Balancing Fairness and Accuracy in Graph Learning via Fairness-Constrained Rewiring

Jason Wang JASONWANG1@COLLEGE.HARVARD.EDU
Lukas Fesser LUKAS_FESSER@G.HARVARD.EDU
Melanie Weber MWEBER@SEAS.HARVARD.EDU
Harvard University, School of Engineering and Applied Sciences

Editors: List of editors' names

Abstract

Algorithmic fairness aims to ensure the safe and responsible use of machine learning tools in applications across domains. In graph learning, several "fair rewiring" approaches have been proposed that perturb edges in the input graph to mitigate feature and relational bias. However, these approaches can lead to a decrease in accuracy in downstream tasks. On the other hand, classical rewiring approaches *improve* accuracy by mitigating over-smoothing and over-squashing effects induced by the graph's topology. In this work we show that those classical rewiring approaches reinforce existing topological biases and boost accuracy at the cost of fairness. We propose a novel fairness metric (*topological bias*) that allows for evaluating relational bias separately from feature bias. We then propose a fairness constraint that can be incorporated into classical rewiring techniques to mitigate topological bias. We show that the resulting *fairness-constrained rewiring* balances fairness and accuracy effectively in graph learning tasks.

Keywords: Graph Neural Networks, Algorithmic Fairness, Graph Rewiring, Fairness-Accuracy Trade-Off

1. Introduction

Graph Neural Networks (GNNs) have emerged as one of the most popular architectures for graph learning. The safe and responsible adoption of such tools in applications necessitates specialized fairness concepts for graph domains. Notions of algorithmic fairness have been previously studied on other domains, specifically for tabular (Feldman et al., 2015; Bird et al., 2020), language (Blodgett et al., 2020; Sun et al., 2019), and vision data (Gustafson et al., 2023; Luo et al., 2024), and fair predictions have been studied in the context of algorithmic decision making in areas such as credit scoring (Bono et al., 2021), hiring (Raghavan et al., 2020; Schumann et al., 2020), and legal proceedings (Sargent and Weber, 2021). One difficulty in defining analogous notions on graph domains stems from the non-independent and identically distributed nature of relational data (Ma et al., 2021). In addition, graph fairness metrics should account for both feature bias and relational bias, which can be challenging to capture in practice. Recently, several fairness notions have been proposed in the graph machine learning literature and leveraged for the design of bias mitigation strategies (Rahman et al., 2019; Kang et al., 2020; Agarwal et al., 2021; Kang et al., 2022). While these approaches have shown promise in mitigating biases, they can lead to a decrease in performance in downstream tasks.

Several of the proposed fairness approaches rely on graph rewiring, where edges in the input graph are perturbed to mitigate bias encoded in the features and the connectivity of the graph (Kang and Tong, 2022; Loveland et al., 2022). Rewiring as a preprocessing routine has also been studied more broadly in graph learning as a tool for mitigating over-smoothing and over-squashing effects that can impact accuracy in downstream tasks (Karhadkar et al., 2022; Topping et al., 2022). In this context, the interplay between these two perspectives on rewiring arises as an interesting question. We work towards addressing this question by investigating trade-offs in accuracy and fairness in rewiring and by introducing techniques to balance both effectively.

We begin by expanding the set of existing fairness metrics for graph learning by defining a notion of *topological bias* that captures relational bias induced by the graph's topology. Next we investigate trade-offs between fairness and accuracy in node-level graph learning tasks using topological bias and compare its utility with existing graph fairness metrics. We will see that while classical rewiring approaches can improve accuracy in downstream tasks, they do so at the cost of reinforcing existing topological biases in the learned node representations. To mitigate this shortcoming, we will introduce a *soft fairness constraint* based on our notion of topological bias that can be incorporated into existing rewiring approaches. The resulting *fairness-constrained rewiring* ensures that the performed edge perturbations improve accuracy in downstream tasks without amplifying topological biases.

1.1. Related Work

Recently, analogues of group, individual, counterfactual, and Rawlsian fairness were defined on graphs (Rahman et al., 2019; Kang et al., 2020; Agarwal et al., 2021; Kang et al., 2022). In addition, several graph-specific notions based on structural properties, such as node degrees and subgraphs, were introduced (Liu et al., 2023; Ma et al., 2021; Subramonian et al., 2024). Along with these fairness concepts, several strategies for combating bias in graph learning tasks have been proposed. This includes model interventions, notably proposals for fair message-passing (Jiang et al., 2022; Lin et al., 2024; Hoang et al., 2023; Tang et al., 2020), data augmentation (Wang et al., 2022), as well as structural constraints that are imposed during training (Li et al., 2022; Liu et al., 2021). While these techniques have shown promise, they require significant changes in the GNN's architecture and training procedure. A much simpler approach for mitigating biases is via preprocessing routines, specifically *rewiring* of the input graph. Here, the edges of the input graph are perturbed, such that underlying biases are reduced (Spinelli et al., 2021; Loveland et al., 2022). While these methods mitigate bias effectively, they can significantly reduce accuracy. Graph rewiring has also been studied in the context of mitigating over-squashing and over-smoothing effects with the goal of improving accuracy, resulting in a wide range of proposed rewiring approaches (Karhadkar et al., 2022; Topping et al., 2022; Fesser and Weber, 2023; Black et al., 2023; Rong et al., 2019). To the best of our knowledge, the impact of the general rewiring approaches on fairness has not been studied systematically.

1.2. Summary of Contributions

The main contributions of this study are as follows:

1. We define a principled fairness metric (*topological bias*) for graph learning that captures relational bias separately from feature bias.

2. We demonstrate that existing rewiring approaches either boost accuracy at the cost of fairness or vice versa, leading to a *fairness-accuracy trade-off*.
3. We propose *fairness-constrained rewiring*, which allows for improving accuracy in downstream task without reinforcing topological biases.[1]

2. Background and Notation

Throughout the paper we consider connected, undirected, and simple graphs $G = (V, E)$ with k-dimensional node attributes $X \in \mathbb{R}^{|V| \times k}$ and edges $E \subseteq V \times V$.

2.1. Message-Passing Graph Neural Networks

Most state-of-the-art Graph Neural Networks (GNNs) are based on the *message-passing* paradigm (Gori et al., 2005; Hamilton et al., 2017), learning a joint representation of a graph's relational structure and attributes via the following iterative procedure: Let $\mathbf{x}_v^l$ denote the representation of node v in layer l. The representation in the next layer is given by $\mathbf{x}_v^{l+1} = \phi_l \left(\bigoplus_{p \in \mathcal{N}_v \cup \{v\}} \psi_l \left(\mathbf{x}_p^l \right) \right)$, where node representations $\mathbf{x}_v^0$ are initialized by the node attributes in the input graph. The function ψ_l aggregates information from the neighbors of the node v (the *message*), the function ϕ_l computes the updated node representation. The specific form of both functions varies across architectures. In this work we consider three popular instances of message-passing GNNs: *Graph Convolutional Networks* (short: GCN) (Kipf and Welling, 2017), *Graph Isomorphism Networks* (short: GIN) (Xu et al., 2018) and *Graph Attention Networks* (short: GAT) (Veličković et al., 2018).

2.2. Graph Rewiring

While GNNs have shown great success across graph domains, they are not without challenges. Two types of instabilities can significantly decrease downstream performance:

Over-squashing and Over-smoothing. *Over-squashing* (Alon and Yahav, 2021) arises from bottlenecks in the information flow between distant nodes, which form as the number of layers increases. This can limit the GNN's ability to accurately encode long-range dependencies in the learned representations. *Over-smoothing* (Li et al., 2018) describes the effect that representations of dissimilar nodes become indistinguishable as the number of layers increases. Both are known to negatively affect accuracy in downstream tasks.

Rewiring. Graph rewiring describes a preprocessing routine that perturbs the edges of the input graph with the goal of mitigating over-squashing and/or over-smoothing to improve accuracy. A range of rewiring techniques has been proposed in the literature, many of which use classical graph characteristics to guide edge perturbations, including the spectrum of the Graph Laplacian (Karhadkar et al., 2022), discrete Ricci curvature (Topping et al., 2022; Nguyen et al., 2023; Fesser and Weber, 2023), effective resistance (Black et al., 2023), and edge density (Rong et al., 2019), among others. Tab. 1 summarizes the three rewiring techniques that we study here, which are the most popular representative of the three most common paradigms in graph rewiring: spectral, curvature-based, and probabilistic rewiring.

1. Our code and experiment scripts are open-sourced and accessible at https://github.com/Weber-GeoML/Rewiring_and_Fairness.

Approach	Over-smooth	Over-squash	Type	Complexity
FoSR (Karhadkar et al., 2022)	✗	✓	spectral	$O(n^2)$
BORF (Nguyen et al., 2023)	✓	✓	curvature-based	$O(m \cdot d_{\max}^3)$
DropEdge (Rong et al., 2019)	✓	✗	probabilistic	$O(m \cdot n^2)$

Table 1: Rewiring approaches for mitigating over-squashing and over-smoothing considered in this study. Here, n denotes the number of nodes, m the number of edges, and $d_{\max}$ the maximum node degree.

2.3. Algorithmic Fairness

Notions of fairness have long been studied in the machine learning literature. We briefly review *statstical parity*, a classical fairness notions that forms the basis of our subsequently proposed graph fairness metric. We note that throughout this paper we focus solely on *group fairness*, as opposed to individual fairness. Additional background on fairness metrics can be found in Apx. A. *Statistical parity* is one of the most widely used algorithmic fairness concepts; it states that the outcome should be independent of the protected attribute:

Definition 1 (Statistical Parity) *Statistical parity is achieved when* $\Pr\left[\hat{Y}=1|A=0\right] = \Pr\left[\hat{Y}=1|A=1\right]$. *In practice, this is difficult to satisfy perfectly, so we measure the statistical parity deviation instead:* $\left|\Pr\left[\hat{Y}=1|A=0\right] - \Pr\left[\hat{Y}=1|A=1\right]\right|$. *Smaller deviation means fairer (closer to satisfying the fairness constraint).*

For ease of notation, the formal definition is stated for a prediction task with binary outcomes ($\hat{Y} \in \{0,1\}$) with respect to a binary protected attribute ($A \in \{0,1\}$); an extension to the multi-class setting is straightforward.

3. Fairness in Graph Machine Learning

3.1. Topological Bias

A crucial challenge in adapting fairness notions to graph learning lies in the observation that biases can arise from feature information, as well as from the relational information encoded in the graph's topology (Fig. 1). A naive extension of the fairness metrics discussed above may allow for evaluating the first but not the second type of bias. For example, two nodes may have identical features, but disparate connections, which could change the resulting prediction. To address this shortcoming, we incorporate a measure of topological bias into classical fairness metrics, by treating relational similarity as a protected attribute. Let $T : G \to \mathbb{R}^{n\times m}$ denote a topological function. Some examples of topological biases include:

1. **Node popularity:** If our downstream task demands that our predictions be independent of node popularity, we introduce protected attributes according to node degree. In that case, we consider the topological feature set $T(V,E) = \deg(V) = [\deg(v_1), \ldots, \deg(v_n)]$.

2. **Neighborhood composition:** For neighborhood composition, we bin the percentage compositions of the feature. In this case we set $T(V,E) = (\beta_1 A + \beta_2 A^2 + \ldots)V$, which

is the "aggregated reachable attribute value" (a similar condition has been considered in (Dong et al., 2022)). Note that A denotes the adjacency matrix here.

3. **Community affiliation:** For community affiliation, we identify groups as clusters determined by a node clustering algorithm (e.g., spectral clustering (von Luxburg, 2007)). We set $T(V, E) = \text{Cluster}(V, E) = [c_1, \ldots, c_n]$.

Then, we can assert fairness for groups of similar topological features by treating the topologically similar nodes together as a demographic group. In effect, this means replacing the condition $A = a$ for some sensitive label a with the condition that $T = t$ for some neighborhood description t (e.g., low-degree and high degree nodes).

Definition 2 (Topological Statistical Parity (SP-Topo)) Topological statistical parity *is achieved when* $\Pr\left[\hat{Y} = 1|T = 0\right] = \Pr\left[\hat{Y} = 1|T = 1\right]$. *Then,* topological statistical parity deviation *is defined as* $\left|\Pr\left[\hat{Y} = 1|T = 0\right] - \Pr\left[\hat{Y} = 1|T = 1\right]\right|$. *Smaller deviation means fairer (closer to satisfying the fairness constraint).*

3.2. Fairness-Accuracy Trade-Off

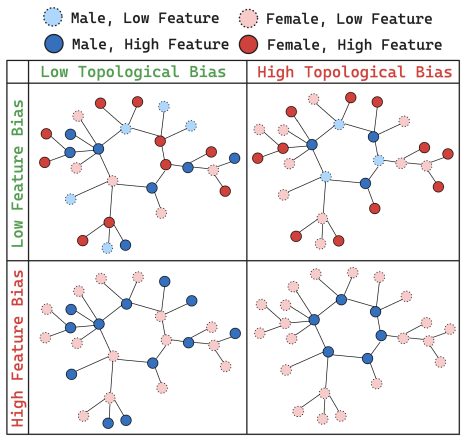


Figure 1: Illustration of *topological bias*. Feature bias is most prominent when different demographic groups have different mean features, whereas topological bias is most prominent when different kinds of nodes (e.g., leaf vs. central ring) have different mean features.

We begin by analyzing the impact of classical graph rewiring techniques on fairness using our proposed topological statistical parity metric. We investigate three rewiring techniques that are broadly representative of the main rewiring paradigms in the literature: FoSR, BORF, and DropEdge (see Sec. 2.2). In Tab. 2 we report accuracy and topological statistical parity for a GCN trained on the original German, Credit, Recidivism (Agarwal et al., 2021) and AMiner-S (Dong et al., 2024) datasets without rewiring ('Baseline'), or on the dataset after rewiring with DropEdge, FoSR, or BORF. We can see that while all rewiring methods boost accuracy, they do so at the cost of worsening fairness metrics. In contrast, fair rewiring can improve fairness, but at the cost of accuracy. We illustrate this on one recently proposed approach, FairEdit (Loveland et al., 2022) (see Tab. 7, 2). We see that FairEdit is able to mitigate both feature and topological bias. However, we observe a noticeable drop in performance, especially on the German Credit data set. We remark that FairEdit leverages a different fairness notion and is hence not directly comparable to the fairness-constrained rewiring approaches that we introduce in the next section. Additional details and results can be found in Apx. B.2.

3.3. Fairness-constrained Rewiring

Following our observation of a fairness-accuracy trade-off in classical and fair rewiring approaches, we ask whether it is possible to balance both desirable properties. In this section

Data	Baseline	FoSR	BORF	DropEdge	FairEdit
German	0.684	0.708	0.711	0.693	0.628
	0.0004	0.054	0.117	0.158	0.019
Credit	0.725	0.732	-	0.713	0.727
	0.046	0.054	-	0.070	0.050
Recidivism	0.882	0.888	-	0.887	0.884
	0.321	0.333	-	0.313	0.339
AMiner-S	0.912	0.918	0.927	0.922	-
	0.177	0.212	0.225	0.196	-

Table 2: **Fairness-Accuracy Trade-off.** We report accuracy (↑, top) and topological statistical parity (↓, bottom) for no rewiring (Baseline), classical rewiring techniques (FoSR, BORF, DropEdge) and fair rewiring (FairEdit). Missing values indicate that experiments were infeasible with the respective method (see discussion of computational complexity). We note that FairEdit does not optimize for topological statistical parity, but for a different fairness metric and is hence not directly comparable (see Apx. B.2 for additional results).

we propose a *soft fairness constraint*, which can be integrated into classical rewiring approaches to counteract the introduced topological biases. Classical rewiring approaches perform edge perturbations according to their presumed contribution to over-smoothing and over-squashing effects. FoSR approximates the effect that adding a new edge would have on the graph's spectral gap to determine which edges could most effectively mitigate over-squashing. Curvature-based rewiring methods such as BORF compute the discrete Ricci curvature of each given edge and remove the most positively curved edges to reduce over-smoothing while adding additional edges to the neighborhoods of particularly negatively curved edges to alleviate over-squashing. DropEdge randomly sparsifies the graph.

Soft fairness constraint We propose to weight the implicit "edge score" computed by these approaches by a fairness factor with the goal of selecting edges to perturb not only according to potential mitigation of over-smoothing and over-squashing, but also to counteract topological biases. We define fairness factors for adding an edge (u, v) that does not already exist as $\mathcal{F}^+(u,v) = \frac{\mathcal{L}(V,E\cup\{(u,v)\})}{\mathcal{L}(V,E)}$ and for removing an existing edge (u, v) as $\mathcal{F}^-(u,v) = \frac{\mathcal{L}(V,E\setminus\{(u,v)\})}{\mathcal{L}(V,E)}$. Here, $\mathcal{L}$ denotes a loss function. In our setting, we choose a topological loss that encodes different types of topological bias. Let $\mathcal{T}_i^a$ denote the distribution of values of the ith topological feature in the demographic group a. Then, the topological loss is computed as $\mathcal{L}_{topo} = \frac{1}{m}\sum_{i=1}^{m} \max_{a,a'} |\mathbb{E}[\mathcal{T}_i^a] - \mathbb{E}[\mathcal{T}_i^{a'}]|)$, i.e., the max absolute difference between the mean of a single topological feature for each group, averaged over all features.

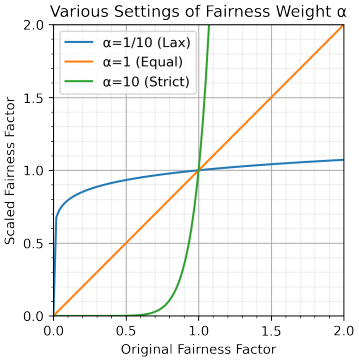


Figure 2: Different values of α to compute the adjusted fairness factor.

The rewiring methods described in Table 1 may add edges (FoSR), remove edges (DropEdge), or both (BORF), so $\mathcal{F}^+$ will be used in Fair-FoSR and Fair-BORF and $\mathcal{F}^-$ will be used in Fair-DropEdge and Fair-BORF.

Fair-FoSR For a given edge $(u, v) \in E$, let x_u and x_v be the eigenvectors of the graph Laplacian corresponding to nodes u and v and let d_u and d_v be their degrees (before adding the edge). Then we compute the fairness score $\frac{2x_u x_v}{\sqrt{(1+d_u)(1+d_v)}}\mathcal{F}^+(u,v)^\alpha$.

The first term is the approximate effect on the spectral gap of G of adding (u, v). FoSR seeks to minimize this term over all edges (u, v) that are not already in the graph. We note that if adding (u, v) to G is beneficial for fairness, $\mathcal{L}(V, E \cup \{(u, v)\}) < \mathcal{L}(V, E)$, so $\mathcal{F}^+(u, v) < 1$, so the whole product becomes smaller and the edge (u, v) is more likely to be selected and added to G. α is the weight on the fairness factor that controls how much impact the fairness factor has (see the "Controlling Accuracy vs. Fairness" paragraph).

Fair-BORF. To mitigate over-squashing, BORF adds edges to the neighborhoods of particularly negatively curved edges (u, v) by adding an edge between the nodes p and q that account for the biggest contribution to the distance between the neighborhoods of the adjacent nodes, i.e., $W_1(\mu_u, \mu_v) = \sum_{(p,q)} \pi(p, q)d(p, q)$. In other words, BORF chooses to add $(p^*, q^*) = \text{argmax } \pi(p, q)d(p, q)$. Adding our fairness factor, the objective becomes $\pi(p, q)d(p, q) \left(\mathcal{F}^+(p, q)\right)^{-\alpha}$. If adding (u, v) to G is beneficial for fairness, we again have $\mathcal{F}^+(u, v) < 1$, so multiplying with the inverse increases the objective, and hence makes (u, v) more likely to be selected in this case. BORF also removes the k most positively curved edges to reduce over-smoothing, where k is a hyperparameter. We propose a fair variant of this by considering the $3k$ most positively curved edges. For each of these, we compute $\kappa(u, v) \left(\mathcal{F}^-(u, v)\right)^{-\alpha}$ and sort the edges according to this product. The k edges with the highest values are removed. Note that the factor 3 was selected based on best empirical performance, but could be tuned separately, if desired.

Fair-DropEdge. DropEdge normally mitigates over-smoothing by removing an edge (u, v) uniformly at random with probability p, where p is a hyperparameter. We propose to instead drop a given edge (u, v) with probability $p \cdot \mathcal{F}^-(u, v)^{-\alpha}$ so that edges whose removal is more beneficial for fairness have a (slightly) higher chance of being removed.

Controlling Accuracy vs. Fairness. To be able to control how much we weight the original objectives of the rewiring methods considered here compared to the fairness factor, we propose to take the powers $\mathcal{F}^+(u, v)^\alpha$ and $\mathcal{F}^-(u, v)^\alpha$, where α is a hyperparameter that can be freely chosen. For example, $\alpha = 0$ will place no value on fairness and reduce the objective to what it was in the original method. Conversely, choosing $\alpha >> 1$ will place most of the weight on the fairness factor.

4. Experiments

Experimental Setup. We consider six node classification datasets commonly used in the graph fairness literature (see Apx. B.1). For each dataset, we use a 50/25/25 train/val/test split and train for up to 500 epochs or until the validation accuracy has not improved for 100 consecutive epochs.

Results. Our results show that fairness-constrained rewiring allows for interpolating between classical and fair rewiring. As we see in Figure 3, for AMiner-Small, fairness-constrained approaches ($\alpha > 0$) consistently improve statistical parity over unconstrained rewiring ($\alpha = 0$) across all three approaches. This is corroborated by our systematic experiments across data sets (Tables 3, 4, and 5). In addition, Figure 3 shows that fairness-constrained rewiring leads to a higher accuracy in comparison to the baseline (no rewiring)

Dataset	FoSR ($\alpha=0$)	$\alpha = 1/10$	$\alpha = 1$	$\alpha=10$	Baseline
AMiner-S	0.918 ± 0.004 0.21238 ± 0.03750	0.916 ± 0.004 0.18562 ± 0.03256	0.915 ± 0.005 0.18829 ± 0.03389	0.916 ± 0.003 0.18160 ± 0.02643	0.912 ± 0.003 0.17731 ± 0.03421
Credit	0.732 ± 0.021 0.05431 ± 0.01577	0.729 ± 0.020 0.04750 ± 0.01389	0.731 ± 0.019 0.05237 ± 0.01164	0.728 ± 0.017 0.04828 ± 0.00967	0.725 ± 0.017 0.04577 ± 0.01222
Facebook	0.783 ± 0.007 0.19199 ± 0.06777	0.784 ± 0.011 0.13079 ± 0.02833	0.783 ± 0.008 0.14013 ± 0.01674	0784 ± 0.004 0.14345 ± 0.00867	0.777 ± 0.005 0.17838 ± 0.00705
German	0.708 ± 0.014 0.05390 ± 0.00468	0.702 ± 0.009 0.00643 ± 0.00151	0.695 ± 0.010 0.00326 ± 0.00117	0.688 ± 0.006 0.00131 ± 0.00089	0.684 ± 0.007 0.00041 ± 0.00033
Recidivism	0.888 ± 0.003 0.33282 ± 0.01145	0.891 ± 0.008 0.32548 ± 0.00971	0.889 ± 0.006 0.32857 ± 0.00924	0.886 ± 0.005 0.30047 ± 0.01326	0.882 ± 0.003 0.32053 ± 0.01278
Tolokers	0.683 ± 0.013 0.26374 ± 0.05228	0.680 ± 0.011 0.25409 ± 0.04521	0.677 ± 0.010 0.25736 ± 0.03892	0.672 ± 0.018 0.24452 ± 0.03662	0.666 ± 0.022 0.24941 ± 0.03074

Table 3: Comparison of accuracy (top, ↑) and topological statistical parity (bottom, ↓) of Fair-FoSR with different values of α. Results for additional values of α can be found in Apx. B.4. Baseline is no rewiring.

Rewiring	AMiner-S	Facebook	German
BORF ($\alpha=0$)	0.927 ± 0.005 0.22459 ± 0.04278	0.796 ± 0.017 0.19842 ± 0.01681	0.711 ± 0.014 0.11723 ± 0.07382
$\alpha=0.01$	0.928 ± 0.007 0.18823 ± 0.03995	0.781 ± 0.016 0.16101 ± 0.02887	0.708 ± 0.010 0.06614 ± 0.01452
$\alpha=0.1$	0.923 ± 0.005 0.17016 ± 0.03820	0.789 ± 0.09 0.18891 ± 0.02301	0.702 ± 0.009 0.05572 ± 0.01870
$\alpha = 1$	0.924 ± 0.006 0.17358 ± 0.04472	0.784 ± 0.005 0.18209 ± 0.00788	0.704 ± 0.011 0.04581 ± 0.01392
$\alpha=10$	0.920 ± 0.004 0.16927 ± 0.04810	0.777 ± 0.015 0.17881 ± 0.01924	0.707 ± 0.010 0.01008 ± 0.00766
$\alpha=100$	0.921 ± 0.003 0.16744 ± 0.03824	0.783 ± 0.017 0.16014 ± 0.03207	0.700 ± 0.011 0.00642 ± 0.00147
Baseline	0.912 ± 0.003 0.17731 ± 0.03421	0.777 ± 0.005 0.17838 ± 0.00705	0.684 ± 0.007 0.00041 ± 0.00033

Table 4: Comparison of accuracy (top, ↑) and topological statistical parity (bottom, ↓) of Fair-BORF with different values of α. Baseline is no rewiring.

on AMiner-Small, which is again cons
analysis shows that this can be observe

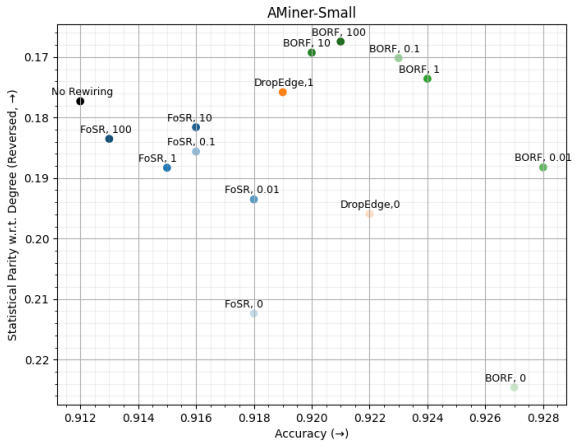


Figure 3: Fairness-accuracy trade-off for all three rewiring methods. The number indicates the fairness weight α; darker colors indicate more weight on fairness. Note that the y-axis is flipped –better is up and to the right. Full results in Tab 3, 4, 5.

Varying α in our experiments confirms that higher values generally lead to lower statistical parity (i.e., better mitigation of biases) and lower values to higher accuracy (i.e., better mitigation of over-smoothing and over-squashing effects). However, the results for different values of α are often close, indicating that the parameter should be carefully tuned in practise. When cross-comparing rewiring approaches across data sets, we observe that BORF generally provides the best fairness-accuracy trade-off. However, BORF has limited scalability, as it requires the computation of Ollivier's Ricci curvature for each edge, which scales as $O(|E|d_{max}^3)$ (where d_{max} denotes the maximum degree). Due to this, we only tested BORF on the three smaller data sets, as the curvature computation was not feasible on the larger data sets with >100k edges and high edge density (i.e., high node degrees).

Dataset	DropEdge	Fair DropEdge	Baseline
AMiner-S	0.922 ± 0.002	0.919 ± 0.002	0.912 ± 0.003
	0.19590 ± 0.02601	0.17581 ± 0.00391	0.17731 ± 0.03421
Credit	0.713 ± 0.023	0.738 ± 0.013	0.725 ± 0.017
	0.07011 ± 0.01549	0.07661 ± 0.05029	0.04577 ± 0.01222
Facebook	0.782 ± 0.013	0.793 ± 0.019	0.777 ± 0.005
	0.18798 ± 0.02750	0.15878 ± 0.02541	0.17838 ± 0.00705
German	0.693 ± 0.006	0.691 ± 0.006	0.684 ± 0.007
	0.15768 ± 0.02681	0.09831 ± 0.00765	0.00041 ± 0.00033
Recidivism	0.887 ± 0.004	0.886 ± 0.002	0.882 ± 0.003
	0.31312 ± 0.04493	0.29020 ± 0.02525	0.32053 ± 0.01278
Tolokers	0.662 ± 0.043	0.709 ± 0.034	0.666 ± 0.042
	0.22546 ± 0.03769	0.21977 ± 0.03521	0.24941 ± 0.03074

Table 5: Comparison of accuracy (top, ↑) and topological statistical parity (bottom, ↓) of Fair-DropEdge ($\alpha = 1$) with vanilla DropEdge. Baseline is no rewiring.

5. Discussion

To the best of our knowledge, this paper is the first to explicitly evaluate fairness-accuracy trade-offs in graph rewiring. We define a first metric for evaluating this trade-off, which captures relational bias as a distinct quantity from feature bias. We restrict ourselves to group fairness notions with a specific focus on statistical parity. Future work could extend this metric to a wider range of classical fairness notions, including individual fairness notions.

Our second main contribution is a fairness-constrained rewiring approach, where we impose our new fairness metric as a soft constraint in classical rewiring approaches. Our experimental results illustrate that adding such a constraint effectively balances fairness and accuracy in node-level tasks. Our framework is modular in that both the fairness metric and the rewiring technique could be easily exchanged for other approaches. In this work we consider three rewiring approaches which are popular representatives of three main types of rewiring techniques. A systematic comparison of a wider range of rewiring approaches could be an interesting direction for future work. We observed that fairness-constrained curvature-based rewiring (BORF) performs well, but is limited in scalability due to the high complexity of the curvature computation. Future work could investigate other curvature-based rewiring approached with more scalable curvature notions (Fesser and Weber, 2023) (see Apx. F for some early insights).

Furthermore, we have focused our evaluation on the rewiring step, which is a pre-processing routine performed prior to training. While we have tuned the hyperparameters of the baseline, additional ablations on the architecture design, such as the choice of the GNN base layer or the role of encodings, could be considered in future work. In addition, we hope to expand the comparison with fair rewiring approaches beyond the ablations in this work. However, we note that fair rewiring approaches are typically designed to enforce a specific fairness notion, which makes cross-comparison of these approaches and comparison with our fairness-constrained rewiring approach challenging. Defining a unified framework of fairness metrics for graph rewiring approaches and performing a systematic comparison is a valuable avenue for further investigation.

Lastly, the experimental evaluation in this paper has focused on node-level tasks. Classical rewiring approaches are often benchmarked on graph-level tasks; it would be interesting to investigate fairness-accuracy trade-offs in these settings, too. A key roadblock for such an analysis is the lack of a dedicated graph-level benchmark data set with protected attributes. Creating such a benchmark is an important direction for future work.

Acknowledgments

JW acknowledges support from the Harvard College Research Program (HCRP). MW was supported by NSF Awards DMS-2406905 and CBET-2112085, and a Sloan Research Fellowship.

References

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

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=i80OPhOCVH2.

Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in AI. Technical Report MSR-TR-2020-32, Microsoft, May 2020. URL https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/.

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *International Conference on Machine Learning*, pages 2528–2547. PMLR, 2023.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online, 2020. Association for Computational Linguistics.

Teresa Bono, Karen Croxson, and Adam Giles. Algorithmic fairness in credit scoring. *Oxford Review of Economic Policy*, 37(3):585–617, 2021.

Yushun Dong, Ninghao Liu, Brian Jalaian, and Jundong Li. Edits: Modeling and mitigating data bias for graph neural networks. In *Proceedings of the ACM web conference 2022*, pages 1259–1269, 2022.

Yushun Dong, Zhenyu Lei, Zaiyi Zheng, Song Wang, Jing Ma, Alex Jing Huang, Chen Chen, and Jundong Li. Pygdebias: A python library for debiasing in graph learning. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1019–1022, 2024.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM*

SIGKDD international conference on knowledge discovery and data mining, pages 259–268, 2015.

Lukas Fesser and Melanie Weber. Mitigating over-smoothing and over-squashing using augmentations of forman-ricci curvature. In *The Second Learning on Graphs Conference*, 2023.

Lukas Fesser, Sergio Serrano de Haro Iváñez, Karel Devriendt, Melanie Weber, and Renaud Lambiotte. Augmentations of forman's ricci curvature and their applications in community detection. *arXiv preprint arXiv:2306.06474*, 2023.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, volume 2, pages 729–734, 2005.

Laura Gustafson, Chloe Rolland, Nikhila Ravi, Quentin Duval, Aaron Adcock, Cheng-Yang Fu, Melissa Hall, and Candace Ross. Facet: Fairness in computer vision evaluation benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20370–20382, 2023.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *NIPS*, pages 1024–1034, 2017.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3323–3331, 2016.

Van Thuy Hoang, O Lee, et al. Mitigating degree biases in message passing mechanism by utilizing community structures. *arXiv preprint arXiv:2312.16788*, 2023.

Zhimeng Jiang, Xiaotian Han, Chao Fan, Zirui Liu, Na Zou, Ali Mostafavi, and Xia Hu. Fmp: Toward fair graph message passing against topology bias. *arXiv preprint arXiv:2202.04187*, 2022.

Jian Kang and Hanghang Tong. Algorithmic fairness on graphs: Methods and trends. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4798–4799, 2022.

Jian Kang, Jingrui He, Ross Maciejewski, and Hanghang Tong. Inform: Individual fairness on graph mining. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 379–389, 2020.

Jian Kang, Yan Zhu, Yinglong Xia, Jiebo Luo, and Hanghang Tong. Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1214–1225, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512169. URL https://doi.org/10.1145/3485447.3512169.

Kedar Karhadkar, Pradeep Kr Banerjee, and Guido Montúfar. Fosr: First-order spectral rewiring for addressing oversquashing in gnns. *arXiv preprint arXiv:2210.11790*, 2022.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017.

Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yanying Li, Xiuling Wang, Yue Ning, and Hui Wang. Fairlp: Towards fair link prediction on social network graphs. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):628–639, May 2022.

Daniil Likhobaba, Nikita Pavlichenko, and Dmitry Ustalov. Toloker Graph: Interaction of Crowd Annotators, 2023. URL https://github.com/Toloka/TolokerGraph.

Xiao Lin, Jian Kang, Weilin Cong, and Hanghang Tong. Bemap: Balanced message passing for fair graph neural network. In *Learning on Graphs Conference*, pages 37–1. PMLR, 2024.

Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. Tail-gnn: Tail-node graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1109–1119, 2021.

Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. On generalized degree fairness in graph neural networks. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i4.25574. URL https://doi.org/10.1609/aaai.v37i4.25574.

Donald Loveland, Jiayi Pan, Aaresh Farrokh Bhathena, and Yiyang Lu. Fairedit: Preserving fairness in graph neural networks through greedy graph editing. *arXiv preprint arXiv:2201.03681*, 2022.

Yan Luo, Min Shi, Muhammad Osama Khan, Muhammad Muneeb Afzal, Hao Huang, Shuaihang Yuan, Yu Tian, Luo Song, Ava Kouhana, Tobias Elze, et al. Fairclip: Harnessing fairness in vision-language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12289–12301, 2024.

Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. Subgroup generalization and fairness of graph neural networks. *Advances in Neural Information Processing Systems*, 34:1048–1061, 2021.

Khang Nguyen, Nong Minh Hieu, Vinh Duc Nguyen, Nhat Ho, Stanley Osher, and Tan Minh Nguyen. Revisiting over-smoothing and over-squashing using ollivier-ricci curvature. In *International Conference on Machine Learning*, pages 25956–25979. PMLR, 2023.

Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 469–481, 2020.

Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: towards fair graph embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 3289–3295. AAAI Press, 2019. ISBN 9780999241141.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

Jackson Sargent and Melanie Weber. Identifying biases in legal data: An algorithmic fairness perspective. *arXiv preprint arXiv:2109.09946*, 2021.

Candice Schumann, Jeffrey Foster, Nicholas Mattei, and John Dickerson. We need fairness and explainability in algorithmic hiring. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2020.

Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Transactions on Artificial Intelligence*, 3(3):344–354, 2021.

Arjun Subramonian, Levent Sagun, and Yizhou Sun. Networked inequality: Preferential attachment bias in graph neural network link prediction. In *Forty-first International Conference on Machine Learning*, 2024.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy, 2019. Association for Computational Linguistics.

Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Investigating and mitigating degree-related biases in graph convoltuional networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, page 1435–1444, 2020.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(1):395–416, 2007.

Ruijia Wang, Xiao Wang, Chuan Shi, and Le Song. Uncovering the structural fairness in graph contrastive learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

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

Appendix A. Additional Background

A.1. Other Fairness Metrics

A key shortcoming of statistical parity, which we introduced in the main text, is that it may not allow for learning an optimal predictor (Dwork et al., 2012), inducing a significant fairness-accuracy trade-off. This has motivated the introduction of several refinements, including equalized odds and equal opportunity (Hardt et al., 2016). *Equalized odds* ensures equal true positive and false positive rates across demographic groups.

Definition 3 (Equalized Odds) *Equalized odds is satisfied when*

$$\Pr\left[\hat{Y}=1|A=0,Y=y\right]=\Pr\left[\hat{Y}=1|A=1,Y=y\right]\quad\forall y\in\{0,1\}\,.$$

Equalized odds deviation is measured as

$$\max_{y\in\{0,1\}}\left(\left|\Pr\left[\hat{Y}=1|A=0,Y=y\right]-\Pr\left[\hat{Y}=1|A=1,Y=y\right]\right|\right).$$

Equal opportunity is weaker than equalized odds in that it only equalizes true positive rates across demographic groups.

Definition 4 (Equal Opportunity) *Equal opportunity occurs when*

$$\Pr\left[\hat{Y}=1|A=0,Y=1\right]=\Pr\left[\hat{Y}=1|A=1,Y=1\right]\,.$$

The equal opportunity deviation is thus

$$\left|\Pr\left[\hat{Y}=1|A=0,Y=1\right]-\Pr\left[\hat{Y}=1|A=1,Y=1\right]\right|\,.$$

Remark 5 *We note that notions of* topological equalized odds *and* topological equal opportunity *that account for topological bias can be introduced in analogy to sec. 3.1.*

Appendix B. Additional Experimental Details and Results

B.1. Datasets

To verify the effectivenss of our proposed method, we consider six node classification datasets that are commonly used in the graph fairness literature. We list the topological characteristics and the protected attributes of all datasets in Table 6. We note that some of the data sets (especially German and Tolokers) have a large class imbalance, which can cause challenges during training. Additional details and hyperparameter choices for all data sets are reported in Apx. G.

Dataset	# Nodes	# Edges	# Features	Label	Protected Attribute
AMiner-S (Dong et al., 2024)	39,424	52,460	5,694	Subject	Continent
Credit (Agarwal et al., 2021)	30,000	200,526	13	Default	Age
Facebook (Leskovec and Mcauley, 2012)	1,045	53,498	573	Education	Gender
German (Agarwal et al., 2021)	1,000	24,970	27	Behavior	Gender
Recidivism (Agarwal et al., 2021)	18,876	403,977	18	Recidivism	Race
Tolokers (Likhobaba et al., 2023)	11,758	519,000	10	Banned	English

Table 6: Overview of node classification datasets.

B.2. Comparison with Fair Rewiring baselines

Fairness-informed rewiring has a short history with various incarnations still being proposed. Previous works in this area are the FairEdit (Loveland et al., 2022) and EDITS (Dong et al., 2022) algorithms. FairEdit seeks to edit edges to minimize counterfactual loss based on the graph fairness of a trained model on the edited graph, identifying edges by the magnitude of the loss gradient on the adjacency matrix. On the other hand, EDITS optimizes against the Wasserstein distance of the distribution of neighborhood node features for various sensitive groups via proximal gradient descent. Both of these methods compute gradients on the adjacency matrix—a tough task. The strength of our method, however, is that our method is a preprocessing technique with interpretable fairness weights, and it is model-agnostic.

While our method is thus not directly comparable to these previous methods for those reasons, we provide fairness measures of the rewired graphs given by FairEdit for the following three datasets: German, Recidivism, and Credit.

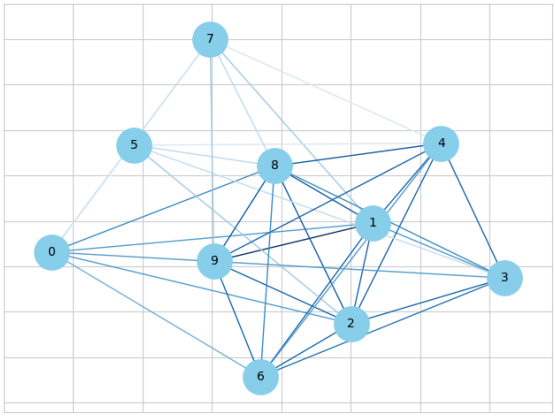
Dataset	Acc (↑)	CI	Time	SP-Demo	SP-Topo	EO-Demo	EO-Topo
German	0.628	0.032	0.294	0.003	0.019	0.003	0.279
Credit	0.727	0.019	0.274	0.049	0.050	0.050	0.178
Recidivism	0.884	0.004	0.391	0.060	0.339	0.023	0.764

Table 7: FairEdit Baseline

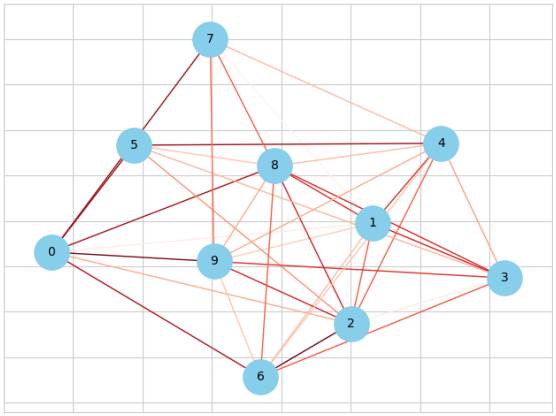
In general, we find that the FairEdit baseline results in worse performance than the no rewiring baseline. This is probably more revealing of the inability for FairEdit-rewired graphs to generalize to even slightly different GCN architectures.

B.3. Comparison of accuracy-preserving and fairness-promoting rewiring

We illustrate the difference between both types of rewiring approaches by visualizing differences in the edges they target in Fig. 4.



Edges colored according to their Ollivier-Ricci curvatures. Darker blue indicates a higher curvature value.



Edges colored according to the fairness factors that their removal would yield. Darker red indicates a higher value.

Figure 4: We illustrate differences in the edges that classical and fair rewiring methods target. We show a subgraph of the Toloker dataset with its edges colored according to (a) their Ollivier-Ricci curvatures and (b) their topological fairness factors. We expect that rewiring based on (a) yields a higher accuracy at the cost of higher topological bias (observe that curvature picks up only on the dense subgraph on the bottom-right), and (b) yields better statistical parity at the cost of reduced accuracy. Note that our proposed fairness-constrained rewiring methods allow for interpolating between both cases.

B.4. Additional rewiring results

For completeness, we provide results for FOSR for a wider range of values of α included in Fig. 3 in Tab. 8.

Dataset	$\alpha = 1/100$	$\alpha = 100$
AMiner-S	0.918 ± 0.006 0.19352 ± 0.04182	0.913 ± 0.004 0.18352 ± 0.02856
Credit	0.734 ± 0.017 0.05761 ± 0.01466	0.729 ± 0.018 0.04623 ± 0.01438
Facebook	0.789 ± 0.010 0.14787 ± 0.03367	0.781 ± 0.004 0.13179 ± 0.00858
German	0.706 ± 0.011 0.02317 ± 0.00613	0.687 ± 0.005 0.00062 ± 0.00046
Recidivism	0.894 ± 0.007 0.32922 ± 0.01108	0.881 ± 0.007 0.30621 ± 0.01235
Tolokers	0.682 ± 0.014 0.25352 ± 0.03124	0.669 ± 0.016 0.22163 ± 0.04172

Table 8: Comparison of Fair-FoSR with different values of α. For each dataset, the top row represents accuracy (↑) and the bottom row represents topological statistical parity (↓).

Appendix C. Equal Opportunity

Figure 5 shows how statistical parity and equal opportunity are not the same (e.g., DropEdge is the worst in terms of statistical parity but only third-worst in terms of equal opportunity). That being said, the overall trends are unchanged between the two fairness measures.

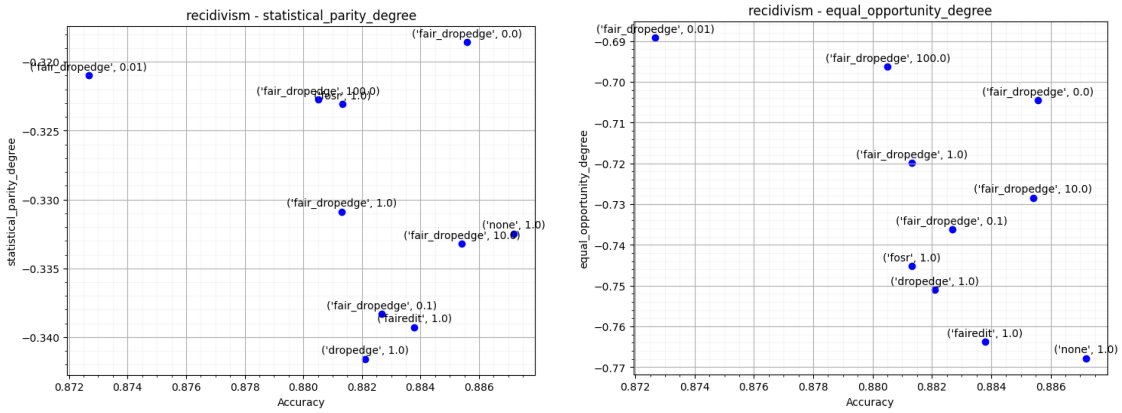


Figure 5: Statistical Parity vs. Equal Opportunity. Fairness measures graphed in negative to match visual default of up is better.

Appendix D. Results for GIN

We present additional results for Fair-FoSR and Fair-BORF for GIN in Tab. 9 and 10.

Dataset	**FoSR**	**Fair FoSR** ($\alpha = 1$)	**Baseline**
AMiner-S	0.836 ± 0.072	0.833 ± 0.081	0.821 ± 0.067
	0.389 ± 0.041	0.346 ± 0.038	0.331 ± 0.026
Credit	0.773 ± 0.038	0.770 ± 0.032	0.754 ± 0.021
	0.481 ± 0.064	0.446 ± 0.057	0.419 ± 0.036
Facebook	0.764 ± 0.017	0.765 ± 0.021	0.752 ± 0.013
	0.130 ± 0.014	0.092 ± 0.012	0.084 ± 0.007
Recidivism	0.903 ± 0.016	0.891 ± 0.009	0.882 ± 0.007
	0.182 ± 0.043	0.145 ± 0.032	0.137 ± 0.028
Toloker	0.790 ± 0.027	0.782 ± 0.045	0.784 ± 0.023
	0.518 ± 0.121	0.439 ± 0.077	0.417 ± 0.062

Table 9: Comparison of Fair-FoSR ($\alpha = 1$) with vanilla FoSR using GIN. For each dataset, the top row represents accuracy (↑) and the bottom row represents topological statistical parity (↓).

Dataset	**BORF**	**Fair BORF** ($\alpha = 1$)	**Baseline**
AMiner-S	0.836 ± 0.032	0.839 ± 0.034	0.821 ± 0.037
	0.396 ± 0.047	0.411 ± 0.055	0.331 ± 0.026
Facebook	0.773 ± 0.022	0.767 ± 0.019	0.752 ± 0.013
	0.183 ± 0.015	0.119 ± 0.016	0.084 ± 0.007

Table 10: Comparison of Fair-BORF ($\alpha = 1$) with vanilla BORF using GIN. For each dataset, the top row represents accuracy (↑) and the bottom row represents topological statistical parity (↓).

Appendix E. Results for GAT

We present additional results for Fair-FoSR and Fair-BORF for GAT in Tab. 11 and 12.

Dataset	FoSR	Fair FoSR ($\alpha = 1$)	Baseline
AMiner-S	0.903 ± 0.004	0.901 ± 0.003	0.898 ± 0.005
	0.19882 ± 0.03305	0.17280 ± 0.02721	0.14368 ± 0.02988
Facebook	0.773 ± 0.006	0.771 ± 0.007	0.764 ± 0.004
	0.17412 ± 0.05332	0.13986 ± 0.01536	0.14163 ± 0.00634

Table 11: Comparison of Fair-FoSR ($\alpha = 1$) with vanilla FoSR using GAT. For each dataset, the top row represents accuracy (↑) and the bottom row represents topological statistical parity (↓).

Dataset	BORF	Fair BORF ($\alpha = 1$)	Baseline
AMiner-S	0.908 ± 0.006	0.906 ± 0.007	0.901 ± 0.004
	0.20532 ± 0.03720	0.15641 ± 0.04227	0.15769 ± 0.02678
Facebook	0.788 ± 0.013	0.781 ± 0.004	0.775 ± 0.005
	0.17257 ± 0.01422	0.14613 ± 0.00625	0.00153 ± 0.00129

Table 12: Comparison of Fair-BORF ($\alpha = 1$) with vanilla BORF using GAT. For each dataset, the top row represents accuracy (↑) and the bottom row represents topological statistical parity (↓).

Appendix F. Results with AFRC

Dataset	AFR-3	Fair AFR-3 ($\alpha = 1$)	Baseline
AMiner-S	0.931 ± 0.008	0.926 ± 0.011	0.912 ± 0.003
	0.21461 ± 0.03379	0.14968 ± 0.04082	0.17731 ± 0.03421
Facebook	0.794 ± 0.016	0.788 ± 0.007	0.777 ± 0.005
	0.19672 ± 0.01623	0.17664 ± 0.00625	0.17838 ± 0.00705

Table 13: Comparison of Fair-AFR-3 ($\alpha = 1$) with vanilla AFR-3 using GCN. For each dataset, the top row represents accuracy (↑) and the bottom row represents topological statistical parity (↓).

BORF has rather large time complexity (cubic in max degree, see Table 1), but there do exist faster-to-compute alternatives. Augmented Forman Curvature (AFRC) (Fesser et al., 2023) is a different formulation that takes $O(m \cdot d_{max})$ to compute for every edge (this kind of curvature, AFR-3, uses cycles of length at most 3). We present results in Table 13, which shows Fair AFR-3 (Fair BORF with the curvature notion replaced) indicates a drop in topological statistical parity deviation compared to the vanilla rewiring. There is an insignificant difference between Fair BORF and Fair AFR-3.

Appendix G. Hyperparameter Choices

For each dataset, we tested GNNs with 3, 4, or 5 layers; hidden dimension 128, 256, and 512; learning rate 1e-3, 1e-4, and 1e-5; dropout rate 0, 0.1, and 0.2; and with and without

skip connections. We always use the architecture with the highest baseline accuracy, i.e. without any rewiring. For the number of edges to add with FoSR, we tried 100, 200, 300, 400, and 500, and use the number that gave the highest accuracy on a given dataset. We did the same with the number of edges to add and to remove in BORF. For DropEdge, we tried 0.1, 0.2, and 0.3.

We note that our analysis focuses on evaluating the effect of different rewiring approaches and fairness weights. This has motivated the aforementioned protocol, where the hyperparameters are tuned for the baseline and the used for all rewiring approaches.

Hyperparameters	**AMiner-S**	**Credit**	**Facebook**	**Recidivism**	**Toloker**
Num. Layers	4	4	4	4	5
Hidden Dim.	128	128	128	128	512
Learning Rate	1e-5	1e-5	1e-5	1e-5	1e-5
Dropout	0	0	0	0	0.2
Skip Connections	No	No	No	No	Yes
FoSR Iter.	100	500	100	500	500
BORF Add	100	500	100	500	500
BORF Remove	100	0	100	100	0
DropEdge	0.2	0.2	0.2	0.2	0.2

Table 14: Overview of parameter choices.