Filtering as Rewiring for Bias Mitigation on Graphs

O. Deniz Kose
Dept. of EECS
University of California Irvine
Irvine, USA
okose@uci.edu

Gonzalo Mateos
Dept. of ECE
University of Rochester
Rochester, USA
gmateosb@ece.rochester.edu

Yanning Shen
Dept. of EECS
University of California Irvine
Irvine, USA
yannings@uci.edu

***Abstract*—Machine learning over graphs (MLoG) has attracted growing attention due to its effectiveness in processing relational data from complex systems such as social networks, financial markets, and the brain. However, MLoG algorithms that use the graph topology for information aggregation have been shown to amplify the already existing bias towards certain under-represented groups, often leading to discriminatory results in downstream tasks. In this context, here we consider the problem of topology-induced algorithmic bias mitigation by cross-pollinating tools from MLoG and graph signal processing. Specifically, we argue that application of a tunable debiasing graph filter can be reinterpreted as a graph rewiring process, thus offering an explicit handle to manipulate the utility versus topological bias tradeoff. Building on this insight, we formulate a fairness-aware network topology inference problem to obtain a rewired graph minimizing a correlation-based, unsupervised bias metric. Node classification experiments on several real-world datasets demonstrate that the proposed approach typically outperforms state-of-the-art baselines in terms of fairness metrics, and without a degradation in classification accuracy.**

***Index Terms*—Fairness, graph neural network, node classification, bias mitigation, graph rewiring.**

I. Introduction

Graph theory offers a natural framework to study complex systems and the pairwise relations between their constituent components, such as protein interactions in biological networks or monetary transactions in financial markets [1]. Accordingly, effective and scalable learning from increasingly ubiquitous *graph data* can impact a gamut of applications in engineering, commerce, and the biobehavioral sciences [2]. The recent successes of machine learning over graphs (MLoG) have been well documented [3], [4], but it remains an active area of research with unique challenges due to e.g., data high-dimensionality, statistical dependencies, and the intertwining between graph structure and nodal attributes or features.

Graphs are mathematical constructs consisting of vertices (or nodes) as well as the edges that connect them. This way graph edges encode relational patterns, while the graph signal processing (GSP) perspective is to view nodal attributes as signals defined on the vertices. Going back to our financial market example, the cash reserves of each trading institution can be represented as a graph signal. GSP broadens the conventional signal processing toolbox [5], [6], introducing e.g., frequency analysis, filtering, and sampling of signals on graphs [7]–[13]. Recent works have shown that workhorse MLoG models such as graph neural networks (GNNs) can be understood and improved by leveraging GSP-based insights [14]–[16]. Here, we fruitfully exploit GSP advances to mitigate bias in MLoG.

Algorithmic bias and fairness in ML. Trustworthy deployment of ML pipelines in real-world decision systems crucially requires the consideration of fairness [17, Ch.10]. This work focuses on *group fairness*, which measures the performance gap between sensitive groups (e.g., genders, ethnicities) [18]. For instance, the performance of a fair financial fraud detection algorithm should not depend on the gender, race, or socio-economic status of the account holders. In this context, algorithmic bias measures the (generally unwanted) stereotypical correlations encoded and propagated by ML algorithms with respect to these sensitive attributes. ML models are known to propagate the pre-existing bias within the training data, leading to discriminatory performance in downstream tasks [19]. This challenge is compounded in MLoG, since information aggregation over the biased (due to homophily) graph topology has been demonstrated to *amplify* biases in the data [20]. Motivated by this finding, several strategies have been proposed to mitigate algorithmic bias in MLoG, such as adversarial regularization [20], [21], fairness constraints [22], [23], and fairness-aware graph data augmentation [24]–[26].

Proposed approach and contributions. We first introduce an unsupervised bias metric that captures the stereotypical correlations within the graph topology (Section III-A). Adopting a fundamentally different perspective on our prior work [27], [28], in Section III-B we show that interleaving a tunable debiasing graph filter between GNN layers can be reintepreted as a graph rewiring mechanism. This suggests a novel approach to bias mitigation, where filter design becomes a search over effective graph topologies driven by fairness-aware optimality criteria (Section III-C). Under the hood, our rewiring problem implicitly retains the eigenvectors of the original graph (hence, the signal representation basis) and optimizes the eigenvalues to minimize the bias measure. Unlike the fairness-aware filter designs in [27], [28], the convex formulation here is devoid of spectral graph decompositions and it allows for explicit consideration of utility (i.e., accuracy for node classification); thus offering an explicit handle to flexibly manipulate the fairness versus utility trade-off. Overall, our contributions are:

Work in this paper was supported in part by the Google Research Scholar Award, and the NSF awards CCF-1750428 and CCF-1934962.

i) We formulate a fair network topology inference problem, which modifies a given graph to minimize a correlation-based bias measure. This rewiring approach is intrinsically equivalent to a debiasing graph filter design in the vertex domain. Overall, our new idea is to explore and exploit the interplay between *signal* and *edge* filtering for topology-induced bias mitigation; **ii)** Our method is algorithm- and task-agnostic. It can be used for different learning models and tasks, which makes it more versatile than most existing fairness-aware MLoG strategies; **iii)** For a given graph, the rewiring algorithm needs to be run once, independent of GNN training for different tasks; and **iii)** Node-classification experiments with several real-world network datasets showcase the effectiveness of the proposed method in mitigating bias, while maintaining similar utility relative to state-of-the-art algorithms; see Section IV.

II. Preliminaries and Problem Statement

This study develops a graph rewiring approach to mitigate topological bias in MLoG algorithms. Next, we describe the setup, provide needed GSP background, and state the problem.

We are given an undirected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{v_1, \ldots, v_N\}$ stands for the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. The topology of $\mathcal{G}$ is encoded in the adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, where $A_{ij} = 1$ if and only if $(v_i, v_j) \in \mathcal{E}$. Extensions to weighted graphs are straightforward. Defining $\mathbf{D} \in \mathbb{R}^{N \times N}$ as the diagonal degree matrix where D_{ii} is the degree of v_i, then $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ denotes the normalized graph Laplacian matrix. Nodal features are given by $\mathbf{X} \in \mathbb{R}^{N \times F}$, whose columns can be viewed as graph signals. Sensitive attributes are nodal features which should not affect the output of a fair learning algorithm. We henceforth consider a single binary sensitive attribute, which we collect in $\mathbf{s} \in \{-1,1\}^N$. Accordingly, the feature vector and the sensitive attribute of node v_i are denoted by $\mathbf{x}_i \in \mathbb{R}^F$ and $s_i \in \{-1,1\}$, respectively. In (semi-supervised) node classification tasks, vertices may have (e.g., binary) labels y_i.

Fourier analysis on graphs. Let the eigendecomposition of the Laplacian be $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ are the non-negative eigenvalues and the columns of $\mathbf{V}$ are the corresponding Laplacian eigenvectors. Then, the graph Fourier transform (GFT) of a graph signal $\mathbf{z} \in \mathbb{R}^N$ is given by $\tilde{\mathbf{z}} = \mathbf{V}^\top\mathbf{z}$, i.e., the projection of $\mathbf{z}$ onto the space spanned by the orthogonal eigenvectors of $\mathbf{L}$ [7], [29], [30]. In this transform, the eigenvalues of the positive semi-definite (PSD) Laplacian correspond to graph frequencies and they quantify the variation of the eigenvectors with respect to $\mathcal{G}$; see e.g., [5] for details.

Graph filters. Just like their classical signal processing counterparts, graph filters are used to manipulate graph signals for e.g., smoothing, denoising [31], [32], and classification [33], [34]. Given an input signal $\mathbf{z}_{\text{in}} \in \mathbb{R}^N$ and a graph filter with frequency response $\tilde{\mathbf{h}} := [\tilde{h}_1, \ldots, \tilde{h}_N]^\top$, the filtering operation is given by (see e.g., [5], [7], [14] and the tutorial [12]):

$$\mathbf{z}_{\text{out}} = \mathbf{V}\text{diag}(\tilde{h}_1, \ldots, \tilde{h}_N)\mathbf{V}^\top\mathbf{z}_{\text{in}} \quad \Leftrightarrow \quad \tilde{\mathbf{z}}_{\text{out}} = \tilde{\mathbf{h}} \circ \tilde{\mathbf{z}}_{\text{in}}. \quad (1)$$

In analogy to the convolution theorem, frequency-domain filtering boils down to point-wise multiplication ($\circ$) of $\tilde{\mathbf{z}}_{\text{in}} = \mathbf{V}^\top\mathbf{z}_{\text{in}}$ with the frequency response $\tilde{\mathbf{h}}$ of the graph filter.

Network topology inference. Estimating latent graph structure from nodal observations has a long history [35], [36]. Noteworthy renditions include Gaussian graphical model selection [37], structural equation models [38], signal smoothness minimization [39], [40], or network deconvolution from spectral templates [41], [42]. However, none of these prior works take algorithmic bias into consideration.

A. Problem statement

Given $\mathcal{G}$ and $\mathbf{s}$, the goal is to minimize a structural bias metric (see Section III-A) by shaping the spectrum of $\mathcal{G}$. As elaborated next, our graph rewiring idea is inspired by a debiasing graph filter design we revisit and reinterpret.

III. Topological Bias Mitigation

A. Topological bias

While algorithmic bias leads to discriminatory ML-based predictions, it can be even more problematic for models that exploit graph structure [20]. This topological bias predicament is rooted in the homophily principle of network formation. Specifically, edges are more likely to link vertices that have similar characterics, which leads to denser connectivity within the group of nodes with the same sensitive attribute [43]. For example, a social network user is more likely to connect with others of shared ethnicity. Thus, by aggregating information from neighbors (mostly with the same sensitive attribute as the anchor node), the use of graph structure in MLoG leads to node representations that are highly correlated with the sensitive attributes. Initial traces of bias in homphilous graphs are prone to amplification as a result of repeated aggregation steps used for represention learning in e.g., GNNs – even when the sensitive attributes are not directly used in training [44].

Bias measure. We adopt $\rho := \|\mathbf{s}^\top\hat{\mathbf{A}}\|_1 = \sum_j |\mathbf{s}^\top\hat{\mathbf{A}}_{:,j}| = \sum_j |\sum_i s_i\hat{A}_{ij}|$ as a topological bias measure, where $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix of $\mathcal{G}$. This bias measure captures the linear correlation between the sensitive attributes $\mathbf{s}$ and $\hat{\mathbf{A}}$, and it is inspired by the above observation that the connectivity patterns among nodes can be highly correlated with $\mathbf{s}$. Indeed, this correlation is proportional to $|\mathbf{s}^\top\hat{\mathbf{A}}_{:,j}|$, since the jth column of $\hat{\mathbf{A}}$ specifies the nodes over which the information will be aggregated for node v_j (together with the corresponding weights). For further insights, consider an homophilous setting where $\hat{A}_{ij}$ typically takes on larger values when node j has the same sensitive attribute as node i, e.g., $s_j = s_i = 1$, and smaller otherwise. This leads to highly-correlated $\mathbf{s}$ and $\hat{\mathbf{A}}_{:,j}$ reflected by a larger $|\sum_i s_i\hat{A}_{ij}|$, and hence a larger ρ relative to the disassortative case where weights A_{ij} bear no relation with s_i and s_j.

All in all, to alleviate algorithmic bias in MLoG a graph shift [45] (or aggregation) operator that is less correlated with $\mathbf{s}$ would be desirable. This conclusion notwithstanding, $\hat{\mathbf{A}}$ carries useful information to aid learning so there is a fairness versus utility trade-off here that we will explore as well.

B. Debiasing graph filter and effective network operator

At the heart of most MLoG approaches are node representations obtained via local aggregation of information (often composed with point-wise nonlinearities) [2]. Disregarding learnable weights that are not essential to our subsequent argument, in its simplest form the aggregation process over $\mathcal{G}$ is given by $\mathbf{R} = \hat{\mathbf{A}}\mathbf{X}$, where $\mathbf{R}$ denotes the resulting node representations, and $\mathbf{X}$ stands for the input features or node embeddings from the previous layer; see e.g., [3], [4].

The adoption of a (non-trained) debiasing graph filter with carefully designed frequency response $\tilde{\mathbf{h}}$ was advocated in [28]. The idea is to filter $\mathbf{X}$ prior to aggregation, namely

$$\begin{aligned}\mathbf{R}_f &= \hat{\mathbf{A}}\mathbf{X}_f \\ &= \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\mathbf{V}^\top \mathbf{X}_f \\ &= \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\mathbf{V}^\top \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top \mathbf{X} \\ &= \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top \mathbf{X} \\ &= \mathbf{A}_f \mathbf{X},\end{aligned} \tag{2}$$

where $\mathbf{A}_f := \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top$ is an *effective* network operator. Note that $\mathbf{A}_f$ and $\hat{\mathbf{A}}$ share the same eigenbasis $\mathbf{V}$, and filtering $\mathbf{X}$ with $\tilde{\mathbf{h}}$ offers N degrees of freedom to shape the eigenvalue spectrum (hence the edge connectivity pattern) of $\mathbf{A}_f$. This simple, but key observation is the crux to the graph rewiring approach presented next.

C. Fairness-aware graph rewiring

Capitalizing on the previously revealed interplay between signal and edge filtering, we formulate the following fair topology inference problem. Our idea is to solve

$$\mathbf{A}_f := \underset{\mathbf{A}\in\mathcal{A}}{\text{argmin}} \left\{ \|\mathbf{s}^\top \mathbf{A}\|_1 + \beta r(\mathbf{A}, \hat{\mathbf{A}}) \right\}, \quad \text{s.to} \;\; \mathbf{A}\hat{\mathbf{A}} = \hat{\mathbf{A}}\mathbf{A}, \tag{3}$$

where the commutation constraint $\mathbf{A}\hat{\mathbf{A}} = \hat{\mathbf{A}}\mathbf{A}$ ensures that $\hat{\mathbf{A}}$ and $\mathbf{A}_f$ share the same set of eigenvectors [cf. (2)]. Given $\mathbf{s}$ and $\hat{\mathbf{A}}$, we think of the solution $\mathbf{A}_f$ as a rewired version of the original graph-shift operator that minimizes the bias measure $\rho(\mathbf{A}) = \|\mathbf{s}^\top \mathbf{A}\|_1$. The convex regularization term $r(\cdot,\cdot)$ is included to control deviations from $\hat{\mathbf{A}}$, and hence account for utility. Here, we choose $r(\mathbf{A}, \hat{\mathbf{A}}) := \|\mathbf{A} - \hat{\mathbf{A}}\|_{1,1}$ and note that $\beta > 0$ is a hyperparameter to adjust the trade-off between fairness and utility. Finally, $\mathcal{A}$ is a convex set that specifies other desired properties of $\mathbf{A}_f$. We consider

$$\mathcal{A} := \left\{ \mathbf{A} \mid A_{ij} \geq 0, \mathbf{A} = \mathbf{A}^\top, \|\mathbf{A}\|_{1,1} = \|\hat{\mathbf{A}}\|_{1,1} \right\}, \tag{4}$$

requiring the rewired graph operator to: (i) have non-negative weights; (ii) be symmetric since $\mathcal{G}$ is undirected; and (iii) preserve the total edge weight sum in the given $\hat{\mathbf{A}}$.

For the aforementioned choices of $r(\cdot,\cdot)$ and $\mathcal{A}$, (3) is a convex optimization problem. In fact, it is a linear program (LP) and hence it can be solved using off-the-shelf methods. Once $\mathbf{A}_f$ is obtained, it can be flexibly used as a surrogate graph in MLoG frameworks such as GNNs; see Fig. 1.

Remark 1 (Eigendecomposition-free filter design). From the discussion in Section III-B, it follows that the graph rewiring formulation in (3) is equivalent to a fairness-aware graph filter design to reduce ρ. Unlike [28] that imposes filter specifications in the graph spectral domain, the implicit design of this paper does not require computing the GFT basis.

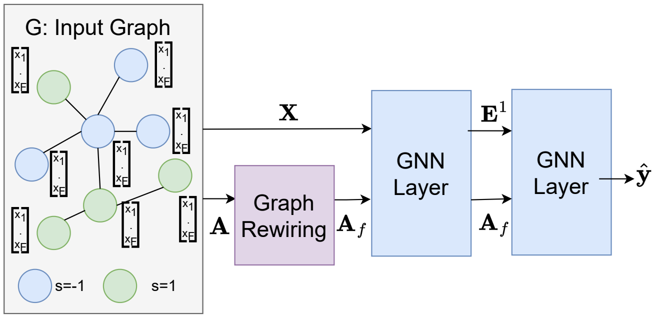

Fig. 1: The employment of the graph rewiring module within a standard two-layer GNN-based learning pipeline as a topology debiasing operator. Here, $\mathbf{E}^l \in \mathbb{R}^{N\times F^{'}}$ represents the hidden node embeddings output by the lth GNN layer.

Remark 2 (Choice of $\mathcal{A}$). The convex set $\mathcal{A}$ plays a critical role in the proposed framework. Indeed, for over-constrained specifications the feasible set might reduce to a singleton, where $\mathbf{A}_f = \hat{\mathbf{A}}$. A detailed study is omitted due to lack of space [41], and will be reported in the journal version.

TABLE I: Dataset statistics.

Dataset	$\lvert\mathcal{S}_{-1}\rvert$	$\lvert\mathcal{S}_1\rvert$	$\lvert\mathcal{Y}_{-1}\rvert$	$\lvert\mathcal{Y}_1\rvert$	$\lvert\mathcal{E}\rvert$
NBA	228	82	152	158	7115
SPokec-n	49	319	101	267	970
SPokec-z	102	343	118	327	848

IV. Experimental Results

Here we assess the effectiveness of the proposed graph rewiring mechanism via experiments on various network datasets. We carry out comparisons with several state-of-the-art bias mitigation strategies for MLoG.

A. Datasets and experimental setup

Datasets. To evaluate $\mathbf{A}_f$, sampled versions of the Pokec social networks are created and utilized together with the NBA dataset [20]. The complexity of solving for $\mathbf{A}_f$ in (3) using CVX [46] limits rewiring feasibility to small/medium-scale graphs. For this reason, Pokec networks [20] are clustered into smaller communities by using greedy modularity maximization. The clusters providing the best balance in terms of different sensitive group/class sizes are employed for the experiments. For these graphs, the user's region is utilized as the sensitive attribute, and the labels for the node classification task are the binarized working field of the users. The sampled versions of Pokec-n and Pokec-z are denoted by "SPokec-n" and "SPokec-z", respectively. Furthermore, the NBA graph is built based on the performance statistics and other attributes (such as, nationality, age, and salary) of 400 NBA players.

TABLE II: Comparative performance evaluation results. Utility and fairness metrics for the NBA and sampled Pokec datasets.

	NBA			SPokec-n			SPokec-z		
	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)
GNN	61.72 ± 6.2	9.24 ± 5.2	10.66 ± 11.1	76.57 ± 2.4	10.63 ± 6.2	6.64 ± 4.1	72.79 ± 3.8	3.91 ± 4.6	10.77 ± 5.6
Adversarial [20]	61.94 ± 4.0	4.93 ± 2.3	7.20 ± 7.7	76.57 ± 2.7	8.95 ± 5.8	5.11 ± 2.0	70.63 ± 2.8	9.56 ± 5.6	7.96 ± 7.9
EDITS [26]	65.38 ± 2.8	6.49 ± 4.6	10.31 ± 6.4	75.37 ± 3.0	$\mathbf{3.70} \pm 3.9$	$\mathbf{3.10} \pm 1.9$	71.53 ± 4.8	6.04 ± 4.2	7.63 ± 4.7
$\tilde{\mathbf{h}}^{\text{fair}}$+ GNN [28]	60.43 ± 6.8	5.89 ± 4.4	10.74 ± 10.5	75.67 ± 1.7	5.43 ± 2.6	4.91 ± 1.8	72.07 ± 3.4	4.14 ± 4.1	7.23 ± 7.8
$\mathbf{A}_f$+ GNN	$\mathbf{68.39} \pm 6.9$	$\mathbf{3.28} \pm 1.6$	$\mathbf{6.61} \pm 4.8$	$\mathbf{78.51} \pm 3.0$	6.27 ± 4.1	$\mathbf{3.09} \pm 2.6$	$\mathbf{72.97} \pm 1.1$	$\mathbf{3.04} \pm 4.5$	$\mathbf{2.49} \pm 3.8$

Edges are created based on "follow" relationships among these players on Twitter. For node classification, the binary labels are generate based on the player's salaries, where their nationalities are utilized as the sensitive attributes. Dataset statistics are compiled in Table I, where $\mathcal{S}_i$ and $\mathcal{Y}_i$ represent the set of nodes with sensitive attribute and class label i, respectively. Note that $N = |\mathcal{S}_{-1}| + |\mathcal{S}_1| = |\mathcal{Y}_{-1}| + |\mathcal{Y}_1|$.

Evaluation metrics. Utility is reported in terms of node classification accuracy. Two widely adopted group-fairness metrics are reported as well, namely *statistical parity*: $\Delta_{SP} = |P(\hat{y} = 1 \mid s = -1) - P(\hat{y} = 1 \mid s = 1)|$ and *equal opportunity*: $\Delta_{EO} = |P(\hat{y} = 1 \mid y = 1, s = -1) - P(\hat{y} = 1 \mid y = 1, s = 1)|$, where y denotes the ground truth label, and $\hat{y}$ represents the predicted label. Here, statistical parity quantifies the decoupling of positive rate from the sensitive attribute, and equal opportunity measures the level of the independence of true positive rate from the sensitive attribute. Lower values of Δ_{SP} and Δ_{EO} are desirable for better fairness performance [20].

Implementation details. Node classification is used to evaluate the effectiveness of the proposed graph rewiring mechanism, where $\mathbf{A}_f$ is input to GNN layers as the graph structure in a two-layer graph convolutional network (GCN) [4], see Fig. 1. The weights of the GNN model are initialized utilizing Glorot [47], and trained for 400 epochs by employing Adam optimizer [48] together with a learning rate of 0.0005 and ℓ_2 weight decay factor of 10^{-5}. Hidden dimension of the node representations is selected as 128 on all datasets. The model is trained over 40% of the vertices, while the remaining nodes are evenly split onto validation and test sets. The hyperparameter β is selected as 0.00, 0.01, and 0.00 for SPokec-z, SPokec-n, and NBA graphs, respectively, via grid search among the values $\{0.00, 0.01, 0.1, 1.0\}$. For all experiments, results are obtained for five random data splits, and their average along with the standard deviations are reported.

Baselines. We also report the results for several fairness-aware baselines: adversarial regularization [20], EDITS [26], and fair graph filter $\tilde{\mathbf{h}}^{fair}$ [28]. Adversarial regularization is a widely utilized bias mitigation technique, where an adversary is trained to predict the sensitive attributes. For adversarial regularization, the regularization parameter is tuned via a grid search over $\{0.1, 1, 10, 100, 1000\}$ (the weight of the classification loss is 1). In addition, EDITS [26] is a model-agnostic debiasing strategy that alleviates the bias in attributed networks before they are input to any MLoG framework. Specifically, it provides debiased versions of the nodal attributes and the graph topology, which are then fed to the GCN network used here for node classification. For EDITS, the threshold proportion used to sparsify the debiased topology is tuned among the values $\{0.015, 0.02, 0.06, 0.29\}$, where these values are suggested for other datasets in [26]. Finally, the fair graph filter $\tilde{\mathbf{h}}^{fair}$ is designed offline, and then used to process the inputs to each GNN layer as suggested in [28].

B. Results and discussion

GNN-based node classification results are tabulated in Table II. The natural (fairness-agnostic) baseline is to employ the exact same GNN model but with the original graph $\hat{\mathbf{A}}$, which is denoted as GNN. Furthermore, Adversarial, EDITS, and $\tilde{\mathbf{h}}^{fair}$+ GNN in Table II correspond to the fairness-aware baselines: adversarial regularization [20], EDITS [26], and fair filter design in [28], respectively. Results in Table II demonstrate that a GNN using $\mathbf{A}_f$ typically achieves the best fairness performance relative to state-of-the-art fairness-aware baselines. While EDITS [26] leads to better/similar fairness measures on SPokec-n, this fairness improvement is accompanied by a drop in utility. Furthermore, our results show that graph rewiring always achieves the best classification accuracy, although it is mainly designed for bias attenuation. This utility gain can be attributed to the denser connectivity in $\mathbf{A}^f$ compared to $\hat{\mathbf{A}}$, which leads to more powerful representations but at the price of higher computational complexity for the message passing operation. Overall, results corroborate the effectiveness of our novel approach in mitigating topological bias, while also providing better utility metrics relative to both fairness-agnostic and fairness-aware baselines.

V. Conclusion

We developed a fairness-aware edge rewiring strategy whose graph output can be flexibly employed in several MLoG pipelines. Our formulation is inspired by a debiasing signal filtering approach, which we reinterpret as a topology inference (or edge filtering) problem to minimize a well-grounded bias criterion. Node classification experiments on real-world networks demonstrate our novel scheme results in better fairness together with better utility relative to state-of-the-art baselines. Exciting future directions in this space include custom-made scalable algorithms and exploring nonlinear relations between sensitive attributes and the graph topology as a bias measure.

References

[1] E. D. Kolaczyk and G. Csárdi, *Statistical Analysis of Network Data with R*. Springer, 2014.

[2] W. L. Hamilton, "Graph representation learning," *Synthesis Lectures on Artifical Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159, 2020.

[3] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy, "Machine learning on graphs: A model and comprehensive taxonomy," *J. Mach. Learn. Res.*, vol. 23, no. 89, pp. 1–64, 2022.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2016, pp. 1–14.

[5] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[6] G. Leus, A. G. Marques, J. M. Moura, A. Ortega, and D. I. Shuman, "Graph signal processing: History, development, impact, and outlook," *IEEE Signal Process. Mag.*, vol. 40, no. 4, pp. 49–60, 2023.

[7] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[8] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[9] R. Shafipour, A. Khodabakhsh, G. Mateos, and E. Nikolova, "Digraph Fourier transform via spectral dispersion minimization," in *Proc. Int. Conf. Acoustics, Speech, Signal Process.*, 2018, pp. 6284–6288.

[10] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, 2017.

[11] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274–288, 2016.

[12] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, "Graph filters for signal processing and machine learning on graphs," *arXiv preprint arXiv:2211.08854*, 2022.

[13] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 764–778, 2016.

[14] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128–138, 2020.

[15] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 5680–5695, 2020.

[16] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Process. Mag.*, no. 6, pp. 117–127, 2020.

[17] K. R. Varshney, *Trustworthy Machine Learning*. Chappaqua, NY, USA: Independently Published, 2022.

[18] D. Pedreshi, S. Ruggieri, and F. Turini, "Discrimination-aware data mining," in *Proc. Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2008, pp. 560–568.

[19] A. Beutel, J. Chen, Z. Zhao, and E. H. Chi, "Data decisions and theoretical implications when adversarially learning fair representations," in *Fairness, Accountability, and Transparency in Machine Learning Workshop (FAT/ML)*, 2017, pp. 1–5.

[20] E. Dai and S. Wang, "Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information," in *Proc. ACM Int. Conf. on Web Search and Data Mining (WSDM)*, March 2021, pp. 680–688.

[21] A. Bose and W. Hamilton, "Compositional fairness constraints for graph embeddings," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 715–724.

[22] M. Buyl and T. D. Bie, "The KL-divergence between a graph model and its fair I-projection as a fairness regularizer," in *Proc. Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 351–366.

[23] O. D. Kose and Y. Shen, "Fast&Fair: Training acceleration and bias mitigation for GNNs," *Trans. Mach. Learn. Res.*, pp. 1–25, 2023.

[24] ——, "Fair contrastive learning on graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 475–488, 2022.

[25] I. Spinelli, S. Scardapane, A. Hussain, and A. Uncini, "Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning," *IEEE Trans. Artificial Intell.*, vol. 3, no. 3, pp. 344–354, 2021.

[26] Y. Dong, N. Liu, B. Jalaian, and J. Li, "EDITS: Modeling and mitigating data bias for graph neural networks," in *Proc. ACM Web Conference*, 2022, pp. 1259–1269.

[27] O. D. Kose, Y. Shen, and G. Mateos, "Fairness-aware graph filter design," in *Proc. Asilomar Conf. on Signals, Systems, Computers*, 2023; see also arXiv:2303.11459 [cs.LG], pp. 1–6.

[28] O. D. Kose, G. Mateos, and Y. Shen, "Fairness-aware optimal graph filter design," *IEEE J. Sel. Topics Signal Process.*, pp. 1–13, 2024 (Early Access).

[29] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.

[30] F. R. Chung and F. C. Graham, *Spectral Graph Theory*. American Mathematical Soc., 1997, vol. 92.

[31] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recognit.*, vol. 41, no. 11, pp. 3328–3342, 2008.

[32] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *Int. Conf. on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 2011, pp. 1–8.

[33] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 912–919.

[34] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Mach. Learn.*, vol. 56, no. 1, pp. 209–239, 2004.

[35] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, 2019.

[36] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. IEEE*, vol. 106, no. 5, pp. 787–807, 2018.

[37] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[38] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 563–575, 2014.

[39] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.

[40] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2016.

[41] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, 2017.

[42] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 3, pp. 481–496, 2017.

[43] B. Hofstra, R. Corten, F. Van Tubergen, and N. B. Ellison, "Sources of segregation in social networks: A novel approach using Facebook," *Am. Sociol. Rev.*, vol. 82, no. 3, pp. 625–656, 2017.

[44] S. Hajian and J. Domingo-Ferrer, "A methodology for direct and indirect discrimination prevention in data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 7, pp. 1445–1459, July 2013.

[45] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, 2014.

[46] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[47] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, May 2010, pp. 249–256.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2015.