$, its conjugate function is defined as $h^*(\mathbf{U}) \overset{\triangle}{=} \sup_{\mathbf{X}}\langle\mathbf{U}, \mathbf{X}\rangle - h(\mathbf{X})$. Based on Fenchel conjugate, the fairness objective can be transformed as variational representation $h_f(\mathbf{p}) = \sup_{\mathbf{u}}\langle\mathbf{p}, \mathbf{u}\rangle - h_f^*(\mathbf{u})$, where $\mathbf{p} = \mathbf{\Delta}_s SF(\mathbf{F}) \in \mathbb{R}^{1\times d_{out}}$ is a predicted probability vector for classification. Furthermore, the original optimization problem is equivalent to

$$\min_{\mathbf{F}}\max_{\mathbf{u}} h_s(\mathbf{F}) + \langle\mathbf{p}, \mathbf{u}\rangle - h_f^*(\mathbf{u}) \quad (4)$$

[5] Such smoothness objective is the most common-used one in existing methods (???). The various other smoothness objectives could be considered to improve the performance of FMP and we leave it for future work.

where $\mathbf{u} \in \mathbb{R}^{1\times d_{out}}$ and $h_f^*(\cdot)$ is the conjugate function of fairness objective $h_f(\cdot)$.

Problem Solution Motivated by Proximal Alternating Predictor-Corrector (PAPC) (**??**), the min-max optimization problem (4) can be solved by the following fixed-point equations with per iteration low computation complexity and convergence guarantee

$$\begin{cases}\mathbf{F} = \mathbf{F} - \nabla h_s(\mathbf{F}) - \frac{\partial\langle \mathbf{p},\mathbf{u}\rangle}{\partial \mathbf{F}},\\ \mathbf{u} = \text{prox}_{h_f^*}\big(\mathbf{u} + \mathbf{\Delta_s} SF(\mathbf{F})\big).\end{cases} \tag{5}$$

where $\text{prox}_{h_f^*}(\mathbf{u}) = \arg\min_{\mathbf{y}} ||\mathbf{y} - \mathbf{u}||_F^2 + h_f^*(\mathbf{y})$. Fortunately, the proximal operators can be obtained with a close form, which makes deep learning model integration feasible. Specifically we provide the close form of the proximal operators in the following proposition:

Proposition 0.1 (Proximal Operators)**.** *The proximal operators $prox_{\beta h_f^*}(\mathbf{u})$ satisfies*

$$prox_{\beta h_f^*}(\mathbf{u})_j = sign(\mathbf{u})_j \min\big(|\mathbf{u}_j|, \lambda_f\big), \tag{6}$$

where $sign(\cdot)$ and λ_f are element-wise sign function and hyperparameter for fairness objective. In other words, such a proximal operator is an element-wise projection into l_∞ ball with radius λ_f.

Similar to "predictor-corrector" algorithm (**?**), we adopt an iterative algorithm to find the saddle point for the min-max optimization problem. Specifically, starting from $(\mathbf{F}^k, \mathbf{u}^k)$, we adopt a gradient descent step on the primal variable $\mathbf{F}$ to arrive $(\bar{\mathbf{F}}^{k+1}, \mathbf{u}^k)$ and then followed by a proximal ascent step in the dual variable $\mathbf{u}$. Finally, a gradient descent step on a primal variable in point $(\bar{\mathbf{F}}^{k+1}, \mathbf{u}^k)$ to arrive at $(\mathbf{F}^{k+1}, \mathbf{u}^k)$. In short, the iteration can be summarized as

$$\begin{cases}\bar{\mathbf{F}}^{k+1} = \mathbf{F}^k - \gamma\nabla h_s(\mathbf{F}^k) - \gamma\frac{\partial\langle \mathbf{p},\mathbf{u}^k\rangle}{\partial \mathbf{F}}\Big|_{\mathbf{F}^k},\\ \mathbf{u}^{k+1} = \text{prox}_{\beta h_f^*}\big(\mathbf{u}^k + \beta\mathbf{\Delta_s} SF(\bar{\mathbf{F}}^{k+1})\big),\\ \bar{\mathbf{F}}^{k+1} = \mathbf{F}^k - \gamma\nabla h_s(\mathbf{F}^k) - \gamma\frac{\partial\langle \mathbf{p},\mathbf{u}^{k+1}\rangle}{\partial \mathbf{F}}\Big|_{\mathbf{F}^k}.\end{cases} \tag{7}$$

where γ and β are the step size for primal and dual variables. Note that the close-form for $\frac{\partial\langle \mathbf{p},\mathbf{u}\rangle}{\partial \mathbf{F}} \in \mathbb{R}^{n\times d_{out}}$ and $\text{prox}_{\beta h_f^*}(\cdot)$ are still not clear, we will provide the solution one by one.

FMP Scheme. Similar to works (**??**), choosing $\gamma = \frac{1}{1+\lambda_s}$ and $\beta = \frac{1}{2\gamma}$, we have

$$\begin{aligned}\mathbf{F}^k - \gamma\nabla h_s(\mathbf{F}^k) &= \Big((1-\gamma)\mathbf{I} - \gamma\lambda_s\tilde{\mathbf{L}}\Big)\mathbf{F}^k + \gamma\mathbf{X}_{trans}\\ &= \gamma\mathbf{X}_{trans} + (1-\gamma)\tilde{\mathbf{A}}\mathbf{F}^k,\end{aligned} \tag{8}$$

Therefore, we can summarize the proposed FMP as two phases, including propagation with skip connection (Step ❶) and bias mitigation (Steps ❷-❺). For bias mitigation, Step ❷ updates the aggregated node features for fairness objective; Steps ❸ and ❹ aim to learn and "reshape" perturbation vector in probability space, respectively. Step ❺ explicitly mitigates the bias of node features based on gradient descent on the primal variable. The mathematical formulation is given as follows:

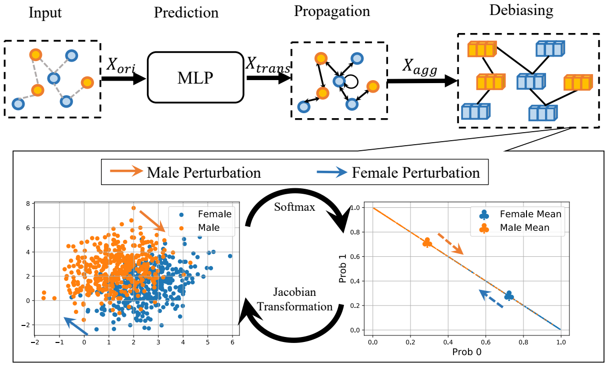

Figure 1: The model pipeline consists of three steps: MLP (feature transformation), propagation with skip connection, and debiasing via low-rank perturbation in probability space.

$$\begin{cases}\mathbf{X}_{agg}^{k+1} = \gamma\mathbf{X}_{trans} + (1-\gamma)\tilde{\mathbf{A}}\mathbf{F}^k, & \text{Step ❶}\\ \bar{\mathbf{F}}^{k+1} = \mathbf{X}_{agg}^{k+1} - \gamma\frac{\partial\langle \mathbf{p},\mathbf{u}^k\rangle}{\partial \mathbf{F}}\Big|_{\mathbf{F}^k}, & \text{Step ❷}\\ \bar{\mathbf{u}}^{k+1} = \mathbf{u}^k + \beta\mathbf{\Delta_s} SF(\bar{\mathbf{F}}^{k+1}), & \text{Step ❸}\\ \mathbf{u}^{k+1} = \min\Big(|\bar{\mathbf{u}}^{k+1}|, \lambda_f\Big)\cdot sign(\bar{\mathbf{u}}^{k+1}), & \text{Step ❹}\\ \mathbf{F}^{k+1} = \mathbf{X}_{agg}^{k+1} - \gamma\frac{\partial\langle \mathbf{p},\mathbf{u}^{k+1}\rangle}{\partial \mathbf{F}}\Big|_{\mathbf{F}^k}. & \text{Step ❺}\end{cases}$$

where $\mathbf{X}_{agg}^{k+1}$ represents the node features with normal aggregation and skip connection with the transformed input $\mathbf{X}_{trans}$.

Gradient Computation Acceleration The softmax property is also adopted to accelerate the gradient computation. Note that $\mathbf{p} = \mathbf{\Delta}_s SF(\mathbf{F})$ and $SF(\cdot)$ represents softmax over column dimension, directly computing the gradient $\frac{\partial\langle \mathbf{p},\mathbf{u}\rangle}{\partial \mathbf{F}}$ based on chain rule involves the three-dimensional tensor $\frac{\partial \mathbf{p}}{\partial \mathbf{F}}$ with gigantic computation complexity. Instead, we simplify the gradient computation based on the property of softmax function in the following theorem.

Theorem 0.2 (Gradient Computation)**.** *The gradient over primal variable $\frac{\partial\langle \mathbf{p},\mathbf{u}\rangle}{\partial \mathbf{F}}$ satisfies*

$$\frac{\partial\langle \mathbf{p},\mathbf{u}\rangle}{\partial \mathbf{F}} = \mathbf{U}_s \odot SF(\mathbf{F}) - Sum_1(\mathbf{U}_s \odot SF(\mathbf{F}))SF(\mathbf{F}). \tag{9}$$

where $\mathbf{U}_s \triangleq \Delta_s^\top \mathbf{u}$, $\odot$ represents the element-wise product and $Sum_1(\cdot)$ represents the summation over column dimension with preserved matrix shape.

Discussion on FMP

In this section, we provide the interpretation and analyze the *efficiency*, and *white-box usage for sensitive attribute* of the proposed FMP scheme. Furthermore, we also discuss how FMP identifies the influence of sensitive attributes from model forward propagation.

FMP Interpretation Note that the gradient of fairness objective over node features $\mathbf{F}$ satisfies $\frac{\partial\langle \mathbf{p},\mathbf{u}\rangle}{\partial \mathbf{F}} = \frac{\partial\langle \mathbf{p},\mathbf{u}\rangle}{\partial SF(\mathbf{F})}\frac{\partial SF(\mathbf{F})}{\partial \mathbf{F}}$ and $\frac{\partial\langle \mathbf{p},\mathbf{u}\rangle}{\partial SF(\mathbf{F})} = \Delta_s^\top \mathbf{u}$, such gradient calculation can be interpreted as three steps: Softmax transformation, perturbation in probability space, and debiasing in representation space. Specifically, we first map the node representation into probability space via softmax transformation. Subsequently, we calculate the gradient of fairness objective in probability space. It is seen that the perturbation $\Delta_s^\top \mathbf{u}$ actually poses *low-rank* debiasing in probability space, where the nodes with different sensitive attributes embrace opposite perturbations. In other words, *the dual variable* $\mathbf{u}$ *represents the perturbation direction in probability space.* Finally, the perturbation in probability space will be transformed into representation space via Jacobian transformation $\frac{\partial SF(\mathbf{F})}{\partial \mathbf{F}}$.

Efficiency. FMP is an efficient message-passing scheme. The computation complexity for the aggregation (sparse matrix multiplications) is $O(md_{out})$, where m is the number of edges in the graph. For FMP, the extra computation mainly focuses on the perturbation calculation, as shown in Theorem 0.2, with the computation complexity $O(nd_{out})$. The extra computation complexity is negligible in that the number of nodes n is far less than the number of edges m in the real-world graph. Additionally, if directly adopting backward propagation to calculate the gradient, we have to calculate the three-dimensional tensor $\frac{\partial \mathbf{p}}{\partial \mathbf{F}}$ with computation complexity $O(n^2 d_{out})$. In other words, thanks to the softmax property, we achieve an efficient fair message-passing scheme.

White-box Usage for Sensitive Attribute. The proposed FMP explicitly achieves graph smoothness and fairness objectives via alternative gradient descent. In other words, the usage of sensitive attributes in propagation to mitigate bias is in a white-box manner. Note that such white-box usage of sensitive attributes is a promising property to understand how sensitive attribute usage forces fairness, which is not achieved by previous fairness methods in GNNs. For example, fair training loss utilizes sensitive attributes to regularize the behavior of model prediction and obtain fairer model parameters via rectifying gradients w.r.t. model parameters. In other words, the sensitive attribute information is implicitly encoded in the well-trained model parameters, which makes it hard to understand how sensitive attribute usage helps fair prediction. Pre-processing fairness methods adopt sensitive attributes to revise data (e.g., node masking and topology rewiring) either in a learnable way or via pre-defined several operations (e.g., node masking and edge deletions). Similarly, the sensitive attribute information is implicitly encoded in the processed data. The understanding of fairness prediction achievement is infeasible. Our FMP can provide a white-box usage for sensitive attributes since we can directly identify that the usage of sensitive attributes is to force the demographic group node representation centers together during forward propagation.

Experiments

In this section, we conduct experiments to validate the effectiveness and efficiency of the proposed FMP. We firstly validate that graph data with large sensitive homophily enhances bias in GNNs via synthetic experiments. Moreover, for experiments on real-world datasets, we introduce the experimental settings and then evaluate our proposed FMP compared with several baselines in terms of prediction performance and fairness metrics.

Experimental Settings

Datasets. We conduct experiments on real-world datasets Pokec-z, Pokec-n [6], and NBA (**?**). Pokec-z and Pokec-n are sampled, based on province information, from a larger Facebook-like social network Pokec (**?**) in Slovakia, where region information is treated as the sensitive attribute and the predicted label is the working field of the users. NBA dataset is extended from a Kaggle dataset [7] consisting of around 400 NBA basketball players. The information of players includes age, nationality, and salary in the 2016-2017 season. The players' link relationships are from Twitter with the official crawling API. The binary nationality (U.S. and overseas player) is adopted as the sensitive attribute and the prediction label is whether the salary is higher than the median.

Evaluation Metrics. We adopt accuracy to evaluate the performance of node classification tasks. As for fairness metrics, we adopt two quantitative group fairness metrics to measure the prediction bias. According to works (**??**), we adopt *demographic parity* $\Delta_{DP} = |\mathbb{P}(\hat{y} = 1|s = -1) - \mathbb{P}(\hat{y} = 1|s = 1)|$ and *equal opportunity* $\Delta_{EO} = |\mathbb{P}(\hat{y} = 1|s = -1, y = 1) - \mathbb{P}(\hat{y} = 1|s = 1, y = 1)|$, where y and $\hat{y}$ represent the ground-truth label and predicted label, respectively.

Baselines. We compare our proposed FMP with representative GNNs, such as GCN (**?**), GAT (**?**), SGC (**?**), and APPNP (**?**), JKNet (**?**), and MLP. We also compared with method "ML1" directly using the gradient of Eq. (1) during model forward propagation. For all models, we train 2 layers of neural networks with 64 hidden units for 300 epochs. Additionally, We also compare adversarial debiasing and adding demographic regularization methods to show the effectiveness of the proposed method [8].

Implementation Details. We run the experiments 5 times and report the average performance for each method. We adopt Adam optimizer with 0.001 learning rate and 10^{-5} weight decay for all models. For adversarial debiasing, we adopt the train classifier and adversary with 70 and 30 epochs, respectively. The hyperparameter for adversary loss is tuned in $\{0.0, 1.0, 2.0, 5.0, 8.0, 10.0, 20.0, 30.0\}$. For adding regularization, we adopt the hyperparameter set $\{0.0, 1.0, 2.0, 5.0, 8.0, 10.0, 20.0, 50.0, 80.0, 100.0\}$.

[6] Pokec-z and Pockec-n datasets are available at https://github.com/EnyanDai/FairGNN/tree/main.

[7] https://www.kaggle.com/noahgift/social-power-nba

[8] Please see the comparison with Fair Mixup (**?**) in Appendix **??**

Table 1: Comparative Results with Baselines on Node Classification.

Models	Pokec-z			Pokec-n			NBA		
	Acc (%) ↑	Δ_{DP} (%) ↓	Δ_{EO} (%) ↓	Acc (%) ↑	Δ_{DP} (%) ↓	Δ_{EO} (%) ↓	Acc (%) ↑	Δ_{DP} (%) ↓	Δ_{EO} (%) ↓
MLP	70.48 ± 0.77	1.61 ± 1.29	2.22 ± 1.01	72.48 ± 0.26	1.53 ± 0.89	3.39 ± 2.37	65.56 ± 1.62	22.37 ± 1.87	18.00 ± 3.52
GAT	69.76 ± 1.30	2.39 ± 0.62	2.91 ± 0.97	71.00 ± 0.48	3.71 ± 2.15	7.50 ± 2.88	57.78 ± 10.65	20.12 ± 16.18	13.00 ± 13.37
GCN	**71.78** ± 0.37	3.25 ± 2.35	2.36 ± 2.09	**73.09** ± 0.28	3.48 ± 0.47	5.16 ± 1.38	61.90 ± 1.00	23.70 ± 2.74	17.50 ± 2.63
SGC	71.24 ± 0.46	4.81 ± 0.30	4.79 ± 2.27	71.46 ± 0.41	2.22 ± 0.29	3.85 ± 1.63	63.17 ± 0.63	22.56 ± 3.94	14.33± 2.16
APPNP	66.91 ± 1.46	3.90 ± 0.69	5.71 ± 1.29	69.80 ± 0.89	1.98 ± 1.30	4.01 ± 2.36	63.80 ± 1.19	26.51 ± 3.33	20.00 ± 4.56
JKNet	66.89 ± 3.79	1.28 ±0.96	1.79 ± 0.82	63.59 ± 6.36	1.91 ± 2.14	**0.70** ± 0.92	67.94 ± 2.73	27.80 ± 8.41	20.33 ± 7.52
ML1	70.42 ± 0.40	2.35 ± 0.83	2.00 ± 0.50	72.36 ± 0.26	1.47 ± 1.12	3.03 ± 1.77	72.70 ± 1.19	26.46 ± 4.93	25.50 ± 8.38
FMP	70.50 ± 0.50	**0.81** ± 0.40	**1.73** ± 1.03	72.16 ± 0.33	**0.66** ± 0.40	1.47 ± 0.87	**73.33** ± 1.85	**18.92** ± 2.28	**13.33** ± 5.89

Experimental Results

Comparison with Existing GNNs. The accuracy, demographic parity, and equal opportunity metrics of proposed FMP for Pokec-z, Pokec-n, NBA datasets are shown in Table 1 compared with MLP, GAT, GCN, SGC, and APPNP. The detailed statistical information for these three datasets is shown in Table **??**. From these results, we can obtain the following observations:

- Many existing GNNs underperform MLP model on all three datasets in terms of fairness metric. For instance, the demographic parity of MLP is lower than GAT, GCN, SGC and APPNP by 32.64%, 50.46%, 66.53% and 58.72% on Pokec-z dataset. The higher prediction bias comes from the aggregation within the same sensitive attribute nodes and topology bias in graph data.
- Our proposed FMP consistently achieves the lowest prediction bias in terms of demographic parity and equal opportunity on all datasets. Specifically, FMP reduces demographic parity by 49.69%, 56.86%, and 5.97% compared with the lowest bias among all baselines in Pokec-z, Pokec-n, and NBA datasets. Meanwhile, our proposed FMP achieves the best accuracy in NBA dataset, and comparable accuracy in Pokec-z and Pokec-n datasets. In a nutshell, the proposed FMP can effectively mitigate prediction bias while preserving the prediction performance.

Comparison with Adversarial Debiasing and Regularization. To validate the effectiveness of the proposed FMP, we also show the prediction performance and fairness metric trade-off compared with fairness-boosting methods, including adversarial debiasing (**?**) and adding regularization (**?**). Similar to (**?**), the output of GNNs is the input of the adversary and the goal of the adversary is to predict the node sensitive attribute. We also adopt several backbones for these two methods, including MLP, GCN, GAT, and SGC. We randomly split 50%/25%/25% for training, validation, and test dataset. Figure 2 shows the Pareto optimality curve for all methods, where the right-bottom corner point represents the ideal performance (highest accuracy and lowest prediction bias). From the results, we list the following observations as follows:

- Our proposed FMP can achieve better DP-Acc trade-off compared with adversarial debiasing and adding regularization for many GNNs and MLP. Such observation validates the effectiveness of the key idea in FMP: aggregation first and then debiasing. Additionally, FMP can reduce demographic parity with negligible performance cost due to transparent and efficient debiasing.
- Message passing in GNNs does matter. For adding regularization or adversarial debiasing, different GNNs embrace huge distinctions, which implies that an appropriate message passing manner potentially leads to better trade-off performance. Additionally, many GNNs underperforms MLP in low-label homophily coefficient dataset, such as NBA. The rationale is that aggregation may not always bring benefit in terms of accuracy when the neighbors have low probability with the same label.

Related Works

Graph Neural Networks. GNNs generalizing neural networks for graph data have already shown great success in various real-world applications. There are two streams in GNNs model design, i.e., spectral-based and spatial-based. Spectral-based GNNs provide graph convolution definition based on graph theory, which is utilized in GNN layers together with feature transformation (**???**). Graph convolutional networks (GCN) (**?**) simplify spectral-based GNN model into spatial aggregation scheme. Since then, many spatial-based GNNs variant is developed to update node representation via aggregating its neighbors' information, including graph attention network (GAT) (**?**), GraphSAGE (**?**), SGC (**?**), APPNP (**?**), et al (**??**). Graph signal denoising is another perspective to understand GNNs. Recently, there are several works show that GCN is equivalent to the first-order approximation for graph denoising with Laplacian regularization (**??**). The unified optimization framework is provided to unify many existing message passing schemes (**??**).

Fairness-aware Learning on Graphs. Many works have been developed to achieve fairness in machine learning community (**?????????**). A pilot study on fair node representation learning is developed based on random walk (**?**). Addi-

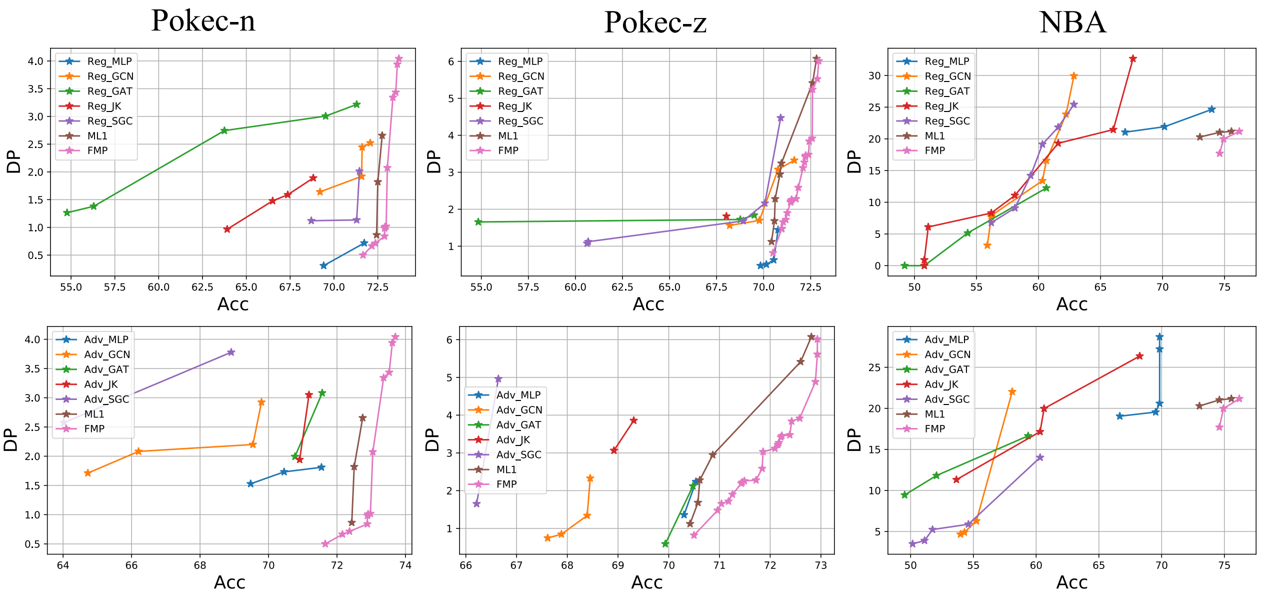

Figure 2: DP and Acc trade-off performance on three real-world datasets compared with adding regularization (Top) and adversarial debiasing (Bottom). The trade-off curve close to the right bottom corner means better trade-off performance. The units for x- and y-axis are percentages (%).

tionally, adversarial debiasing is adopted to learn fair prediction or node representation so that the well-trained adversary can not predict the sensitive attribute based on node representation or prediction (**???**). A Bayesian approach is developed to learn fair node representation via encoding sensitive information in the prior distribution in (**?**). Work (**?**) develops a PAC-Bayesian analysis to connect subgroup generalization with accuracy parity. (**??**) aims to mitigate prediction bias for link prediction. Fairness-aware graph contrastive learning is proposed in (**???**). Graph data preprocessing, such as node feature masking and graph topology rewire, are also developed in (**?????**) for node classification and link prediction tasks. However, the aforementioned works ignore the requirement of transparency in fairness. In this work, we develop an efficient and transparent fair message passing scheme explicitly rendering sensitive attribute usage.

Conclusion

In this work, we improve fairness in graphs from the model architecture perspective. We design a fair message-passing scheme to achieve fair prediction for node classification using vanilla training loss without data pre-processing. Specifically, motivated by the unified optimization framework for GNNs, FMP is designed as aggregation first and then bias mitigation to explicitly chase smoothness and fairness objectives. We also provide a comprehensive discussion of FMP from model architecture interpretation, efficiency, and the white-box usage of sensitive attributes aspects. Experimental results on real-world datasets demonstrate the effectiveness of FMP compared with several baselines in node classification tasks.

Acknowledgements

We would like to sincerely thank everyone who has provided their generous feedback for this work. Thank the anonymous reviewers for their thorough comments and suggestions. The work is in part supported by NSF grants IIS-1900990, IIS-1939716, and IIS-2239257. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

Enim exercitationem cupiditate rerum nemo quasi odio minus facere esse earum ullam, unde facere dolores, tempore impedit assumenda explicabo ipsum libero, optio repudiandae rerum id sapiente non veritatis alias magni.Nam ut dolorem aperiam fugit doloribus esse, molestiae ipsam voluptate atque, nihil sit eveniet quidem deleniti aspernatur, quas rem aperiam sunt dolore ducimus labore eaque quod veritatis at necessitatibus, distinctio quibusdam ea praesentium aliquam expedita numquam illo quisquam illum fuga.Beatae doloribus laboriosam unde dignissimos tempora quod obcaecati neque necessitatibus, consectetur error repudiandae tempore, voluptate veniam vitae ut rerum vel eum voluptatum quos accusamus, ea quia enim fugit nobis dignissimos accusantium placeat asperiores?Ipsa voluptates ipsum necessitatibus quibusdam, ratione maxime ut praesentium mollitia atque perspiciatis illo et sapiente magni, unde quae quo ipsum nesciunt reiciendis quam modi veritatis dolor, facere aliquam ratione animi quos quod possimus accusantium commodi pariatur quo dolorem.Minima voluptates ullam ipsum unde minus doloribus odio modi similique cumque, dolorum accusamus commodi eos nesciunt impedit quibusdam sunt, delectus omnis hic natus nemo non et veniam atque animi?Adipisci consequuntur voluptas quasi autem harum odio ut totam, doloribus ab odit minima ad officiis facere expedita qui reiciendis impedit dolorem, doloribus officia soluta eius ea tenetur, itaque vitae ratione nulla quaerat consequatur?