Fairness-Aware Optimal Graph Filter Design

O. Deniz Kose, Gonzalo Mateos, *Senior Member, IEEE*, and Yanning Shen

***Abstract*—Graphs are mathematical tools that can be used to represent complex real-world interconnected systems, such as financial markets and social networks. Hence, machine learning (ML) over graphs has attracted significant attention recently. However, it has been demonstrated that ML over graphs amplifies the already existing bias towards certain under-represented groups in various decision-making problems due to the information aggregation over biased graph structures. Faced with this challenge, here we take a fresh look at the problem of bias mitigation in graph-based learning by borrowing insights from graph signal processing. Our idea is to introduce predesigned graph filters within an ML pipeline to reduce a novel unsupervised bias measure, namely the correlation between sensitive attributes and the underlying graph connectivity. We show that the optimal design of said filters can be cast as a convex problem in the graph spectral domain. We also formulate a linear programming (LP) problem informed by a theoretical bias analysis, which attains a closed-form solution and leads to a more efficient fairness-aware graph filter. Finally, for a design whose degrees of freedom are independent of the input graph size, we minimize the bias metric over the family of polynomial graph convolutional filters. Our optimal filter designs offer complementary strengths to explore favorable fairness-utility-complexity tradeoffs. For performance evaluation, we conduct extensive and reproducible node classification experiments over real-world networks. Our results show that the proposed framework leads to better fairness measures together with similar utility compared to state-of-the-art fairness-aware baselines.**

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

I. Introduction

We live in the era of connectivity, where the actions of humans and devices are increasingly driven by their relations to others. Concurrently, a significant amount of data describing different interconnected systems, such as social networks, the Internet of Things (IoT), the Web, and financial markets, is increasingly available. Processing and learning from such data can provide significant understanding and advancements for the corresponding networked systems [16], [26]. In this context, machine learning (ML) over graphs has attracted increasing attention [7], [24], since graphs are widely utilized to represent complex underlying relations in real-world networks [38].

These relational patterns can be captured by graph edges, while attributes of nodes (nodal features) can be interpreted as signals defined on the vertices. For example, in a social network, user ages can be modeled as a graph signal, and the friendship information can be encoded by the edges. Graph signal processing (GSP) extends the tools in classical signal processing to graph signals [41], such as frequency analysis, sampling, and filtering [22], [23], [37], [45], [48], [52]. GSP and ML over graphs are closely intertwined, where the tools in one domain can be useful in the other one [10], [41]. For instance, it has been demonstrated that graph neural networks (GNNs) can be designed, analyzed, and improved by leveraging GSP-based insights [10], [13], [14], which underscores the advancements that can be made by cross-pollinating the findings in both domains. In this paper, we align with this vision and leverage GSP advances to enhance fairness in ML over graphs pipelines.

The pursuit of fairness in ML over graphs: Despite the growing interest in learning over graphs, the widespread deployment of these algorithms in real-world decision systems depends heavily on how socially responsible they are. Motivated by this concern, fairness in ML algorithms has attracted significant attention recently [20], [39], [43]. This work focuses on group fairness, which ensures that the learning algorithms incur no performance gap with respect to sensitive/protected attributes (such as ethnicity and religion). For example, the predictions of a job recommendation algorithm should be independent of the gender of applicants for a fair algorithm with respect to the sensitive attribute gender. Moreover, throughout this paper, *algorithmic bias* refers to the stereotypical correlations the learning algorithms encode and further propagate with respect to these sensitive attributes. Despite how critical the fairness of algorithms is for their applicability in real-world decision systems, several studies have demonstrated that ML models propagate the historical bias within the training data and lead to discriminatory results in ensuing applications [3], [39], [43]. Specific to graph-based learning, the utilization of graph structure in the algorithm design has been shown to amplify the already existing bias [9]. Recognizing these compounded challenges, recent works focus on fairness-aware learning over graphs and advocate different techniques to mitigate bias, such as adversarial regularization [4], [9], fairness constraints [5], [28], and fairness-aware graph data augmentation [11], [30], [54]; see also Section II for additional discussion on related work.

Manuscript received 23 June 2023; revised 14 December 2023; accepted 3 January 2024. Date of publication 10 January 2024; date of current version 3 July 2024. This Work was supported by NSF under Awards CCF-1750428, CCF-1934962, and ECCS 2207457. Preliminary ideas that inspired this work presented at the Asilomar Conference on Signals, Systems, and Computers, 2023. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Maria Sabrina Greco. *(Corresponding author: Yanning Shen.)*

O. Deniz Kose and Yanning Shen are with the Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92617 USA (e-mail: okose@uci.edu; yannings@uci.edu).

Gonzalo Mateos is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: gmateosb@ur.rochester.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSTSP.2024.3350508, provided by the authors.

Digital Object Identifier 10.1109/JSTSP.2024.3350508

Proposed approach and innovations in context: In this study, we advocate fairness-aware optimal graph filter designs. In order to mitigate bias derived from the graph topology, we subsequently introduce these predesigned filters within standard ML pipelines. To this end, we introduce a bias metric, ρ, which can be employed in graph-based unsupervised learning approaches and measures the linear correlation between an effective (filter-dependent) connectivity pattern and the sensitive attributes. We show that the ρ-minimizing optimal filter design can be cast as a convex problem in the graph spectral domain. While this proposed approach is remarkably effective in mitigating graph-amplified biases, the total number of optimization variables is equal to the input graph size. Accordingly, solving the optimization problem becomes computationally expensive for large input graphs. For a more efficient fairness-aware solution, we carry out a bias analysis and upper bound the bias metric ρ by bringing to bear GSP notions. Based on these theoretical findings, we formulate a novel linear programming (LP) filter design problem that attains a closed-form solution minimizing the derived upper bound. We finally propose a design whose degrees of freedom are independent of the size of the input graph, by minimizing ρ over the family of polynomial graph convolutional filters.

Our previous endeavor [31] is also built upon spectral analysis of graph signals, where a fairness-aware *dimensionality reduction* algorithm was developed. However, in [31], the information carried in certain frequencies is completely removed, which can adversely affect the overall utility (accuracy for node classification) of the underlying ML task. Instead, in the present work, we propose a suite of bias mitigation approaches to effectively filter out traces of the sensitive attribute signal (e.g., race, gender in social networks), while also offering the flexibility to delineate favorable fairness-utility-complexity tradeoffs in ML over graphs. Furthermore, unlike the intuitive but heuristic approach in the conference precursor to this paper [29], the fairness-aware graph filter designs proposed here are rooted on well-defined optimality criteria.

Summary of contributions: Overall, our contributions are:

i) We introduce a novel, correlation-based bias metric for graphs, which can facilitate fairness-aware unsupervised learning from network data;

ii) We show that filtering nodal representations which are obtained via graph aggregation can be used to manipulate the bias metric. An optimal graph filter is designed to minimize ρ by solving a convex optimization problem in the spectral domain;

iii) For a more efficient bias mitigation solution, we upper bound ρ by utilizing GSP-based tools and then minimize this surrogate cost, leading to an LP problem that attains a closed-form optimal solution. By restricting the search to the class of polynomial graph convolutional filters, the number of optimization variables decouples from the input graph size, and the resulting fairness-aware filters can be implemented in a distributed fashion;

iv) The novel filter designs are versatile and can be employed in different stages of the learning pipeline, as well as for various graph-based learning frameworks; and

V) Comprehensive experimental results for node classification on real-world networks corroborate the effectiveness of the proposed methods in mitigating bias while providing comparable utility to state-of-the-art fairness-aware baselines. In the interest of reproducible research, the code used to obtain all results in this paper is publicly available.

Notation: The entries of a matrix $\mathbf{V}$ and a vector $\mathbf{v}$ are denoted by V_{ij} and v_i, respectively. Calligraphic capital letters are utilized to represent sets. $\mathbf{I}_N$ refers to an $N \times N$ identity matrix. The notation $^\top$ stands for the transpose operation. For a vector $\mathbf{v}$, $\operatorname{diag}(\mathbf{v})$ represents a diagonal matrix whose ith diagonal entry equals to v_i. The ℓ_p-norm of vector $\mathbf{v}$ is given by $\|\mathbf{v}\|_p := (\sum_{i=1}^{n} |v_i|^p)^{1/p}$.

II. Related Work

Here, we briefly review relevant related work to better position our contributions in context.

A. Graph Filters

Extending classical signal processing tools to networked systems, graph filters are specific operators to manipulate graph signals. The existing literature generally focuses on linear graph filters represented by polynomials of a graph-shift operator [14], [22], [45], [48], [50], [51]. Graph filters are utilized for a number of applications, including but not limited to modeling the dynamics of opinion formation in social networks [18], [42], or modeling the diffusion/percolation dynamics over networks [40], [51]. Recently, with the success of graph neural networks (GNNs) for a number of graph-based tasks, graph filters have attracted increasing attention as the key component of GNNs [14], [21], [36], [46], [58]. However, to the best of our knowledge, there has been no prior attempt to examine the benefits of pretrained filters towards decorrelating learned nodal representations from sensitive attributes. So far, optimal graph filter designs have not incorporated fairness criteria.

B. Fairness-Aware Learning on Graphs

In the fairness-aware graph-based learning domain, [44] is a pioneering study that proposes a bias mitigation solution for random walk-based algorithms. Moreover, motivated by its success in general fairness-aware ML, adversarial regularization is also employed by several graph-based ML frameworks [4], [9], [12], [15]. Specifically, [9] focuses on partially available sensitive attributes, and [12] considers knowledge graphs. By modeling the sensitive attribute signal in the prior distribution, [6] proposes a Bayesian strategy for fair node representation learning. In addition, [35] links the subgroup generalization to accuracy disparity based on a PAC-Bayesian analysis, while [56] presents multiple strategies to reduce the algorithmic bias in the representations of heterogeneous information networks. There is also a line of work that designs fair graph data augmentations to mitigate the bias within nodal features and the graph topology [1], [27], [30], [54]. Finally, with a specific focus on link prediction, [33], [34] introduce fairness-aware strategies that alter the adjacency matrix, while [5] employs a fairness-aware regularizer. Unlike most of these works, the proposed strategies herein are based on a theoretical bias analysis and enjoy well-defined optimality. Furthermore, the collection of fairness-aware graph filters designed in Section V can be employed in a versatile manner as both a pre-processing and post-processing operator in a number of graph-based learning environments; see also the numerical tests in Section VI. While the draft of this paper was being finalized, we became aware of an interesting unpublished preprint [32] that explores fairness for GSP-based graph mining applications with a markedly different goal than ours. Indeed, [32] advocates a GNN framework as a surrogate of a fairness-aware graph filter, and here we design graph filters to mitigate bias in general ML on graphs pipelines. The approach in [32] is to "edit" the input graph signal for fairness enhancement and does not focus on optimal filter design. Overall, our study is the first attempt to design fair graph filters to mitigate intrinsic bias by cross-pollinating the tools of GSP and ML over graphs.

III. Preliminaries and Problem Statement

The focus of this study is to mitigate bias in graph-based learning algorithms by employing graph filters for a given undirected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{v_1, v_2, \ldots, v_N\}$ denotes the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Connectivity of the input graph is encoded in the symmetric adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$, where $A_{ij} = 1$ if and only if $(v_i, v_j) \in \mathcal{E}$. In addition, $\mathbf{X} \in \mathbb{R}^{N\times F}$ represents the nodal features of $\mathcal{G}$, whose columns are graph signals (one per feature). The diagonal degree matrix is $\mathbf{D} \in \mathbb{R}^{N\times N}$, where D_{ii} denotes the degree of v_i. Let $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ denote the normalized graph Laplacian matrix, where the normalized adjacency matrix is represented by $\hat{\mathbf{A}} := \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$.

The *sensitive attribute* is the nodal feature (such as ethnicity, religion) on which the decisions should not be dependent for fair decision-making. Herein, the sensitive attribute is assumed to be binary and is denoted by $\mathbf{s} \in \{-1,1\}^N$. 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, some vertices have (e.g., binary) labels y_i. For concrete examples of nodal features, labels, and sensitive attributes in several real-world network datasets, see Section VI-A.

A. Graph Signal Processing Fundamentals

The graph Fourier transform (GFT) is an orthonormal transform that provides the representation of a graph signal $\mathbf{z} \in \mathbb{R}^N$ in the graph spectral domain [8], [17], [52]. Specifically, taking the GFT of a graph signal amounts to projecting the signal onto a space spanned by the orthogonal eigenvectors of the positive semi-definite (PSD) normalized graph Laplacian matrix $\mathbf{L}$ [52]. Let the eigendecomposition of the normalized Laplacian be $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ collects the non-negative eigenvalues and $\mathbf{V}$ is the matrix of Laplacian eigenvectors. Then, the GFT of the graph signal $\mathbf{z} \in \mathbb{R}^N$ is given by $\tilde{\mathbf{z}} = \mathbf{V}^\top\mathbf{z}$. Graph frequencies correspond to the eigenvalues of the Laplacian (a measure of smoothness of the eigenvectors with respect to the graph), meaning that the GFT decomposes signals into frequency modes (i.e., the eigenvectors of $\mathbf{L}$) of different variability over $\mathcal{G}$.

In classical signal processing, filters are utilized to manipulate signals such that their, e.g., unwanted components are attenuated or removed. Similarly, graph filters can be used to modify graph signals for different purposes, including graph signal classification [2], [59], smoothing, and denoising [53], [57]. Filtering an input graph signal $\mathbf{z}_{\text{in}} \in \mathbb{R}^N$ via a filter with frequency response $\tilde{\mathbf{h}} := [\tilde{h}_1, \ldots, \tilde{h}_N]^\top$ can be mathematically expressed as (e.g., [14], [41], [52])

$$\mathbf{z}_{\text{out}} = \mathbf{V}\underbrace{\text{diag}(\tilde{h}_1, \ldots, \tilde{h}_N)\mathbf{V}^\top\mathbf{z}_{\text{in}}}_{\text{Frequency domain filtering}}. \tag{1}$$

Therefore, filtering in the frequency domain corresponds to point-wise multiplication of the input signal's GFT, $\tilde{\mathbf{z}}_{\text{in}}$, with the frequency response of graph filter, $\tilde{\mathbf{h}}$. Identity (1) is akin to a convolution theorem for graph signals.

Convolutional filters are commonly utilized in ML due to their computational efficiency and parameter-sharing property, which motivates their generalization to the graph domain. Graph convolutional filters' input-output relation can be described via shift and sum operations. Specifically, a graph convolutional filter of order L is a linear mapping of the form

$$\mathbf{H} := \sum_{l=0}^{L-1} h_l \hat{\mathbf{A}}^l, \tag{2}$$

with input-output relation $\mathbf{z}_{\text{out}} = \mathbf{H}\mathbf{z}_{\text{in}}$. Here, $\mathbf{h} := [h_0, \ldots, h_{L-1}]^\top \in \mathbb{R}^L$ are the filter coefficients, and $\hat{\mathbf{A}}$ is the selected graph-shift operator [49]. Notice how (2) resembles a finite impulse response (FIR) filter, with the identification of $\hat{\mathbf{A}}^l$ as an lth-order shift operator acting on graph signals [22]. By utilizing the spectral decomposition of $\hat{\mathbf{A}}$, $\mathbf{H}$ can also be written as $\mathbf{H} = \mathbf{V}(\sum_{l=0}^{L-1} h_l(\mathbf{I}_N - \mathbf{\Lambda})^l)\mathbf{V}^\top$. Thus, the filter is diagonalized by the graph's eigen basis and $\tilde{\mathbf{H}} := \sum_{l=0}^{L-1} h_l(\mathbf{I}_N - \mathbf{\Lambda})^l = \text{diag}(\tilde{\mathbf{h}})$ can be regarded as the frequency response of $\mathbf{H}$, where $\tilde{\mathbf{h}} = [\tilde{h}_1, \ldots, \tilde{h}_N]^\top$ collects the filter's spectral response at the N discrete graph frequencies. Moreover, notice that upon defining the $N \times L$ Vandermonde matrix $\mathbf{\Psi}$, where $\Psi_{ij} := (1 - \Lambda_{ii})^{j-1}$, then it holds that $\tilde{\mathbf{h}} := \mathbf{\Psi}\mathbf{h}$ [49].

Unlike the general frequency response in (1), for polynomial graph convolutional filters (2) one introduces an explicit parameterization $\tilde{h}_i = \sum_{l=0}^{L-1} h_l(1-\lambda_i)^l$. Accordingly, the number of filter coefficients (or degrees of freedom) is L, *independent* of the graph size N.

B. Problem Statement

In this paper, given $\mathcal{G}$ and $\mathbf{s}$, we address the problem of designing graph filters with frequency response $\tilde{\mathbf{h}} \in \mathbb{R}^N$, so that the bias caused by the graph topology can be attenuated with the application of the designed filters in the learning algorithm. A possible application of the fairness-aware graph filter in a GNN-based learning pipeline is depicted in Fig. 1. As we elaborate in Section IV-B, bias attenuation will be pursued by minimization of a judicious bias metric; namely, the linear correlation between $\mathbf{s}$ and the effective graph aggregation operator that results upon filtering with $\tilde{\mathbf{h}}$.

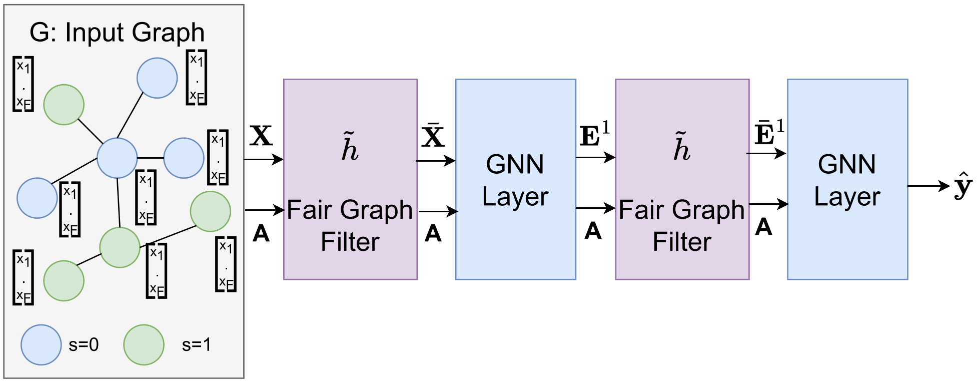

Fig. 1. Employment of a fairness-aware graph filter $\tilde{\mathbf{h}}$ within a standard two-layer GNN-based learning pipeline as a pre-trained bias mitigation operator. Here, $\mathbf{E}^l \in \mathbb{R}^{N \times F'}$ represents the hidden node embeddings output by the lth GNN layer.

IV. Bias Mitigation Rationale and Criterion

In this section, we first motivate our filtering approach for bias mitigation and provide a graph spectral domain illustration of the fairness-utility tradeoff. We then propose a filter-dependent bias measure that will serve as a criterion for our subsequent designs.

A. Spectrum Analysis

The homophily principle suggests that nodes with similar attributes are more likely to connect in networks, which hints at denser connectivity between the nodes with the same sensitive attributes and also with the same label [19]. Hence, both the sensitive attributes $\mathbf{s}$ and node labels $\mathbf{y}$ are expected to be smooth signals over $\mathcal{G}$. In the GSP parlance, this implies higher energy concentration for $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{y}}$ over lower frequencies. Now, we wish to design a filter that preserves the necessary information for a downstream task (node classification in this paper) after “filtering out” traces of the sensitive attribute. Naturally, the extent of the overlap between the spectra of $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{y}}$ plays an important role in the feasibility of said fairness-aware filter design.

To examine this tension, the GFT coefficients in $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{y}}$ over lower frequencies are depicted in Fig. 2 for two real-world social networks with more than 6000 nodes. For additional details of the datasets, see Section VI-A. Fig. 2 depicts normalized GFT coefficients, i.e., $\tilde{\mathbf{s}}/\max(\tilde{\mathbf{s}})$ and $\tilde{\mathbf{y}}/\max(\tilde{\mathbf{y}})$, to better highlight the discrepancy between these two signals. As expected, it can be observed that the spectra of $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{y}}$ exhibit similar characteristics. However, there are certain (low) frequencies where the magnitudes of $\tilde{\mathbf{s}}$ are markedly higher than those of $\tilde{\mathbf{y}}$. This subtle but important discordance (which is not just an artifact of these datasets) inspires our pursuit of frequency-selective graph filters for bias mitigation. The goal is to attenuate the sensitive information while preserving graph signals necessary for downstream ML tasks.

B. Bias Metric

It has been demonstrated that leveraging graph structure in learning algorithms amplifies already existing bias due to the biased connectivity information [9]. To exemplify this important point, in social networks, users (nodes) are often more likely to connect to other users with the same sensitive attributes (e.g., ethnicity, religion). This leads to denser connectivity between the nodes from the same sensitive groups, and hence a graph structure that is highly correlated with the sensitive attributes [19]. Motivated by this, the linear correlation between the sensitive

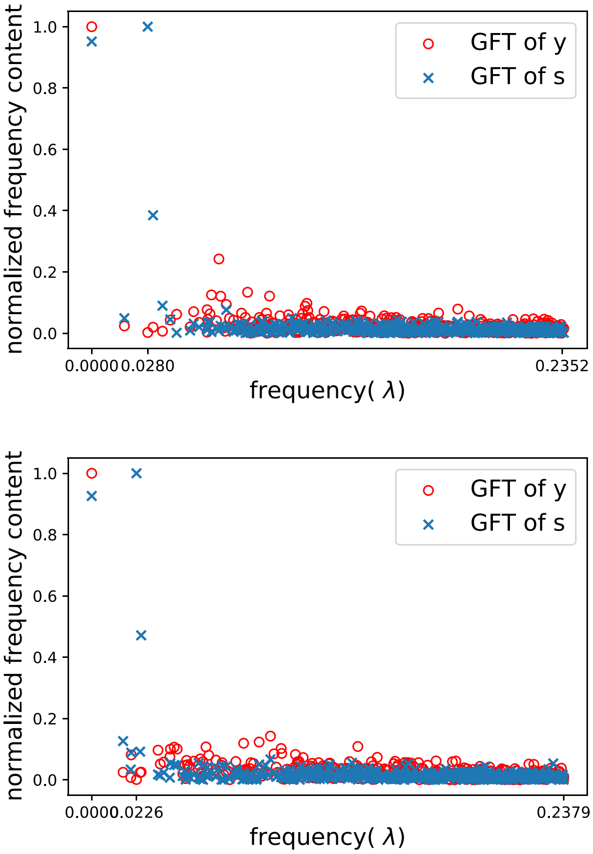

Fig. 2. Spectra of the graph signals $\mathbf{s}$ (sensitive attributes) and $\mathbf{y}$ (labels) over different graph frequencies, for the real-world social network datasets (top) Pokec-z and (bottom) Pokec-n. Dataset statistics are presented in Table I. There are few low frequencies where the magnitudes of $\tilde{\mathbf{s}}$ are markedly higher than those of $\tilde{\mathbf{y}}$.

attribute signal $\mathbf{s}$ and graph topology is considered for the ensuing bias analysis and mitigation strategy.

Several graph-based learning approaches rely on node representations obtained via local aggregation of information (possibly followed by a pointwise non-linearity) [24], [41]. In the simplest possible terms, this process can be summarized as

$$\mathbf{R} = \hat{\mathbf{A}}\mathbf{X}, \tag{3}$$

where $\mathbf{R}$ denotes the aggregated node representations, and $\mathbf{X}$ is the input graph signal (or the representations from the previous layer as in Fig. 1); see, e.g., [7], [24]. In (3), we have purposely omitted learnable weights to simplify the notation while retaining the components essential to our argument. Hence, if a filtered graph signal $\bar{\mathbf{X}} = \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top\mathbf{X}$ is input, the obtained representation becomes

$$\begin{aligned}\mathbf{R}^{\text{f}} &= \hat{\mathbf{A}}\bar{\mathbf{X}}\\ &= \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\mathbf{V}^\top\bar{\mathbf{X}}\\ &= \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\mathbf{V}^\top\mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top\mathbf{X}\\ &= \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top\mathbf{X}\\ &= \bar{\mathbf{A}}\mathbf{X},\end{aligned} \tag{4}$$

where $\bar{\mathbf{A}} := \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top$. Therefore, if a filtered signal $\bar{\mathbf{X}}$ is fed to the aggregation process, the effective network operator that is utilized in the information aggregation becomes $\bar{\mathbf{A}}$. In other words, while forming the representation of node v_i, the information coming from other nodes v_j, where $\{j : \bar{A}_{ij} \neq 0\}$, will be combined with strength proportional to the values of $\bar{A}_{ij}$. All in all, (4) shows that filtering the input signal $\mathbf{X}$ can be equivalently viewed as a mechanism to modify the network aggregation operator used to obtain node representations.

Building on this quite simple but key observation, the linear correlation between the sensitive attributes $\mathbf{s}$ and $\bar{\mathbf{A}}$ is employed as a bias measure. This metric is inspired by the finding that many real-world graphs exhibit denser connectivity among nodes with common sensitive attributes; i.e., $\mathbf{s}$ is a smooth signal as shown in Fig. 2. Hence, aggregating information predominantly from nodes with the same sensitive attribute will exacerbate algorithmic bias [27]. For this reason, it is desirable to have a lower correlation between $\mathbf{s}$ and the effective network operator, $\bar{\mathbf{A}}$, which governs information aggregation when filters are applied to the input signals. This correlation is proportional to $|\mathbf{s}^\top\bar{\mathbf{A}}_{:,i}|$, for the ith column of $\bar{\mathbf{A}}$, as $\bar{\mathbf{A}}_{:,i} = \bar{\mathbf{A}}_{i,:}$ and $\bar{\mathbf{A}}_{:,i}$ encodes the nodes over which the information aggregation will be executed for node v_i (together with the corresponding weights). For example, if $\bar{\mathbf{A}}_{i,:}$ takes significantly higher values for the nodes with the same sensitive attributes to node v_i compared to the nodes with different sensitive attributes, this implies a high correlation between $\mathbf{s}$ and $\bar{\mathbf{A}}_{i,:}$. This correlation can also be reflected by $|\mathbf{s}^\top\bar{\mathbf{A}}_{i,:}|$, as it takes a higher value for the described case compared to the case where $\bar{\mathbf{A}}_{:,i}$ values are uniform for different sensitive attributes (specifically for $s_i = -1$ and $s_i = 1$).

Overall, we aim at minimizing the *total correlation* [30] between $\bar{\mathbf{A}}$ and $\mathbf{s}$, which we denote as $\rho := \|\mathbf{s}^\top\bar{\mathbf{A}}\|_2$. Notice that $\rho = \rho(\tilde{\mathbf{h}})$ because $\bar{\mathbf{A}} = \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top$, hence we can search over filter frequency responses to reduce graph-induced bias. This filter design problem is the subject we deal with next.

V. Fair Graph Filter Designs

A. Direct Optimization of ρ

Here we describe our convex optimization framework for fairness-aware optimal graph filter design. The idea is to formulate the following optimization problem to reduce the bias metric $\rho = \|\mathbf{s}^\top\bar{\mathbf{A}}\|_2$ via the employment of a graph filter with frequency response $\tilde{\mathbf{h}}$:

$$\begin{aligned}\tilde{\mathbf{h}}^{\text{f}} := \ &\underset{\tilde{\mathbf{h}}}{\text{argmin}} \ \ \rho(\tilde{\mathbf{h}})\\ &\text{s. to } \rho(\tilde{\mathbf{h}}) = \|\mathbf{s}^\top\mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top\|_2,\\ &\sum_{i=1}^{N}\tilde{h}_i \geq N\tau,\\ &0 \leq \tilde{h}_i \leq 1, \forall i \in \{1,\ldots,N\}.\end{aligned} \tag{5}$$

While we have discussed the criterion at length, the constraints deserve justification. Here, τ is a hyperparameter to control the amount of filtered information. It is important to emphasize that ρ can be minimized by setting $\tilde{\mathbf{h}}^{\text{f}} = \mathbf{0}$ ($\tau = 0$), which is equivalent to filtering out all information. This trivial solution is fair but naturally undesirable, because it sacrifices all utility. At the other extreme, if $\tau = 1$, all filter coefficients become 1, i.e., $\tilde{\mathbf{h}}^{\text{f}} = \mathbf{1}$, which would mean that we preserve all information without any fairness consideration. As we argued in Section IV-A, there needs to be a trade-off between utility and fairness. This trade-off can be empirically adjusted via the design parameter τ. Furthermore, the entries of $\tilde{\mathbf{h}}^{\text{f}}$ are constrained to not exceed 1. The spectrum of the input graph signal does not change for those frequencies λ_i, where $\tilde{h}_i = 1$. Thus, this constraint is utilized to preserve information in the frequencies that do not propagate bias as dictated by ρ. Overall, this choice is motivated by utility considerations in the downstream tasks. Note that the formulation in (5) is convex for the specified constraints; thus, it can be solved to global optimality using off-the-shelf methods.

Remark 1 (Spectral-domain design and eigendecomposition): The advocated graph spectral-domain design of the bias mitigating filter necessitates computing an eigendecomposition of the normalized Laplacian $\mathbf{L}$ prior to optimization. This $O(N^3)$ step can certainly challenge the applicability of the proposed approach when it comes to learning over large-scale graphs. This limitation nonwithstanding, our experimental results demonstrate this framework can comfortably handle network datasets with several thousands of nodes. Follow-up work on eigendecomposition-free filter designs in the vertex domains is certainly of interest; see also the related discussion preceding Remark 2.

B. Linear Programming With Closed-Form Solution

The formulation in (5) involves the optimization of N variables, which incurs high complexity for large graphs. To sidestep this potential computational bottleneck, we derive a surrogate cost that is amenable to efficient minimization. Specifically, we first conduct a bias analysis and upper bound the bias metric ρ. We then show that minimization of the upper bound results in an LP, whose solution is a filter with frequency response $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$. Remarkably, the solution to the LP attains a closed-form solution, which sidesteps the need for iterative solvers, and hence brings computational savings compared to (5). Together with this improved computational efficiency, once more, $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$ can effectively "filter out" the sensitive information from the bias-amplifying graph connectivity.

First, Proposition 1 reveals the sources of bias and provides an upper bound on the total correlation between $\mathbf{s}$ and $\bar{\mathbf{A}}$.

Proposition 1: Consider filtering signals using a graph filter with frequency response $\tilde{\mathbf{h}}$ prior to aggregation using $\hat{\mathbf{A}}$, and let $\bar{\mathbf{A}} := \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\mathrm{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top$. Then, $\rho := \|\mathbf{s}^\top \bar{\mathbf{A}}\|_2$ can be upper bounded by

$$\rho \leq \sqrt{N} \sum_{i=1}^{N} |\tilde{s}_i||(1-\lambda_i)||\tilde{h}_i|. \tag{6}$$

Proof: Leveraging the definitions of ρ and the effective aggregation operator $\bar{\mathbf{A}}$, we have $\bar{\mathbf{A}} := \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\mathrm{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top$:

$$\begin{aligned}\rho &= \|\mathbf{s}^\top \bar{\mathbf{A}}\|_2 \\ &= \|\mathbf{s}^\top \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\mathrm{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top\|_2. \end{aligned} \tag{7}$$

Furthermore, (7) can be reformulated based on the definition of GFT for the sensitive attribute signal $\mathbf{s}$:

$$\rho = \|\tilde{\mathbf{s}}^\top (\mathbf{I}_N - \mathbf{\Lambda})\mathrm{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top\|_2. \tag{8}$$

By utilizing the norm inequality, ρ can be upper bounded:

$$\begin{aligned}\rho &\leq \|\tilde{\mathbf{s}}^\top (\mathbf{I}_N - \mathbf{\Lambda})\mathrm{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top\|_1 \\ &= \sum_{j=1}^{N} |\sum_{i=1}^{N} \tilde{s}_i(1-\lambda_i)\tilde{h}_i v_{ji}|. \end{aligned} \tag{9}$$

Based on the triangle inequality, the following inequality can further be derived:

$$\begin{aligned}\rho &\leq \sum_{j=1}^{N}\sum_{i=1}^{N} |\tilde{s}_i(1-\lambda_i)\tilde{h}_i v_{ji}| \\ &\leq \sum_{j=1}^{N}\sum_{i=1}^{N} |\tilde{s}_i||(1-\lambda_i)||\tilde{h}_i||v_{ji}| \\ &= \sum_{i=1}^{N} |\tilde{s}_i||(1-\lambda_i)||\tilde{h}_i| \sum_{j=1}^{N} |v_{ji}| \\ &= \sum_{i=1}^{N} |\tilde{s}_i||(1-\lambda_i)||\tilde{h}_i|\|\mathbf{V}_{:,i}\|_1 \end{aligned} \tag{10}$$

Moreover, the relation between the ℓ_1 and ℓ_2-norms of a vector $\mathbf{a} \in \mathbb{R}^N$ can be written as $\|a\|_1 \leq \sqrt{N}\|a\|_2$, based on which it follows that:

$$\begin{aligned}\rho &\leq \sum_{i=1}^{N} |\tilde{s}_i||(1-\lambda_i)||\tilde{h}_i|\|\mathbf{V}_{:,i}\|_1 \\ &\leq \sqrt{N}\sum_{i=1}^{N} |\tilde{s}_i||(1-\lambda_i)||\tilde{h}_i|\|\mathbf{V}_{:,i}\|_2 \\ &= \sqrt{N}\sum_{i=1}^{N} |\tilde{s}_i||(1-\lambda_i)||\tilde{h}_i|, \end{aligned} \tag{11}$$

where the last equality holds because the eigenvectors of $\mathbf{L}$ are orthonormal. □

Proposition 1 shows that the linear correlation between the effective graph topology and the sensitive attributes is a function of $\sum_{i=1}^{N} |\tilde{s}_i||(1-\lambda_i)||\tilde{h}_i|$. Therefore, we can design a "matched" graph filter to reduce this term and hence the bias. Define $m_i := |\tilde{s}_i||(1-\lambda_i)|$, for all $i = 1, \ldots, N$, and let $\mathbf{m} \in \mathbb{R}^N$ be the vector whose ith component is m_i. Then, the following LP problem can be formulated for the design of an optimal fair graph filter:

$$\begin{aligned}\tilde{\mathbf{h}}^{f}_{cf} := \ & \underset{\tilde{\mathbf{h}}}{\mathrm{argmin}} \ \ \mathbf{m}^\top \tilde{\mathbf{h}} \\ \text{s. to } & \sum_{i=1}^{N} \tilde{h}_i \geq N\tau, \\ & 0 \leq \tilde{h}_i \leq 1, \forall i \in \{1, \ldots, N\}. \end{aligned} \tag{12}$$

The same set of constraints as in (5) are employed here. Let $\boldsymbol{\alpha} = \mathrm{argsort}(-\mathbf{m})$ be the vector containing the indices of the elements in $\mathbf{m}$ sorted in descending order. The closed-form solution for this LP problem can be obtained as:

$$(\tilde{h}^{\mathrm{f}}_{cf})_{\alpha_i} = \left[1 - \left[N(1-\tau) - \sum_{j=1}^{i-1}\left(1 - (\tilde{h}^{\mathrm{f}}_{cf})_{\alpha_j}\right)\right]_+\right]_+, \tag{13}$$

where $[x]_+ := \max(0, x)$ is a projection operator onto the non-negative reals.

Proof (sketch): It always holds that $m_i \geq 0$ and $\tilde{h}_i \geq 0$, for all $i = 1, \ldots, N$, due to the definition of $\mathbf{m}$ and the box constraints on each of the $\tilde{h}_i$. Therefore, the cost function is always non-negative, i.e., $\mathbf{m}^\top \tilde{\mathbf{h}} \geq 0$, and the equality is achieved when $\tilde{\mathbf{h}} = \mathbf{0}$. However, such a solution does not satisfy the constraint that lower bounds the sum of elements in $\tilde{\mathbf{h}}$ (unless when $\tau = 0$, but as discussed in Section V-A, this case is of no practical interest). The conclusion is that the optimal solution is attained on the boundary of the feasible set, where $\tilde{h}_i$ takes the smallest possible values for the largest entries of $\mathbf{m}$, as long as the first constraint holds. Specifically, the optimal $\tilde{\mathbf{h}}$ has null entries (or entries that are smaller than 1) in the indices where vector $\mathbf{m}$ takes the largest values as long as the filtering budget (imposed by the first constraint) is not exhausted, which provides the recursive solution in (13). □

For the budget prescribed by τ, the recursive definition of the filter's frequency response in (13) specifies the optimal solution of the LP design in (12).

C. Polynomial Graph Convolutional Filter

The LP-based filter design in the previous section admits a closed-form solution, and accordingly, it offers computational savings relative to (5), since solving the latter necessitates an iterative procedure. Here, instead, we adopt a polynomial graph filter parameterization [22], which offers an explicit handle on the number of optimization variables. This way, the number of variables decouples from (and can be markedly smaller than) the size of $\mathcal{G}$.

Polynomial graph convolutional filters are often the operators of choice in several SP and ML tasks due to their parameter sharing property, locality and linear computational complexity. When these filters are used in GNNs, the parameter sharing property allows them to learn complex relations within graphs (including large-scale ones) based on a limited number of training samples. The locality property implies they can be implemented in a distributed fashion, solely via exchanges of information with neighboring nodes in the graph. Finally, their linear computational complexity aids scalability [22].

In the frequency domain, as also mentioned in Section III-A, the graph convolutional filter's response is given by $\tilde{\mathbf{h}} := \boldsymbol{\Psi}\mathbf{h}$, for the $N \times L$ Vandermonde matrix $\boldsymbol{\Psi}$, where $\Psi_{ij} := (1 - \Lambda_{ii})^{j-1}$ [49]. Based on this parameterization, the optimization problem in (5) can be reformulated as:

$$\begin{aligned} \mathbf{h}^{\text{f}} := \; & \underset{\mathbf{h}}{\operatorname{argmin}} \;\; \rho(\mathbf{h}) \\ & \text{s. to } \rho(\mathbf{h}) = \|\mathbf{s}^{\top}\mathbf{V}(\mathbf{I}_N - \boldsymbol{\Lambda})\operatorname{diag}(\boldsymbol{\Psi}\mathbf{h})\mathbf{V}^{\top}\|_2, \\ & \qquad \sum_{i=1}^{N} (\boldsymbol{\Psi}\mathbf{h})_i \geq N\tau, \\ & \qquad 0 \leq (\boldsymbol{\Psi}\mathbf{h})_i \leq 1, \forall i \in \{1, \ldots, N\}. \end{aligned} \tag{14}$$

The number of optimization variables is L, regardless of the number of nodes in $\mathcal{G}$. By selecting a filter order that satisfies $L \ll N$, this approach can be a better fit for large graphs. Meanwhile, the fairness improvement it provides may be limited when compared to our previous designs, as its degrees of freedom are purposedly reduced. Another salient feature of the polynomial graph filter (2) obtained by solving (14) is that it can be *directly implemented* in the vertex domain via (distributed) information exchanges among neighbors. Polynomial parameterizations of the filters $\tilde{\mathbf{h}}^f$ and $\tilde{\mathbf{h}}^f_{cf}$ can be obtained as well, because they are jointly diagonalizable with $\hat{\mathbf{A}}$ by construction [51, Prop. 1]. However, this requires extra computation to interpolate the designed frequency responses and will likely necessitate a high value of L.

Note that ρ can also be optimized in the vertex domain by minimizing $\rho = \|\mathbf{s}^{\top}\bar{\mathbf{A}}\|_2 = \|\mathbf{s}^{\top}\hat{\mathbf{A}}\mathbf{H}\|_2 = \|\sum_{l=0}^{L-1} h_l \mathbf{s}^{\top}\hat{\mathbf{A}}^{l+1}\|_2$ with respect to the graph filter coefficients $\mathbf{h}$. This way, one eliminates the need to calculate eigenvectors and eigenvalues of the graph Laplacian (cf. Remark 1). However, the design of constraints for this formulation is less intuitive than in the frequency domain and becomes non-trivial. We leave this interesting endeavor as a future research direction.

Remark 2 (Flexible use of the proposed filter designs): The designed fair filters $\tilde{\mathbf{h}}^{\text{f}}$, $\mathbf{h}^{\text{f}}$, and $\tilde{\mathbf{h}}^{\text{f}}_{cf}$ can be employed in a flexible way to mitigate bias for different graph-based learning algorithms. They can be applied to the graph signals that are input to or output from the learning algorithms. Models designed for attributed graphs generally utilize the information from both the nodal features and graph topology [24]. Thus, the proposed filters can be applied to the nodal features before they are fed to the learning pipeline in order to prevent the amplification of bias due to the graph connectivity. Alternatively, for any algorithm that outputs a graph signal (e.g., node labels in node classification), $\tilde{\mathbf{h}}^{\text{f}}$, $\mathbf{h}^{\text{f}}$, and $\tilde{\mathbf{h}}^{\text{f}}_{cf}$ can be employed on the output graph signal as fairness-aware post-processing operators. Overall, the impact of the proposed fair filter designs can permeate several GNN-based learning frameworks in a versatile manner.

D. Discussion

We have proposed three novel designs with complementary strengths to mitigate bias in the network topology via graph filtering. Each design has certain advantages over the others when it comes to manipulating the effect that input graph structures and sensitive attributes have on learned representations. In the first design, a fair graph filter $\tilde{\mathbf{h}}^{\text{f}}$, is obtained by solving a convex optimization problem that directly minimizes the bias measure ρ. Compared to $\tilde{\mathbf{h}}^{\text{f}}_{cf}$ whose design is based on an upper bound on ρ, $\tilde{\mathbf{h}}^{\text{f}}$ is expected to yield better bias mitigation performance, especially when the bound gets looser for the input graph. Moreover, as $\tilde{\mathbf{h}}^{\text{f}}$ has higher degrees of freedom than the polynomial filter $\mathbf{h}^{\text{f}}$, again its application is expected to decrease ρ in a more effective way. On the other hand, both $\tilde{\mathbf{h}}^{\text{f}}_{cf}$ and $\mathbf{h}^{\text{f}}$ provide computationally more efficient bias mitigation solutions than $\tilde{\mathbf{h}}^{\text{f}}$. Furthermore, while $\tilde{\mathbf{h}}^{\text{f}}_{cf}$ is given in closed form and thus eliminates the need for iterative solvers, the number of optimization variables in the problem defining $\mathbf{h}^{\text{f}}$ is independent of the input graph size (the complexity of the sorting operation in the recursive computation of $\tilde{\mathbf{h}}^{\text{f}}_{cf}$ still grows with N). Granted, the number of constraints in (14) does depend on N, and that is why a full-blown vertex domain formulation is still of interest; see the discussion preceding Remark 2. All in all, both $\tilde{\mathbf{h}}^{\text{f}}_{cf}$ and $\mathbf{h}^{\text{f}}$ can provide the most efficient solution based on the input graph properties.

Overall, all our proposed fairness-aware graph filter designs can be employed in a flexible and efficient manner in several graph-based ML frameworks. For example, within GNN structures, these filters can be utilized as *pre-trained* bias mitigation operators before each GNN layer, e.g., see Fig. 1. It is important to emphasize that the employment of these filters as bias mitigation sub-layers within NNs does not modify the training process, unlike the majority of existing approaches that utilize fairness-aware regularizers and constraints [4], [5], [9], [12], [15], [34]. Therefore, our filters can lead to more stable training compared to these strategies, especially adversarial regularization-based ones that are known to suffer from instability issues [25]. Moreover, the proposed filters need to be computed only once for a

TABLE I
DATASET STATISTICS

Dataset	$\lvert\mathcal{S}_{-1}\rvert$	$\lvert\mathcal{S}_{1}\rvert$	$\lvert\mathcal{Y}_{-1}\rvert$	$\lvert\mathcal{Y}_{1}\rvert$	$\lvert\mathcal{E}\rvert$
Pokec-z	4851	2808	3856	3803	29476
Pokec-n	4040	2145	3432	2753	21844

given $\mathcal{G}$, after which they can be utilized for various tasks on said graph.

VI. Experimental Results

A. Dataset and Experimental Setup

Datasets: The performance of the proposed fair filter designs is evaluated on the node classification task over real-world social networks Pokec-z and Pokec-n. Pokec-z and Pokec-n are the sampled versions of the 2012 Pokec network [55], which is a Facebook-like social network in Slovakia [9]. The region of the users is utilized as the sensitive attribute, where the users are from two major regions. Labels for the node classification task are the binarized working field of the users. Statistics for the utilized datasets are presented 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: Accuracy is adopted as the utility metric of node classification. For fairness assessment, two quantitative measures of group fairness metrics are reported, 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 represents the ground truth label, and $\hat{y}$ is the predicted label. Here, statistical parity is a measure for the independence of positive rate from the sensitive attribute, and equal opportunity signifies the level of the independence of true positive rate from the sensitive attribute. Lower values for Δ_{SP} and Δ_{EO} indicate better fairness performance [9] and are more desirable.

Implementation details: We evaluate the proposed filter designs in two different environments. First, they are employed as bias mitigation sub-layers to filter the input representations to GNN layers in a two-layer graph convolutional network (GCN) [24]; see also Fig. 1. The GCN model is trained for node classification by employing the negative log-likelihood function as the objective. For this setting, the training set consists of 40% of the nodes, while the remaining nodes are evenly split to create validation and test sets. The hyperparameter τ is selected via grid search among the values $\{0.0003, 0.0004, 0.0005, 0.0006\}$ for the filters $\tilde{\mathbf{h}}^{\mathrm{f}}$ and $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$. Specifically, for $\tilde{\mathbf{h}}^{\mathrm{f}}$, τ is chosen to be 0.0005 and 0.0003 on Pokec-z and Pokec-n, respectively, while it equals 0.0004 for $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$ on both datasets. Moreover, for the proposed polynomial filter, L is selected as 40 and 50 on datasets Pokec-z and Pokec-n, respectively, based on a grid search among the values $\{30, 40, 50\}$. To alleviate the hyperparameter tuning step for $\mathbf{h}^{\mathrm{f}}$, $\tau = 0.0004$ is directly utilized on both datasets without any fine-tuning.

Second, to illustrate the use of the fair filters as post-processing operators, we use them to filter the predicted nodal labels computed by the classification algorithm presented in [47]. In the filtered signal, the components that are larger than a threshold are assigned to the first class, while the others are assigned to the second class. Note that this threshold is selected to be 0 for labels -1 and 1 in the experiments, however it can be adaptively chosen based on the input graph. In this second setting, 40% of the nodes are used to train the model and the remaining ones contribute to the test set. The hyperparameter tuning process is kept the same as in the case where the filters are employed as pre-processing operators. For $\tilde{\mathbf{h}}^{\mathrm{f}}$, τ is chosen to be 0.0004 on both datasets, while it equals to 0.0004 and 0.0006 for $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$ on Pokec-z and Pokec-n, respectively. For the polynomial filter, L is selected again as 40 and 50 on datasets Pokec-z and Pokec-n.

For all experiments, results are obtained for five random data splits, and their average along with the standard deviations are reported in the tables that follow. Further implementation details can be found in the publicly available code shared as supplementary material to this paper, which can be used to generate all results reported in this section.

Baselines: Fairness-aware baselines in the experiments include adversarial regularization [9], EDITS [11], and FairDrop [54]. Adversarial regularization is a widely utilized fairness enhancement strategy, where an adversary is trained to predict the sensitive attributes. For adversarial regularization, the multiplier of the regularizer is tuned via a grid search among the values $\{0.1, 1, 10, 100, 1000\}$ (the multiplier of classification loss is assigned to be 1). Furthermore, EDITS [11] is a model-agnostic debiasing framework that mitigates the bias in attributed networks before they are fed into any GNN. Specifically, it creates debiased versions of the nodal attributes and the graph structure, which are then input to the GCN network used here for node classification. For EDITS, the threshold proportion is tuned among the values $\{0.015, 0.02, 0.06, 0.29\}$, where these values are the optimized thresholds for other datasets used in [11]. Finally, FairDrop [54] proposes a biased edge dropout strategy for a more balanced graph topology in terms of the edges connecting different (and the same) sensitive groups. The hyperparameter δ in the FairDrop algorithm is tuned among the values $\{0.7, 0.8, 0.9\}$.

B. Results

Comparative results for the proposed fairness-aware graph filters, $\tilde{\mathbf{h}}^{\mathrm{f}}$, $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$, and $\mathbf{h}^{\mathrm{f}}$ are presented in Table II, for the case where they are utilized as bias mitigation layers. The natural baseline for the proposed strategies is to employ the GNN model without any fairness-aware operations, where this scheme is denoted by "GNN" in Table II. Moreover, "Adversarial", "EDITS", and "FairDrop" in Table II stand for the adoption of adversarial regularization in training [9], and state-of-the-art fairness-aware baselines EDITS [11], and FairDrop [54], respectively.

The results in Table II demonstrate that all of the proposed filter designs improve upon the naive GNN baseline in terms of both fairness metrics, while also providing similar utility. Specifically, the proposed strategies achieve 30% to 90% improvement in all fairness measures for every dataset compared to GNN. The results further demonstrate the superior fairness

TABLE II
PROPOSED FILTERS AS BIAS MITIGATION LAYERS IN A GNN MODEL

	Pokec-z			Pokec-n		
	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)
GNN	**66.52** ± 0.27	6.79 ± 2.45	7.26 ± 3.29	64.96 ± 0.19	6.79 ± 2.45	7.26 ± 3.29
Adversarial	64.26 ± 1.79	4.85 ± 2.16	5.99 ± 2.71	64.22 ± 0.71	4.34 ± 3.87	3.84 ± 2.71
EDITS [11]	62.67 ± 2.64	3.17 ± 2.49	4.54 ± 2.99	62.67 ± 0.51	4.40 ± 2.41	5.38 ± 1.92
FairDrop [54]	66.79 ± 0.65	9.11 ± 1.89	8.35 ± 3.81	64.33 ± 0.44	4.46 ± 1.67	5.02 ± 1.84
$\tilde{\mathbf{h}}^{\mathrm{f}}$+ GNN	66.05 ± 0.30	**1.08** ± 1.20	2.20 ± 2.06	**65.07** ± 0.21	**2.12** ± 1.01	**2.42** ± 1.96
$\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$+ GNN	66.34 ± 0.27	1.23 ± 1.43	**2.15** ± 1.96	**65.05** ± 0.21	**2.13** ± 0.93	**2.39** ± 1.78
$\mathbf{h}^{\mathrm{f}}$+ GNN	66.32 ± 0.27	3.36 ± 1.99	4.21 ± 2.43	**65.07** ± 0.21	4.39 ± 2.01	5.13 ± 2.00

The bold values indicate best results for the corresponding metric.

TABLE III
TOTAL PEARSON CORRELATION COEFFICIENTS [30] BETWEEN REPRESENTATIONS AND SENSITIVE ATTRIBUTES BEFORE/AFTER $\tilde{\mathbf{h}}^f$

	Pokec-z		Pokec-n	
	Before $\tilde{\mathbf{h}}^{fair}$	After $\tilde{\mathbf{h}}^{fair}$	Before $\tilde{\mathbf{h}}^{fair}$	After $\tilde{\mathbf{h}}^{fair}$
1st layer	4.27	1.98	4.60	1.94
2nd layer	4.25 ± 0.03	2.96 ± 0.01	3.12 ± 0.05	2.21 ± 0.02

performance of $\tilde{\mathbf{h}}^{\mathrm{f}}$ over the polynomial filter $\mathbf{h}^{\mathrm{f}}$, which is expected, as $\tilde{\mathbf{h}}^{\mathrm{f}}$ can better optimize our bias metric ρ with a higher number of degrees of freedom compared to $\mathbf{h}^{\mathrm{f}}$. Moreover, it is observed that our designs, $\tilde{\mathbf{h}}^{\mathrm{f}}$ and $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$, lead to similar fairness improvements, which signifies that the derived upper bound in (11) is a successful surrogate bias measure for the Pokec graphs.

The results in Table II also show that $\tilde{\mathbf{h}}^{\mathrm{f}}$ and $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$ always achieve better fairness performance together with similar/better utility, compared to other fairness-aware baselines, namely Adversarial [9], EDITS [11] and FairDrop [54]. While the polynomial filter, $\mathbf{h}^{\mathrm{f}}$ generally leads to a similar fairness improvement compared to other fairness-aware baselines, this fairness performance is typically accompanied by a better utility for $\mathbf{h}^{\mathrm{f}}$. Furthermore, it can be observed that the employment of the novel filters generally leads to the lowest standard deviation values, and therefore enhances the stability of the results. The improved utility provided by our filter designs compared to the GNN baseline on Pokec-n might seem counter-intuitive, due to the expected fairness-utility trade-off. Here, the higher classification accuracy can be attributed to the denser structure of the effective graph operator $\bar{\mathbf{A}}$ used for message passing, when compared to the sparser graph adjacency matrix $\mathbf{A}$ used in the fairness-agnostic GNN. Denser connections can lead to more powerful node representations (depending on the data), however at the cost of higher computational complexity for the message passing operations. Overall, the results corroborate the efficacy of the proposed filter designs design in mitigating bias, while also providing similar utility measures compared to the state-of-the-art fairness-aware baselines.

Fairness performance in Table II is reported in terms of commonly utilized group fairness measures; namely, statistical parity and equal opportunity, same as prior works [9], [11], [54]. In Table III, we also provide the total correlation values between the sensitive attributes and representations that are input to or output from the designed filter $\tilde{\mathbf{h}}^{\mathrm{f}}$. With reference to the two-layer GNN architecture in Fig. 1 that is used for this experiment, in the first row of Table III we report $\|\mathbf{s}^\top\mathbf{X}\|_1$ (before $\tilde{\mathbf{h}}^{\mathrm{f}}$) and $\|\mathbf{s}^\top\bar{\mathbf{X}}\|_1$ (after $\tilde{\mathbf{h}}^{\mathrm{f}}$). Likewise, in the second row, we report the total correlations $\|\mathbf{s}^\top\mathbf{H}_1\|_1$ (before $\tilde{\mathbf{h}}^{\mathrm{f}}$) and $\|\mathbf{s}^\top\bar{\mathbf{H}}_1\|_1$ (after $\tilde{\mathbf{h}}^{\mathrm{f}}$). Overall, the results demonstrate that $\tilde{\mathbf{h}}^{\mathrm{f}}$ can significantly reduce the correlation that is expected to lead to intrinsic bias, which is also reflected in the improved Δ_{SP} and Δ_{EO} values in Table II. This correlation reduction is observed at both stages in this two-layer GCN and for both datasets. Notice that $\|\mathbf{s}^\top\mathbf{H}_1\|_1 > \|\mathbf{s}^\top\bar{\mathbf{X}}\|_1$ because the GCN layer (mapping $\bar{\mathbf{X}}$ to $\mathbf{H}_1$) aggregates information using $\hat{\mathbf{A}}$, and the latter is highly correlated with $\mathbf{s}$ as discussed in Section IV-B. Furthermore, by comparing the first and second rows in Table III it is observed that the correlation reduction is more pronounced before any GNN layer is used to process the data. Since the representations output by a GNN layer are learned to maximize utility, this phenomenon is an expected result of the corresponding fairness-utility tradeoff.

We also provide experimental results herein, whereby the proposed fair filters are employed as post-processing operators on the predicted labels (a graph signal) of a node classification algorithm. For this setting, the classification results are obtained via the algorithm presented in [47], and we subsequently filter these predicted labels to debiase them. The results are presented in Table IV, which exhibit similar tendencies as those in Table II. Overall, our experiments confirm the efficacy of the proposed graph filters in improving fairness measures and also for the setting where they are employed as post-processing operators. In addition, similar to the findings of Table II, better fairness measures are typically accompanied by better stability and similar utility to the fairness-agnostic baseline [47].

Ablation study: To examine the effect of filter placement in the adopted two-layer GCN, we carry out an ablation study whose results are presented in Table V. Therein, "$\tilde{\mathbf{h}}^{\mathrm{f}}$+GNN" corresponds to an architecture where the designed filter is employed before both of the GCN layers; exactly as in Fig. 1. In the meantime, "$\tilde{\mathbf{h}}^{\mathrm{f}}$ before first layer" and "$\tilde{\mathbf{h}}^{\mathrm{f}}$ before second layer" denote architectures that utilize *a single filter* placed before the first layer only, or, the second layer only, respectively. Naturally, "GNN" corresponds to a baseline model which does not employ bias-mitigating filters. The key conclusion from this study is that using at least one filter, regardless of its placement

TABLE IV
ADAPTIVE FILTER, $\tilde{\mathbf{h}}^{\mathrm{F}}$ AS FAIRNESS-AWARE POST-PROCESSING OPERATOR

	Pokec-z			Pokec-n		
	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)
[47]	$\mathbf{64.83} \pm 0.54$	8.33 ± 2.64	9.38 ± 2.54	65.44 ± 0.42	6.27 ± 4.83	8.78 ± 6.18
[47] + $\tilde{\mathbf{h}}^{\mathrm{f}}$	64.44 ± 0.38	$\mathbf{1.58} \pm 1.01$	1.69 ± 1.41	65.75 ± 0.91	$\mathbf{2.11} \pm 2.19$	$\mathbf{3.48} \pm 3.44$
[47]+$\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$	64.70 ± 0.48	$\mathbf{1.57} \pm 1.24$	$\mathbf{1.55} \pm 1.21$	$\mathbf{65.80} \pm 0.86$	2.27 ± 2.14	$\mathbf{3.48} \pm 3.34$
[47] + $\mathbf{h}^{\mathrm{f}}$	64.62 ± 0.54	5.19 ± 2.39	6.09 ± 3.00	$\mathbf{65.78} \pm 0.93$	4.90 ± 3.28	6.44 ± 5.48

The bold values indicate best results for the corresponding metric.

TABLE V
ABLATION STUDY FOR THE EMPLOYMENT OF $\tilde{\mathbf{h}}^{\mathrm{F}}$ AS BIAS MITIGATION LAYERS

	Pokec-z			Pokec-n		
	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)
GNN	$\mathbf{66.52} \pm 0.27$	6.79 ± 2.45	7.26 ± 3.29	64.96 ± 0.19	6.79 ± 2.45	7.26 ± 3.29
$\tilde{\mathbf{h}}^{\mathrm{f}}$+ GNN	66.05 ± 0.30	$\mathbf{1.08} \pm 1.20$	2.20 ± 2.06	$\mathbf{65.07} \pm 0.21$	2.12 ± 1.01	$\mathbf{2.42} \pm 1.96$
$\tilde{\mathbf{h}}^{\mathrm{f}}$ before first layer	66.22 ± 0.23	1.33 ± 1.00	$\mathbf{1.98} \pm 2.18$	$\mathbf{65.05} \pm 0.31$	2.49 ± 1.08	2.55 ± 2.32
$\tilde{\mathbf{h}}^{\mathrm{f}}$ before second layer	66.17 ± 0.24	1.13 ± 1.26	2.06 ± 1.80	$\mathbf{65.10} \pm 0.18$	$\mathbf{2.05} \pm 1.09$	2.46 ± 1.91

The bold values indicate best results for the corresponding metric.

TABLE VI
SENSITIVITY ANALYSIS FOR THE HYPERPARAMETER τ IN $\tilde{\mathbf{h}}^{\mathrm{F}}$ AS A BIAS MITIGATION LAYER

	Pokec-z			Pokec-n		
	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)
GNN	66.52 ± 0.27	6.79 ± 2.45	7.26 ± 3.29	64.96 ± 0.19	6.79 ± 2.45	7.26 ± 3.29
$\tau = 0.0003$	66.33 ± 0.25	1.34 ± 1.39	2.26 ± 2.07	$\mathbf{65.07} \pm 0.21$	$\mathbf{2.12} \pm 1.01$	$\mathbf{2.42} \pm 1.96$
$\tau = 0.0004$	66.33 ± 0.22	1.15 ± 1.33	2.26 ± 1.75	65.04 ± 0.20	2.18 ± 0.95	2.47 ± 1.87
$\tau = 0.0005$	66.05 ± 0.30	$\mathbf{1.08} \pm 1.20$	$\mathbf{2.20} \pm 2.06$	$\mathbf{65.05} \pm 0.18$	2.41 ± 0.86	2.82 ± 1.68
$\tau = 0.0006$	$\mathbf{66.76} \pm 0.25$	1.49 ± 1.27	2.73 ± 2.27	64.97 ± 0.12	2.46 ± 0.66	2.93 ± 1.58

The bold values indicate best results for the corresponding metric.

TABLE VII
SENSITIVITY ANALYSIS FOR THE HYPERPARAMETER τ IN $\tilde{\mathbf{h}}^{\mathrm{F}}_{cf}$ AS A BIAS MITIGATION LAYER

	Pokec-z			Pokec-n		
	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)
GNN	$\mathbf{66.52} \pm 0.27$	6.79 ± 2.45	7.26 ± 3.29	64.96 ± 0.19	6.79 ± 2.45	7.26 ± 3.29
$\tau = 0.0003$	66.30 ± 0.24	1.34 ± 1.38	2.34 ± 2.07	$\mathbf{65.07} \pm 0.21$	$\mathbf{2.12} \pm 1.01$	$\mathbf{2.42} \pm 1.96$
$\tau = 0.0004$	66.34 ± 0.27	1.23 ± 1.43	$\mathbf{2.15} \pm 1.96$	$\mathbf{65.05} \pm 0.21$	$\mathbf{2.13} \pm 0.93$	$\mathbf{2.39} \pm 1.78$
$\tau = 0.0005$	66.34 ± 0.22	$\mathbf{1.19} \pm 1.36$	2.35 ± 1.74	64.99 ± 0.19	2.16 ± 0.82	$\mathbf{2.39} \pm 1.84$
$\tau = 0.0006$	66.19 ± 0.36	1.66 ± 1.10	2.64 ± 2.27	65.01 ± 0.10	2.28 ± 0.92	2.58 ± 1.83

The bold values indicate best results for the corresponding metric.

within the architecture, always helps improve fairness measures. Furthermore, if only one filter is used, we find that placing it deeper (meaning before the second layer) results in better/similar fairness measures compared to an earlier placement of the filter. Finally, results in Table V are inconclusive as to whether employing the filter before all layers is always the best strategy due to the high variances. Still, we find that employing the proposed filter before every layer achieves a similar fairness performance in the worst case compared to single filter placements. Thus, we suggest the use of filters in all layers for a simpler design.

Sensitivity analyses: For our designs $\tilde{\mathbf{h}}^{\mathrm{f}}$ and $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$, sensitivity analyses are presented in Tables VI and VII, respectively; in order to assess their sensitivity to their hyperparameter, τ, for the case where they are employed as bias mitigation layers. Note that Fig. 2 suggests that the number of frequencies where the magnitudes of $\tilde{\mathbf{s}}$ are markedly higher than $\tilde{\mathbf{y}}$ is around 3 for both datasets. Thus, the range of τ is chosen so that the total number of spectral components for which the filters' frequency response is approximately equal to 0 is less than 10. This way, we expect to improve markedly in terms of fairness without incurring a major degradation in utility. Overall, the results

TABLE VIII
SENSITIVITY ANALYSIS FOR THE HYPERPARAMETER τ IN $\tilde{\mathbf{h}}^{\mathrm{F}}$ AS A POST-PROCESSOR

	Pokec-z			Pokec-n		
	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)
[47]	$\mathbf{64.83} \pm 0.54$	8.33 ± 2.64	9.38 ± 2.54	65.44 ± 0.42	6.27 ± 4.83	8.78 ± 6.18
$\tau = 0.0003$	64.40 ± 0.33	1.67 ± 0.95	1.83 ± 1.30	65.48 ± 0.50	2.33 ± 2.27	3.85 ± 3.58
$\tau = 0.0004$	64.44 ± 0.38	$\mathbf{1.58} \pm 1.01$	$\mathbf{1.69} \pm 1.41$	$\mathbf{65.75} \pm 0.91$	$\mathbf{2.11} \pm 2.19$	$\mathbf{3.48} \pm 3.44$
$\tau = 0.0005$	64.45 ± 0.42	1.94 ± 0.81	2.28 ± 1.57	$\mathbf{65.78} \pm 0.91$	$\mathbf{2.14} \pm 2.20$	$\mathbf{3.50} \pm 3.47$
$\tau = 0.0006$	64.27 ± 0.48	2.01 ± 1.17	2.63 ± 1.85	65.81 ± 0.88	2.20 ± 2.18	$\mathbf{3.51} \pm 3.44$

The bold values indicate best results for the corresponding metric.

TABLE IX
SENSITIVITY ANALYSIS FOR THE HYPERPARAMETER τ IN $\tilde{\mathbf{h}}^{\mathrm{F}}_{cf}$ AS A POST-PROCESSOR

	Pokec-z			Pokec-n		
	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)	Accuracy (%)	Δ_{SP} (%)	Δ_{EO} (%)
[47]	$\mathbf{64.83} \pm 0.54$	8.33 ± 2.64	9.38 ± 2.54	65.44 ± 0.42	6.27 ± 4.83	8.78 ± 6.18
$\tau = 0.0003$	$\mathbf{64.80} \pm 0.39$	1.64 ± 0.82	1.76 ± 1.00	65.68 ± 0.33	2.61 ± 2.19	5.57 ± 2.91
$\tau = 0.0004$	64.70 ± 0.48	$\mathbf{1.57} \pm 1.24$	$\mathbf{1.55} \pm 1.21$	65.48 ± 0.50	2.33 ± 2.04	4.54 ± 3.00
$\tau = 0.0005$	64.35 ± 0.50	1.83 ± 1.14	2.37 ± 0.47	$\mathbf{65.79} \pm 0.90$	$\mathbf{2.24} \pm 2.14$	3.55 ± 3.40
$\tau = 0.0006$	64.07 ± 0.39	2.31 ± 1.70	3.09 ± 1.72	$\mathbf{65.80} \pm 0.86$	$\mathbf{2.27} \pm 2.14$	$\mathbf{3.48} \pm 3.34$

The bold values indicate best results for the corresponding metric.

demonstrate that the filters, $\tilde{\mathbf{h}}^{\mathrm{f}}$ and $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$, always lead to better fairness measures compared to the fairness-agnostic GNN baseline, within a broad range of hyperparameter choices. The sensitivity analyses for setting where the filters are used as post-processing operators are deferred to the Appendix, which lead to a similar conclusion.

C. On the Effective Network Operator

Based on (4), the effective network operator used in the learning process is defined to be $\bar{\mathbf{A}} := \mathbf{V}(\mathbf{I}_N - \mathbf{\Lambda})\mathrm{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top$. Therefore, employing the proposed graph filters can be interpreted as a modification to the original graph connectivity with the final aim of reducing the structural bias. For graphs encountered in various application domains, it is typically observed that the number of edges connecting the same sensitive groups, intra-edges, is significantly larger than the number of edges linking different sensitive groups, inter-edges, due to the homophily principle [9], [30]. Based on this observation, several studies have demonstrated that the imbalance between the number of intra and inter-edges is a major factor for the resulting algorithmic bias [30], [34], [54]. Motivated by this, here we visualize the intra- and inter-edges in a sub-graph extracted from the Pokec network and their distributional change in the effective network operator after applying the filter $\tilde{\mathbf{h}}^{\mathrm{f}}$. Specifically, Fig. 3 illustrates the intra- and inter-edges using the colors green and red, respectively, for the original subgraph and the modified effective graph structure after $\tilde{\mathbf{h}}^{\mathrm{f}}$ is employed. The figure reveals that the application of $\tilde{\mathbf{h}}^{\mathrm{f}}$ has a balancing effect in the number of intra- and inter-edges in the resulting graph structure, which can help visualize the bias mitigation mechanisms of the proposed strategies. Note that this balancing effect is also supported by comparing the total weights of intra- and inter-edges in the original network operator $\hat{\mathbf{A}}$ versus the effective one $\bar{\mathbf{A}}$ obtained when accounting for the filters. Specifically, in the original topology, the total weights of the intra- and inter-edges are 3102 and 165, respectively. On the other hand, the application of $\tilde{\mathbf{h}}^{\mathrm{f}}$ has a balancing effect resulting in 1824 and 1424 intra- and inter-edges, respectively.

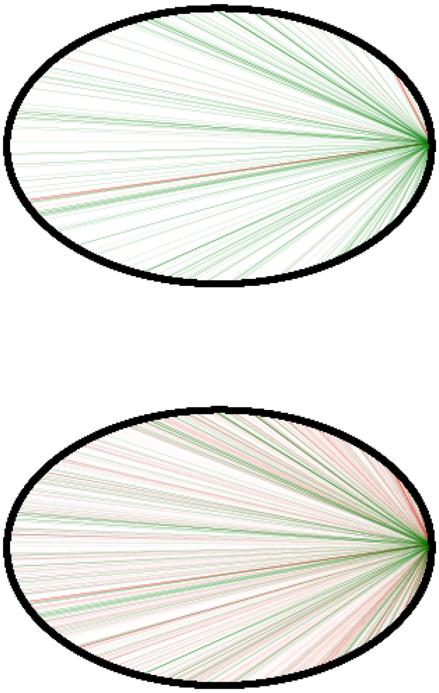

Fig. 3. For a sampled subgraph from the Pokec network, the distribution of the intra-edges (green) and inter-edges (red) in the effective network topology without (top)/ with (bottom) the application of $\tilde{\mathbf{h}}^{\mathrm{f}}$.

VII. Conclusion

In this study, we put forth three novel graph filter designs with the goal of mitigating bias stemming from the graph topology. Specifically, we first introduce a bias metric, ρ, that is applicable to unsupervised learning settings and measures the correlation between the connectivity pattern and sensitive attributes. Our first graph filter design, $\tilde{\mathbf{h}}^{\mathrm{f}}$, is obtained as a solution to a convex optimization problem that minimizes ρ. For a more efficient solution, we carry out a bias analysis and formulate an LP problem that targets the minimization of an upper bound on ρ. Remarkably, we show the LP attains a closed-form optimal solution for a fair graph filter $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$. Finally, we take a fair, polynomial graph convolution filter, $\mathbf{h}^{\mathrm{f}}$, into consideration, where the number of optimization variables in the corresponding design is independent of the input graph size. The proposed fairness-aware graph filters can be flexibly employed in various graph-based ML and SP algorithms at different stages of learning. Node classification experiments on real-world networks demonstrate that all of the proposed filter designs mitigate bias effectively. We observe they typically lead to better fairness measures when compared to other state-of-the-art fairness-aware baselines, and without notably sacrificing utility (i.e., classification accuracy).

This work opens up several exciting future directions. First, the proposed designs assume the existence of a single sensitive attribute, whereas considering multiple sensitive attributes in our designs would be certainly of interest. Second, this study focuses on the linear correlation between the graph structure and sensitive attributes as a bias measure, and extending our analysis to non-linear correlation metrics is another important future direction. Finally, robust adaptations of the proposed designs to accommodate several real-world challenges, including but not limited to missing sensitive attribute/graph structure information, and privacy constraints, are important components of our future research agenda.

Appendix
Further Sensitivity Analyses

The sensitivity analyses are further provided in Tables VIII and IX for the case where the proposed filters, $\tilde{\mathbf{h}}^{\mathrm{f}}$ and $\tilde{\mathbf{h}}^{\mathrm{f}}_{cf}$, are employed in the post-processing step. The results in these tables signify that the proposed strategies always improve the natural baseline in terms of fairness for a wide range of τ values also for their employment as post-processing operators.

References

[1] C. Agarwal, H. Lakkaraju, and M. Zitnik, “Towards a unified framework for fair and stable graph representation learning,” in *Proc. Thirty-Seventh Uncertainty Artif. Intell. Conf.*, 2021, pp. 2114–2124.

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

[3] A. Beutel, J. Chen, Z. Zhao, and E. H. Chi, “Data decisions and theoretical implications when adversarially learning fair representations,” in *Proc. Fairness, Accountability, Transparency Mach. Learn. Workshop*, 2017, pp. 1–5.

[4] A. Bose and W. Hamilton, “Compositional fairness constraints for graph embeddings,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 715–724.

[5] 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 Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2021, pp. 351–366.

[6] M. Buyl and T. D. Bie, “DeBayes: A Bayesian method for debiasing network embeddings,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1220–1229.

[7] 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.

[8] F. R. Chung and F. C. Graham, *Spectral Graph Theory*, vol. 92.Ann Arbor, MI, USA: Amer. Math. Soc., 1997.

[9] 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. Web Search Data Mining*, 2021, pp. 680–688.

[10] 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.*, vol. 37, no. 6, pp. 117–127, Nov. 2020.

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

[12] J. Fisher, A. Mittal, D. Palfrey, and C. Christodoulopoulos, “Debiasing knowledge graph embeddings,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 7332–7345.

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

[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, Nov. 2020.

[15] D. Guo, C. Wang, B. Wang, and H. Zha, “Learning fair representations via distance correlation minimization,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 16, 2022, doi: 10.1109/TNNLS.2022.3187165.

[16] W. L. Hamilton, “Graph representation learning,” *Synth. Lectures Artifical Intell. Mach. Learn.*, vol. 14, no. 3, pp. 1–159, 2020.

[17] 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.

[18] R. Hegselmann and U. Krause, “Opinion dynamics and bounded confidence models, analysis, and simulation,” *J. Artif. Societies Social Simul.*, vol. 5, no. 3, pp. 1–2, 2002.

[19] B. Hofstra, R. Corten, F. V. Tubergen, and N. B. Ellison, “Sources of segregation in social networks: A novel approach using facebook,” *Amer. Sociol. Rev.*, vol. 82, no. 3, pp. 625–656, 2017.

[20] K. Holstein, J. W. Vaughan, H. Daumé III, M. Dudik, and H. Wallach, “Improving fairness in machine learning systems: What do industry practitioners need?,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–16.

[21] E. Isufi, F. Gama, and A. Ribeiro, “EdgeNets: Edge varying graph neural networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7457–7473, Nov. 2022.

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

[23] 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, Jan. 2017.

[24] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[25] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, “On convergence and stability of GANs,” 2017, *arXiv:1705.07215*.

[26] E. D. Kolaczyk and G. Csárdi, *Statistical Analysis of Network Data With R*, vol. 65. Berlin, Germany: Springer, 2014.

[27] O. D. Kose and Y. Shen, “Demystifying and mitigating bias for node representation learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 17, 2023, doi: 10.1109/TNNLS.2023.3265370.

[28] O. D. Kose and Y. Shen, “Fast&fair: Training acceleration and bias mitigation for GNNs,” *Trans. Mach. Learn. Res.*, 2023. [Online]. Available: https://openreview.net/forum?id=nOk4XEB7Ke

[29] O. D. Kose, Y. Shen, and G. Mateos, “Fairness-aware graph filter design,” 2023, *arXiv:2303.11459*.

[30] O. D. Kose and Y. Shen, “Fair contrastive learning on graphs,” *IEEE Trans. Signal Process. over Netw.*, vol. 8, pp. 475–488, 2022.

[31] O. D. Kose and Y. Shen, “Fairness-aware selective sampling on attributed graphs,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5682–5686.

[32] E. Krasanakis and S. Papadopulos, “Graph neural network surrogates of fair graph filtering,” 2023, *arXiv:2303.08157*.

[33] C. Laclau, I. Redko, M. Choudhary, and C. Largeron, “All of the fairness for edge prediction with optimal transport,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1774–1782.

[34] P. Li, Y. Wang, H. Zhao, P. Hong, and H. Liu, “On dyadic fairness: Exploring and mitigating bias in graph connections,” in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–18.

[35] J. Ma, J. Deng, and Q. Mei, “Subgroup generalization and fairness of graph neural networks,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 1048–1061.

[36] Y. Ma, X. Liu, T. Zhao, Y. Liu, J. Tang, and N. Shah, “A unified view on graph neural networks as graph signal denoising,” in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 1202–1211.

[37] 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, Nov. 2017.

[38] 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, May 2019.

[39] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–35, 2021.

[40] J. Mei and J. M. Moura, “Signal processing on graphs: Estimating the structure of a graph,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5495–5499.

[41] 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, May 2018.

[42] S. Patterson and B. Bamieh, “Interaction-driven opinion dynamics in online social networks,” in *Proc. First Workshop Social Media Analytics*, 2010, pp. 98–105.

[43] D. Pessach and E. Shmueli, “A review on fairness in machine learning,” *ACM Comput. Surv.*, vol. 55, no. 3, pp. 1–44, 2022.

[44] T. A. Rahman, B. Surma, M. Backes, and Y. Zhang, “Fairwalk: Towards fair graph embedding,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3289–3295.

[45] 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, Feb. 2017.

[46] L. Ruiz, F. Gama, and A. Ribeiro, “Graph neural networks: Architectures, stability, and transferability,” *Proc. IEEE*, vol. 109, no. 5, pp. 660–682, May 2021.

[47] A. Sandryhaila and J. M. Moura, “Classification via regularization on graphs,” in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2013, pp. 495–498.

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

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

[50] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, “Interpolation of graph signals using shift-invariant graph filters,” in *Proc. IEEE 23rd Eur. Signal Process. Conf.*, 2015, pp. 210–214.

[51] S. Segarra, A. G. Marques, and A. Ribeiro, “Optimal graph-filter design and applications to distributed linear network operators,” *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.

[52] 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, May 2013.

[53] D. I. Shuman, P. Vandergheynst, and P. Frossard, “Chebyshev polynomial approximation for distributed signal processing,” in *Proc. IEEE Int. Conf. Distrib. Comput. Sensor Syst. Workshops*, 2011, pp. 1–8.

[54] I. Spinelli, S. Scardapane, A. Hussain, and A. Uncini, “FairDrop: Biased edge dropout for enhancing fairness in graph representation learning,” *IEEE Trans. Artif. Intell.*, vol. 3, no. 3, pp. 344–354, Jun. 2022.

[55] L. Takac and M. Zabovsky, “Data analysis in public social networks,” in *Proc. Int. Sci. Conf. Int. Workshop Present Day Trends Innovations*, 2012, pp. 1–6.

[56] Z. Zeng, R. Islam, K. N. Keya, J. Foulds, Y. Song, and S. Pan, “Fair representation learning for heterogeneous information networks,” in *Proc. Int. AAAI Conf. Web Social Media*, 2021, pp. 877–887.

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

[58] M. Zhu, X. Wang, C. Shi, H. Ji, and P. Cui, “Interpreting and unifying graph neural networks with an optimization framework,” in *Proc. Web Conf.*, 2021, pp. 1215–1226.

[59] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions,” in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 912–919.

O. Deniz Kose received the B.S. and M.S. degrees in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 2017 and 2020, respectively. She is currently a Ph.D. Student and Research Assistant with the Department of Electrical Engineering and Computer Science, University of California Irvine, Irvine, CA, USA. Her research interests include trustworthy machine learning, graph signal processing, and speech processing.

Gonzalo Mateos (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from Universidad de la República, Montevideo, Uruguay, in 2005 and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 2009 and 2011, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA, as well as an Asaro Biggar Family Fellow in data science. He is also the Associate Director for Research with the Goergen Institute for Data Science, University of Rochester. During 2013, he was a Visiting Scholar with the CS Department, Carnegie Mellon University, Pittsburgh, PA, USA. From 2004 to 2006, he was a Systems Engineer with Asea Brown Boveri, Montevideo, Uruguay. His research interests include the areas of statistical learning from complex data, network science, decentralized optimization, and graph signal processing.

Yanning Shen received the Ph.D. degree from the University of Minnesota, Minneapolis, MN, USA, in 2019. She is currently an Assistant Professor with the University of California, Irvine, CA, USA. She was a finalist for the Best Student Paper Award at the 2017 IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing and the 2017 Asilomar Conference on Signals, Systems, and Computers. She was selected as a Rising Star in EECS by Stanford University, Stanford, CA, USA, in 2017. She was the recipient of Microsoft Academic Grant Award for AI Research in 2021 and the Google Research Scholar Award in 2022. She is also an awardee of the MIT Technology Review 35 Innovators under 35 Asian Pacific in 2022. Her research interests include the areas of machine learning, network science, data science, and signal processing.