Certification and Trade-off of Multiple Fairness Criteria in Graph-based Spam Detection

Kai Burkholder[a*], Kenny Kwock[b*], Yuesheng Xu[c*], Jiaxin Liu[b], Chao Chen[b], Sihong Xie[b†]
[a]University of San Francisco, [b]Lehigh University, [c]New York University
[*]Equal contribution, ordered by last name alphabetically [†]Corresponding author (xiesihong1@gmail.com)

ABSTRACT

Spamming reviews are prevalent in review systems to manipulate seller reputation and mislead customers. Spam detectors based on graph neural networks (GNN) exploit representation learning and graph patterns to achieve state-of-the-art detection accuracy. The detection can influence a large number of real-world entities and it is ethical to treat different groups of entities as equally as possible. However, due to skewed distributions of the graphs, GNN can fail to meet diverse fairness criteria designed for different parties. We formulate linear systems of the input features and the adjacency matrix of the review graphs for the certification of multiple fairness criteria. When the criteria are competing, we relax the certification and design a multi-objective optimization (MOO) algorithm to explore multiple efficient trade-offs, so that no objective can be improved without harming another objective. We prove that the algorithm converges to a Pareto efficient solution using duality and the implicit function theorem. Since there can be exponentially many trade-offs of the criteria, we propose a data-driven stochastic search algorithm to approximate Pareto fronts consisting of multiple efficient trade-offs. Experimentally, we show that the algorithms converge to solutions that dominate baselines based on fairness regularization and adversarial training.

CCS CONCEPTS

• **Information systems → Spam detection**; • **Computing methodologies → Neural networks**; • **Applied computing → Multi-criterion optimization and decision-making**.

KEYWORDS

Fairness; multiple objective optimization; graphs

ACM Reference Format:
Kai Burkholder[a*], Kenny Kwock[b*], Yuesheng Xu[c*], Jiaxin Liu[b], Chao Chen[b], Sihong Xie[b†]. 2021. Certification and Trade-off of Multiple Fairness Criteria in Graph-based Spam Detection. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482325

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482325

1 INTRODUCTION

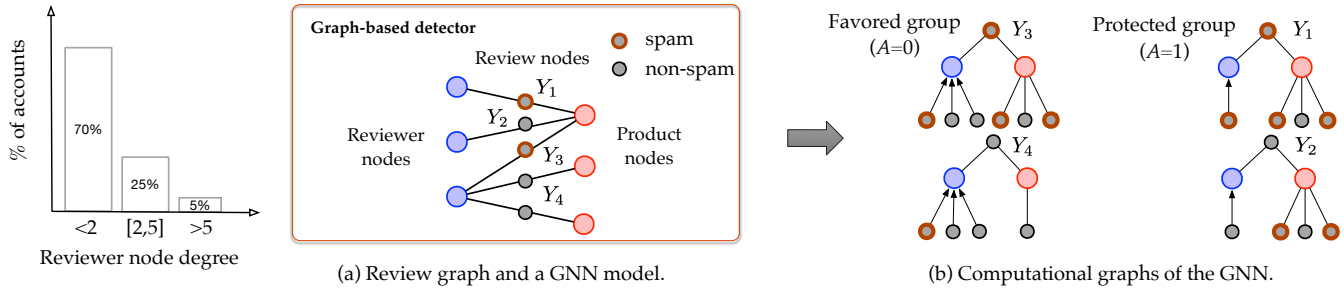
Online reviews evaluate the reputations of businesses and guide customers on e-commerce websites, such as Amazon [14], Yelp [30], and Google Play [35]. However, these websites have also attracted many spammers[1] to manipulate product ratings and the less informed customers. Numerous detectors are proposed, using features derived from texts [20, 34, 42], reviewer behaviors [27, 33, 46], and graphs [19, 21, 28, 37]. The literature has seen a steady improvement in detection accuracy, which is nonetheless not the only evaluation metric. Since the detection can affect many parties in e-commerce, ethical aspects, such as fairness, have caught much attention. We focus on the fairness of graph neural networks (GNN), which combine representation learning and patterns of the review graphs connecting reviewers, reviews, and products to deliver superior detection accuracy [28]. The GNN detector outputs $\Pr(\hat{Y}_i = 1|G)$ as the probability of the suspiciousness of the i-th review. The reviews with high probabilities will be screened or automatically removed.

Whether a review needs to be blocked should depend only on the characteristics relevant to spamming (such as the intention and impact of the review [47]). However, such characteristics are unobservable due to the anonymity of the spammers. Furthermore, the review graphs can contain attributes, such as the number of reviews by an author, that can bias the detection[2].

We focus on the fairness issue due to the highly skewed distribution of node degree (see Figure 1, panel (a)). Prior accuracy-focusing detectors did not consider the degree and can have unfair detection. For example, reviewers or products with fewer reviews can have a higher chance of being screened and such treatment is unfair to the majority of reviewers. We define the protected group of reviewers (indicated by the sensitive attribute $A = 1$) as those that have posted less than a certain number of reviews, and the remaining reviewers are in the favored group ($A = 0$). The reviews are grouped according to their authors (reviewers).

To demonstrate the fairness issue, Figure 1 panel (b) shows the computation graphs for example reviews from the two groups. A review from the favored group ($A = 0$) is connected to its author (the larger blue circles), which is connected to many other reviews that can be spam (red circles) or non-spams (black circles). Informative detection signals of the reviews in the input (bottom) layer can be diluted by the aggregation operator (averaging or summing) as messages are passed up. $\Pr(\hat{Y}_i = 1|G)$ calculated at the root of reviews from group 0 can be lower than those from group 1, due to the dilution. The discrepancy between the probabilities is termed "disparate impact" [11], and "statistical parity" means the two probabilities are equal.

[1]According to [40], about 40% of the reviews on Amazon are fake ones.
[2]The unobservable characteristics form the "construct feature space" (CFS) and the observable features form the "observable feature space" (OFS) [15]



(a) Review graph and a GNN model.

(b) Computational graphs of the GNN.

Figure 1: (a) An example review graph G. The GNN runs on G and predicts $\Pr(Y_j = 1|G)$, the probability of the corresponding review being suspicious, given the graph G. (b) Computation graphs of $\Pr(Y_j = 1|G)$ for reviews from the two groups. The roots in group 0 have blue children (the reviewers), which have many children at the input (bottom) layer that can be spams or non-spams. The messages from non-spams can dilate the suspicious features from the spam nodes, making group 0 spams harder to detect, thus a gap between the NDCGs of the two groups.

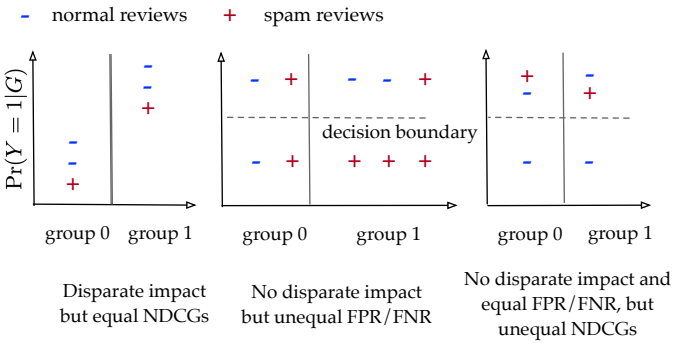


Figure 2: Satisfying one fairness criterion does not guarantee satisfaction of another fairness criterion.

Statistical parity is not sufficient to ensure fairness: different false positive rates between two groups means that the innocent reviews from one group are more likely to be screened, even when the same percentage of reviews from the two groups are screened [17]. It is necessary to enforce multiple fairness constraints (e.g., statistical parity and equalized odds [17]). Note that enforcing one fairness criterion does not guarantee fairness in another metric (see Figure 2 and Figure 4). Prior work on fair learning on graphs [1, 5, 7, 36] only concern about a single fairness criterion.

The first challenge is to understand the conflict among multiple fairness criteria and detection performance. Recent work [15, 24, 26] certified multiple fairness constraints using the feasibility of the equality and/or inequality constraints. However, the constraints are not formulated based on graph properties and detection performance is not considered. Since the class distribution in spam detection is highly skewed, ranking-based metrics, such as NDCG, may be more appropriate. However, closed-form constraints for certificating fair ranking have not been formulated. Fair representation learning [1, 5, 9, 31] is agnostic about fairness metrics and cannot reveal their conflicts. Second, when multiple hard fairness constraints are infeasible, one needs to relax the constraints and find Pareto efficient trade-offs among the relaxations[3] Prior work [1, 9, 23] used fairness-regularized optimization, where a user-specific hyperparameter controls the relative importance of fairness and classification accuracy. Nonetheless, these methods did not guarantee Pareto efficiency and thus not reveal the necessary trade-offs (see Figure 6). While there are multi-objective optimization (MOO) algorithms designed for fair machine learning [22], the convergence proof is incomplete and it is not clear how the graph and GNN can affect the convergence. Lastly, the relative importance of the multiple metrics is unknown before model is trained and evaluated. Preference-based MOO [22, 32] requires preference vectors from users. However, the number of preference vectors increases exponentially in the number of objectives.

To address the above challenges, we study four widely-used fairness criteria, including a ranking-based metric, and formulate linear constraints to certificate the simultaneous satisfaction of these criteria. The constraints are GNN-specific and expressed in terms of the underlying graph structure and the input feature vectors. The formulation is of independent interest beyond spam detection. We adopt a well-studied gradient-based MOO algorithm to search a Pareto optimal solution efficiently. We develop a proof of the convergence of the algorithm using the implicit function theorem, completing the proof in [22]. Unlike [3, 16, 45, 48], we don't assume convex formulations of the fairness criteria, since convexity can over-relax the fairness constraints and lose the regularization [29]. Building on the convergence of the MOO algorithm, we adopt a stochastic search algorithm [39] [4] for a more efficient data-driven search without user-specified preference vectors. Experimentally, we demonstrate the need to enforce multiple fairness criteria and the convergence of the proposed algorithms. The stochastic search algorithm finds more efficient solutions that dominate solutions found by regularization-based methods [45, 48] and adversarial learning-based methods [1, 5, 31].

2 PRELIMINARIES

2.1 Spam detection based on GNN

Graph neural networks have been used for spam detection [28]. GNN operates on a graph $G = (\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$. Each node v_i has a feature vector $\mathbf{x}_i$ encoding various

[3] A trade-off is "Pareto efficient", if improving one metric (e.g., accuracy) necessarily harm at least another metric (e.g., a fairness criterion).

[4] In their paper, GNN is not studied and certificates are not analyzed.

spam detection features. The undirected edge $e_{ij} \in \mathcal{E}$ indicates that v_i and v_j are related. Let $W \in \{0,1\}^{N\times N}$ be the adjacency matrix of the graph G, so that $W_{ij} = 1$ if and only if $e_{ij} \in \mathcal{E}$. GNN is a K-layered neural network. Let $\{h_j^{(k)}, j = 1,\ldots,N, k = 1,\ldots,K\}$ be the feature vectors of the node v_j output at the k-th layer. The input feature vector $\mathbf{x}_j$ is considered to be $h_j^{(0)}$ at layer 0. GNN computes $h_j^{(k)}, k \in \{1,\ldots,K\}$, as follows:

$$a_j^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{h_i^{(k-1)} : e_{ij} \in \mathcal{E}\right\}\right), \quad (1)$$

$$h_j^{(k)} = \text{COMBINE}^{(k)}\left(h_j^{(k-1)}, a_j^{(k)}\right), \quad (2)$$

where the AGGREGATE function finds a single vector $a_j^{(k)}$ from the vectors of the neighboring nodes at the previous layer. These functions can take various forms. For example, AGGREGATE$^{(k)}$ can be the mean of the input vectors, and COMBINE$^{(k)}$ the composition of the ReLU and an affine mapping parameterized by $\boldsymbol{\theta}^{(k)}$ [25].

Let $\boldsymbol{\theta} = [\boldsymbol{\theta}^{(0)},\ldots,\boldsymbol{\theta}^{(K)}] \in \mathbb{R}^d$ denote all the d trainable parameters of the GNN. The prediction $\Pr(\hat{Y}_j = y_j|\boldsymbol{\theta}; G)$ is computed by the sigmoid function $\sigma(z_j)$. $z_j = \left\langle \boldsymbol{\theta}^{(K)}, h_j^{(K)} \right\rangle$ is the logic and $\boldsymbol{\theta}^{(K)}$ maps the last layer's output $h_j^{(K)}$ to z_j.

The aggregation function in multiple layers can represented by a computation graph. For a node v_j, to compute $\Pr(\hat{Y}_j|\boldsymbol{\theta}; G)$, a spanning tree of the graph G is constructed with root at v_j. The tree is cut-off at the depth K. We will use the computation graphs to analyze the compatibility of multiple fairness criteria.

GNN needs to be trained on labeled nodes (assumed to be the first n of the N nodes on G, whose labels are denoted by $y_j \in \{0,1\}$, $j = 1,\ldots,n$). Since the number of spams and non-spams are imbalanced, we choose to maximize the NDCG metric for evaluating rankings:

$$\frac{1}{Z}\sum_{j=1}^{n} \mathbb{1}[y_j = 1]\frac{1}{\log(r_j + 1)}, \quad (3)$$

where r_j is the ranking position of the j-th labeled node among all labeled nodes sorted in descending order of $\Pr(\hat{Y}_j = y_j|\boldsymbol{\theta}; G)$. Z is the maximal possible value of $\sum_{j=1}^{n} \mathbb{1}[y_j = 1]\frac{1}{\log(r_j+1)}$ across all rankings so that the loss is in $[0,1]$. NDCG is not differentiable due to the sorting, and we adopt the differentiable surrogate [6]

$$\ell_1(\boldsymbol{\theta}; G) = \frac{1}{Z}\sum_{j,j':y_j<y_j} \log(1 + \exp(z_i - z_j)), \quad (4)$$

where Z is the total number of pairs of positive and negative nodes. The detection can be evaluated on the test set using the NDCG.

2.2 Optimizing fairness metrics

A commonly found fairness criteria is disparate impact [8, 11]

$$\min\left\{\frac{\Pr(\hat{Y} = 1|A = 0)}{\Pr(\hat{Y} = 1|A = 1)}, \frac{\Pr(\hat{Y} = 1|A = 1)}{\Pr(\hat{Y} = 1|A = 0)}\right\}. \quad (5)$$

The fairness is maximized when the above metric is 1, and the predicted probability is independent of A. To facilitate gradient-based optimization, we adopt the corresponding surrogate [8, 10]:

$$\ell^{(\text{DI})}(\boldsymbol{\theta}; G) = |\Pr(\hat{Y} = 1|A = 0) - \Pr(\hat{Y} = 1|A = 1)|, \quad (6)$$

where $\Pr(\hat{Y} = 1|A = a)$ is estimated as the percentage of the reviews from group a classified positive by $\hat{Y}$. Since the GNN has probabilistic outputs, we use the approximation

$$\Pr(\hat{Y} = 1|A = a; \boldsymbol{\theta}, G) = \frac{\sum_{j=1}^{n} \mathbb{1}[A_j = a]\Pr(\hat{Y}_j = 1|\boldsymbol{\theta}, G)}{\sum_{j=1}^{m} \mathbb{1}[A_j = a]}. \quad (7)$$

Equalized odd is another fairness criterion proposed in [18]. Two specific instances of equalized odd is "equalized false positive rate", enforced by the fairness loss

$$\ell^{(\text{EFPR})}(\boldsymbol{\theta}; G) = |\Pr(\hat{Y} = 1|A = 0, Y = 0) - \Pr(\hat{Y} = 1|A = 1, Y = 0)|, \quad (8)$$

and "equalized false negative rate" enforced by the loss

$$\ell^{(\text{EFNR})}(\boldsymbol{\theta}; G) = |\Pr(\hat{Y} = 0|A = 0, Y = 1) - \Pr(\hat{Y} = 0|A = 1, Y = 1)|. \quad (9)$$

The conditional probabilities in $\ell^{(\text{EFPR})}$ and $\ell^{(\text{EFNR})}$ can be estimated similarly as in Eq. (7) but conditioning on both A and Y. Lastly, we would prefer the detection performance measured in NDCG to be equal across two groups, using the following loss function

$$\ell^{(\text{XN})}(\boldsymbol{\theta}; G) = \left|\frac{1}{Z_0}\sum_{j=1}^{n_0} \mathbb{1}[y_j = 1, A_j = 0]\frac{1}{\log(r_j^0 + 1)}\right. \quad (10)$$

$$\left. - \frac{1}{Z_1}\sum_{j=1}^{n_1} \mathbb{1}[y_j = 1, A_j = 1]\frac{1}{\log(r_j^1 + 1)}\right|, \quad (11)$$

with Z_0 and Z_1 being the normalization for the two groups and r_j^0 and r_j^1 the ranking position of the j-th node from two groups ($A = 0$ and $A = 1$), respectively (nodes are ranked within each group). We approximate the group-wise NDCG using Eq. (4) within individual groups $\ell^{(\text{XN})}(\boldsymbol{\theta}; G)$.

3 MULTI-OBJECTIVE FAIR DETECTION

3.1 Motivating multi-objective optimization

Let's revisit Figure 2. In the left subfigure, by averaging the posteriors of the three instances within each of the two groups, it is clear that the averages are different, leading to disparate impact. However, the detection NDCG are the same across the groups. In the middle, we let all the $\Pr(Y = 1|G)$ be positioned so that their distances to the decision boundary are the same, leading to statistical parity (no disparate impact). However, the FPR and FNR are different between both groups. Lastly, on the right subfigure, we see no disparate impact and the FPR/FNR are equal across the groups, but the NDCGs are different between the two groups. The above example shows that enforcing one fairness criterion (e.g., FPR) is insufficient for ensuring another fairness criterion (e.g., NDCG). Is it possible to satisfy a set of fairness criteria simultaneously?

3.2 (Im)possibility of satisfying multiple fairness criteria

The above question has been studied in [24, 26], but their statements are not about the representation learning on graphs. To simplified our analysis, we consider the linearized GNN [44]:

$$\Pr(\hat{Y}_j = 1|G; \boldsymbol{\theta}) = \sigma\left((\tilde{W})^K \mathrm{H}^{(0)} \prod_{k=0}^{K} \boldsymbol{\theta}^{(k)}\right), \quad (12)$$

where $H^{(0)} = [h_1^{(0)}, \ldots, h_n^{(0)}]^\top$ is the input feature matrix, $\tilde{W} = D^{-1}W$, and $D = \text{diag}(W\mathbb{1}_{n\times 1})$. We prove that certain fairness criteria can be translated into *linear* constraints over H^0 and W.

Let G_0 and G_1 be two groups defined by a sensitive attribute A and the random variables in group G_i be denoted by $\{Y_{i,j}\}$, where $i \in \{0,1\}$ and $j \in \{1, \ldots, |G_i|\}$. Define the indicator vector $\mathbb{1}[G_i]$, with 1's in the entries j for $Y_j \in G_i$ and 0 otherwise. The computation graph of the simplified GNN, when predicting the class of any node Y_j, is a spanning tree of height K rooted at $\hat{Y}_j$. The spanning tree's leaves are the input vectors $h_{j'}^{(0)}$ of the node $Y_{j'}$ reachable from Y_j on the graph G in K hops [5]. Example computation graphs are given in Figure 3.

Theorem 3.1. *Assume the linearized GNN with fixed parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(K)})$. If the rows of the matrix $\prod_{k=0}^{K} \boldsymbol{\theta}^{(k)}$ are linearly independent, then an equality fairness constraint C based on disparate impact, EFPR, and EFNR, defined using the logits $z_{i,j}$ for nodes $Y_{i,j}$, is satisfied if*

$$\frac{1}{|G_0|}\mathbb{1}[G_0]^\top(\tilde{W})^K H^{(0)} = \frac{1}{|G_1|}\mathbb{1}[G_1]^\top(\tilde{W})^K H^{(0)}, \tag{13}$$

Proof. The averaged logits from group G_i is

$$\frac{1}{|G_i|}\mathbb{1}[G_i]^\top(\tilde{W})^K H^{(0)} \prod_{k=0}^{K} \boldsymbol{\theta}^{(k)}. \tag{14}$$

By equating the two averages, we have

$$\left[\frac{1}{|G_0|}\mathbb{1}[G_0]^\top(\tilde{W})^K H^{(0)} - \frac{1}{|G_1|}\mathbb{1}[G_1]^\top(\tilde{W})^K H^{(0)}\right] \prod_{k=0}^{K} \boldsymbol{\theta}^{(k)} = 0. \tag{15}$$

Since the rows of $\prod_{k=0}^{K} \boldsymbol{\theta}^{(k)}$ are linearly independent,

$$\frac{1}{|G_0|}\mathbb{1}[G_0]^\top(\tilde{W})^K H^{(0)} - \frac{1}{|G_1|}\mathbb{1}[G_1]^\top(\tilde{W})^K H^{(0)} = \mathbf{0}. \tag{16}$$

□

Corollary 3.1.1. *Under the assumptions of Theorem 3.1, the compatibility of S fairness equality criteria $C_1, \ldots, C_S$, with two groups $G_{0,s}$ and $G_{1,s}$, $s = 1, \ldots, S$, can be certificated by the feasibility of the following linear system*

$$\frac{1}{|G_{0,s}|}\mathbb{1}[G_{0,s}]^\top(\tilde{W})^K H^{(0)} = \frac{1}{|G_{1,s}|}\mathbb{1}[G_{1,s}]^\top(\tilde{W})^K H^{(0)}, s = 1, \ldots, S.$$

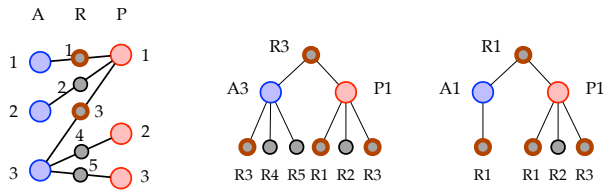
where $\mathbb{1}[G_{0,s}]$ and $\mathbb{1}[G_{1,s}]$ are the binary indicator vectors for the two groups defined by the criterion C_s.

Corollary 3.1.2. *Under the assumptions of Theorem 3.1, if an fairness criterion is defined across over S groups, the criterion is satisfied if the following linear system is feasible*

$$\frac{1}{|G_s|}\mathbb{1}[G_s]^\top(\tilde{W})^K H^{(0)} = \frac{1}{|G_t|}\mathbb{1}[G_t]^\top(\tilde{W})^K H^{(0)}, \forall s, t \in \{1, \ldots, S\}.$$

where $\mathbb{1}[G_s]$ and $\mathbb{1}[G_t]$ are the binary indicator vectors for the two groups G_s and G_t.

[5] In most neural network implementations, such as PyTorch, network parameters are leaf nodes of computation graphs. We don't consider parameters in the constraints since the parameters are fixed as constants.



(a) GNN (b) computational graphs

Figure 3: A running example demonstrating the (im)possibility.

A similar equality constraint can be proved for the ranking-based fairness criterion of equalized NDCG. Let G_i^+ (G_i^-, resp.) be the set of positive (negative, resp.) of group i, and $\mathbb{1}[G_i^+]$ and $\mathbb{1}[G_i^-]$ be the corresponding indicator vectors. Let $n_i^+ = |G_i^+|$ and $n_i^- = |G_i^-|$ be the number of positive and negative examples in group i, for $i = 0, 1$.

Theorem 3.2. *Under the assumptions of Theorem 3.1, the fairness criterion of equality in NDCG approximated using logits $z_{i,j}$ of $Y_{i,j}$ is satisfied if*

$$\frac{1}{n_0^- \times n_0^+}\Delta_0^\top(\tilde{W})^K H^{(0)} = \frac{1}{n_1^- \times n_1^+}\Delta_1^\top(\tilde{W})^K H^{(0)}, \tag{17}$$

with $\Delta_i = n_i^+ \times \mathbb{1}[G_i^-] - n_i^- \times \mathbb{1}[G_i^+]$, for $i = 0, 1$.

Proof. The proof of the theorem is similar to Theorem 3.1. To use the logits to define equality in NDCG, we replace $\log(1+\exp(z_{j'} - z_j))$ in the approximated NDCG loss function $\ell^{(\text{XN})}(\boldsymbol{\theta}; G)$ with $z_{j'} - z_j$. The summation in the definition of the loss function is then replaced with the inner product $\Delta_0^\top(\tilde{W})^K H^{(0)}$ and $\Delta_1^\top(\tilde{W})^K H^{(0)}$ for groups 0 and 1, respectively. □

The above theorems are applicable to general graphs beyond the review graphs. In the proof, rather than working with the output probabilities in the fairness constraints, as defined in Seciton 2.2, we relax the fairness criteria that use the probabilities $\Pr(Y|X)$ and $\Pr(Y|X, A)$ to use the logits z_j for the ease of analysis. Be cautioned that the closeness in the averaged logits is not equivalent to the closeness in the averaged probabilities. However, when the sigmoid function is used to compute the probabilities and the logits or their differences are near 0, the approximation is close.

A running example. We demonstrate the theorem on the review graph, in Figure 3 with three reviewer accounts, 5 reviews, and 3 products. The two spamming reviews are highlighted with red circles. The letters A, R, and P above the three columns denote the types (reviewer, review, and product) of the nodes in the respective columns, and the numbers besides each node identify a node of the type in those columns. For example, A_1 is the first reviewer and R_3 is the third review, which is a spamming review. For the ease of analysis, we assume that the simplified GNN has two layers ($K = 2$), and there is no edge connecting a node to itself. The adjacency matrix is row normalized by node degrees as in Eq. (12). The computation graphs are shown in Figure 3 panel (b).

The protected group has reviews R_1 and R_2, and the favored group has reviews R_3, R_4, and R_5. Statistical parity requires

$$\frac{1}{2}\left(h_{R1}^{(2)} + h_{R2}^{(2)}\right) = \frac{1}{3}\left(h_{R3}^{(2)} + h_{R4}^{(2)} + h_{R5}^{(2)}\right).$$

The superscripts indicate the second (output) layer of the GNN. Using the computational graphs, if the rows of $\boldsymbol{\theta}^{(0)}\boldsymbol{\theta}^{(1)}\boldsymbol{\theta}^{(2)}$ are linearly independent, the requirement becomes the following linear equality constraint:

$$\frac{1}{12}\left(5h_{R1}^{(0)} + 5h_{R2}^{(0)} + 2h_{R3}^{(0)}\right) = \frac{1}{18}\left(h_{R1}^{(0)} + h_{R2}^{(0)} + 4h_{R3}^{(0)} + 6h_{R4}^{(0)} + 6h_{R5}^{(0)}\right).$$

3.3 Learning a GNN satisfying multiple fairness requirements

What if multiple desired fairness criteria cannot be satisfied simultaneously? One can then find GNN models that trade one criterion for others, with the constraint that the trade-offs are efficient, meaning that improving one fairness criterion necessarily harms at least another criterion. Such a model is called "Pareto optimal (efficient)", which can be found by the following multi-objective optimization:

$$\min_{\boldsymbol{\theta}} \quad \ell(\boldsymbol{\theta}) = (\ell_1(\boldsymbol{\theta}), \ldots, \ell_m(\boldsymbol{\theta}))^\top, \tag{18}$$

where ℓ_i is some loss function mapping from $\boldsymbol{\Theta}$ to $\mathbb{R}_+$ and L is a function mapping from $\mathbb{R}^d$ to $\mathbb{R}^m$. We assume all loss functions are differentiable so that their gradients are well-defined. In particular, we always let the first objective function ℓ_1 be the ranking loss Eq. (4) to optimize spam detection performance. Depending on what fairness criteria are desired, the corresponding fairness loss functions can be appended as objective functions. For example, if we care about fairness defined by disparate impact, we let $m = 2$ and $\ell_2(\boldsymbol{\theta})$ be the loss defined in Eq. (5); if we want to ensure fairness defined in DI, FNR, FPR, and xNDCG, we can let $m = 5$ and $\ell_2(\boldsymbol{\theta}) = \ell^{\text{DI}}(\boldsymbol{\theta})$, $\ell_3(\boldsymbol{\theta}) = \ell^{\text{FNR}}(\boldsymbol{\theta})$, $\ell_4(\boldsymbol{\theta}) = \ell^{\text{FPR}}(\boldsymbol{\theta})$, and $\ell_5(\boldsymbol{\theta}) = \ell^{\text{XN}}(\boldsymbol{\theta})$.

Definition 3.3 (Dominance). *A model $\boldsymbol{\theta}$ is dominated by the model $\boldsymbol{\theta}'$, if $\ell(\boldsymbol{\theta}') \leq \ell(\boldsymbol{\theta})$ element-wisely and for at least one $i \in \{1, \ldots, m\}$, $\ell_i(\boldsymbol{\theta}') < \ell_i(\boldsymbol{\theta})$.*

Definition 3.4 (Pareto optimal and front). *A model $\boldsymbol{\theta}$ is Pareto optimal (or efficient) if it is not dominated by any other model. The Pareto front is image of the set of all Pareto optimal solutions under the mapping $\ell : \Theta \to \mathbb{R}^m$.*

To characterize Pareto optimal solutions, we define the $m \times d$ Jacobian matrix

$$(J(\boldsymbol{\theta}))_{i,j} = \frac{\partial \ell_i}{\partial \theta_j}(\boldsymbol{\theta}). \tag{19}$$

Unlike single objective optimization, at a local Pareto optimal solution $\boldsymbol{\theta}$, the Jacobian matrix $J(\boldsymbol{\theta})$ may not be all zero. That is, there exists a Pareto optimal solution $\boldsymbol{\theta}$ so that the gradient of ℓ_i is not a zero vector for at least one $i \in \{1, \ldots, m\}$. A necessary condition of a local Pareto optimal solution is that there is no vector $\mathbf{g} \in \mathbb{R}^d$ so that $J(\boldsymbol{\theta})\mathbf{g} < 0$, where the inequality is element-wise in the m objective values. If there is a vector $\mathbf{g}$ so that $J(\boldsymbol{\theta})\mathbf{g} < 0$, then $\mathbf{g}$ is a descent direction to make $\ell(\boldsymbol{\theta} + \alpha\mathbf{g}) \leq \ell(\boldsymbol{\theta}) + \beta\alpha\mathbf{g}^\top(J(\boldsymbol{\theta}))_j$ smaller than $\ell(\boldsymbol{\theta})$ with sufficiently small $\alpha \in (0, \beta)$ and $\beta \in (0, 1)$.

To certificate that $\boldsymbol{\theta}$ is Pareto optimal, or equivalently that there is no descent direction to further reduce all objectives, one can solve the following optimization problem [13]:

$$\min_{\tau, \mathbf{g}} \quad \tau + \tfrac{1}{2}\|\mathbf{g}\|^2 \tag{20}$$

$$\text{s.t.} \quad (A\mathbf{g})_i \leq \tau, i = 1, \ldots, m. \tag{21}$$

Algorithm 1 MOO for finding one Pareto optimal solution

Input: m objective functions $\ell_1, \ldots, \ell_m$ (NDCG and some fairness objective(s)), a small positive tolerance $\epsilon > 0$.
Output: a Pareto optimal solution $\boldsymbol{\theta}$.
Initialize GNN model $\boldsymbol{\theta}$.
for $t = 1, \ldots,$ **do**
  Find the gradients $(J(\boldsymbol{\theta}))_i$ of individual objective functions ℓ_i at the current solution $\boldsymbol{\theta}$.
  Use a QP solver to find the optimal dual variables $\lambda_1^*, \ldots, \lambda_m^*$, by solving the dual problem Eq. (22)-(23).
  Compute the multi-gradient $\mathbf{g} = \sum_{j=1}^m \lambda_j^*(J(\boldsymbol{\theta}))_j$.
  if at $\mathbf{g}$, $\max_j(J(\boldsymbol{\theta})\mathbf{g})_j > -\epsilon$ **then**
    break
  end if
  Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_k\mathbf{g}$.
end for
Return the GNN model $\boldsymbol{\theta}$.

where $A = J(\boldsymbol{\theta})$ is a constant matrix given $\boldsymbol{\theta}$. If $\boldsymbol{\theta}$ is a Pareto optimal solution, then $A\mathbf{g} \geq 0$ for any $\mathbf{g} \in \mathbb{R}^d$ and the optimal value of the above optimization is 0 by taking $\tau = 0$ and $\mathbf{g} = \mathbf{0} \in \mathbb{R}^d$. If $\boldsymbol{\theta}$ is not a Pareto optimal solution, then there is a $\mathbf{g} \neq \mathbf{0}$ so that $A\mathbf{g} < 0$ and $\tau = \max_i(A\mathbf{g})_i \leq -\frac{1}{2}\|\mathbf{g}\|^2 < 0$. Note that τ and the descent direction $\mathbf{g}$ are both functions of the current solution $\boldsymbol{\theta}$.

In practice, it is not necessary to find the global optimum of the above strongly convex optimization problem. Instead, finding a descent direction $\mathbf{g}$ so that $\tau + \frac{1}{2}\|\mathbf{g}\|^2$ is sufficiently smaller than 0 is good enough. According to [13], it is more common to solve the following dual problem of the above primal problem:

$$\max_{\boldsymbol{\lambda}} \quad -\tfrac{1}{2}\|\textstyle\sum_{j=1}^m \lambda_j(J(\boldsymbol{\theta}))_j\|^2 \tag{22}$$

$$\text{s.t.} \quad \textstyle\sum_{j=1}^m \lambda_j = 1, \lambda_j \geq 0, j = 1, \ldots, m. \tag{23}$$

The dual problem is a quadratic programming (QP) problem and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]$ is the set of dual variables for the m inequality constraints in Eq. (21). Off-shelf software and library can be adopted to find the approximately optimal $\boldsymbol{\lambda}^*$. After the QP is solved, if the current solution $\boldsymbol{\theta}$ is not Pareto optimal, a descent direction is obtained as a so-called "multi-gradient" $\mathbf{g} = \sum_{j=1}^m \lambda_j^*(J(\boldsymbol{\theta}))_j$, which is used to update the GNN parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_k \sum_{j=1}^m \lambda_j^*(J(\boldsymbol{\theta}))_j. \tag{24}$$

Otherwise, if $\tau = \max_j(A\mathbf{g})_j$ is not sufficiently smaller than 0 and $\boldsymbol{\theta}$ can be claimed to be Pareto optimal. The algorithm description is given in Algorithm 1. The learning rate η_k should be adjusted so that $\eta_k < (1 - \beta)/(2L_{\max})$ where $0 < \beta < 1$ is a pre-specified hyper-parameter and $L_{\max}$ is the maximum of the Lipschitz constants of the gradients of the objective functions.

Relation to regularization-based approaches. Compared with the training a fairness-regularized GNN [9], such as

$$\ell(\boldsymbol{\theta}; G) = \ell_1(\boldsymbol{\theta}; G) + \lambda\ell^{\text{DI}}(\boldsymbol{\theta}; G), \tag{25}$$

the QP-based approach can find the relative importance of different objective functions which are unknown a prior. Further, the

regularized GNN does not guarantee a Pareto optimal solution, as shown in the experiments.

Finding Pareto fronts. To find multiple Pareto optimal solutions in the Pareto fronts, Algorithm 2 is adopted from [39]. It maintains a list of dominating solutions in each outer iteration, while in each of the inner iterations, it randomly perturbs each previous dominating solution into several slightly different solutions ("local search"), which are further optimized by Algorithm 1. Dominated solutions are removed at the end of each outer iteration.

Algorithm 2 Searching the Pareto front with Stochastic Multi-Gradient

Input: graph G
Initialization: a list of a single GNN model $\mathcal{L}_0 = \{\boldsymbol{\theta}\}$.
for $t = 0, 1, \ldots$ **do**
  Let $\mathcal{L}_{t+1} = \emptyset$.
  for each model $\boldsymbol{\theta}$ in $\mathcal{L}_t$ **do**
    Sample r GNN parameters independently from $\mathcal{N}(\boldsymbol{\theta}, \sigma^2 I)$ (adding Gaussian noise to each dimension of $\boldsymbol{\theta}$).
    Add the sampled model to $\mathcal{L}_{t+1}$.
  end for
  Let $\mathcal{L}'_{t+1} = \emptyset$.
  for each model $\boldsymbol{\theta}$ in $\mathcal{L}_{t+1}$ **do**
    Apply Algorithm 1 to update $\boldsymbol{\theta}$ to $\boldsymbol{\theta}'$.
    Add $\boldsymbol{\theta}'$ to $\mathcal{L}'_{t+1}$.
  end for
  Remove models that are dominated from $\mathcal{L}'_{t+1}$.
  Let $\mathcal{L}_{t+1} = \mathcal{L}'_{t+1}$.
end for

3.4 Convergence to a Pareto efficient solution

It has been proved in [13], that Algorithm 1 will converge to a local Pareto optimal solution given that the objectives are Lipschitz continuously differentiable and the step sizes are selected using the Armijo method. Further, in [12], the authors proved that the rate of convergence for non-convex, convex, and strongly convex objective functions. There are discussions on whether to use convex relaxation of fairness metrics [3]. On the one hand, using convex objective functions can ensure convergence and the rate. On the other hand, neural networks are typically non-convex, even with convex loss functions, and too much relaxation can cause the fairness objectives to lose their effect [29].

We prove the convergence of Algorithm 1, with a key results stated in [12, 13] without proof. We found the convergence proof (Proof of Theorem 3, Section 4.6.4) in [22] also miss a key step (not proving how the dual variables converge to a stationary point). We close the gap by completing the proof in [12, 13].

Theorem 3.5. *(Theorem 3.1 of [12]) All loss functions are lower-bounded by zero. Let $\boldsymbol{\theta}^{(0)}$ be the initial GNN model and the maximal loss function value be $F^{\max} = \max\{\ell_1(\boldsymbol{\theta}^{(0)}), \ldots, \ell_m(\boldsymbol{\theta}^{(0)})\}$. Algorithm 1 generates a sequence $\{\boldsymbol{\theta}^{(t)}\}$ such that*

$$\min_{t=0,\ldots,T-1} \|\mathbf{g}^{(t)}\| \leq \sqrt{\frac{F^{\max}}{M}} \frac{1}{\sqrt{T}}, \tag{26}$$

where $M = \beta\eta_{\min}/2$ and $\eta_{\min} = \min\{(1-\beta)/2L_{\max}, 1\}$.

The theorem shows that the descent direction sequence $\{\mathbf{g}^{(t)}\}$ satisfies

$$\liminf_{t\to\infty} \|\mathbf{g}^{(t)}\| \to 0, \tag{27}$$

and by passing to a subsequence, there is a subsequence of the descent directions $\|\boldsymbol{\theta}^{(t_k)}\| \to 0$ as $k \to \infty$. $\boldsymbol{\theta}^{(t_k)}$ converges to a limit point $\boldsymbol{\theta}^*$ where the corresponding $\|\mathbf{g}^*\| = 0$. By Eq. (20)-(21), $\boldsymbol{\theta}^*$ is a Pareto optimal solution. In [12], the authors stated but did not prove why the corresponding dual variables $\boldsymbol{\lambda}$ also converges to a stationary point $\boldsymbol{\lambda}^*$. We close the gap by proving that $\boldsymbol{\lambda}(\boldsymbol{\theta})$ is a continuous function $\boldsymbol{\theta}$.

Theorem 3.6. *Let $f(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \sum_{j=1}^{m} \lambda_j \ell_j(\boldsymbol{\theta})$ be the objective function Eq. (22) of the dual problem, and $\boldsymbol{\lambda}^*$ be an optimal solution of the problem. $f : \mathbb{R}^{d+m} \to \mathbb{R}^m$. If $\nabla_{\boldsymbol{\lambda}} f(\boldsymbol{\theta}, \boldsymbol{\lambda})$ is full-rank, then there is a differentiable function $\boldsymbol{\lambda}(\boldsymbol{\theta})$ near $\boldsymbol{\lambda}^*$.*

Proof. Since the dual problem is a linearly constrained quadratic programming and is convex, there is a unique solution $\boldsymbol{\lambda}^*$ if the corresponding Hessian w.r.t. $\boldsymbol{\lambda}$ is positive definite. The constraints of the QP are

$$h(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \sum_{j=1}^{m} \lambda_j - 1 = 0, \tag{28}$$

$$f_j(\boldsymbol{\theta}, \boldsymbol{\lambda}) = -\lambda_j \leq 0, j = 1, \ldots, m. \tag{29}$$

The Lagrangian of the constrained QP is

$$L(\boldsymbol{\lambda}, \mu, \nu, \boldsymbol{\theta}) = f(\boldsymbol{\theta}, \boldsymbol{\lambda}) + \sum_{j=1}^{m} \mu_j f_j(\boldsymbol{\theta}, \boldsymbol{\lambda}) + \nu h(\boldsymbol{\theta}, \boldsymbol{\lambda})$$

The KKT conditions are

$$f_j(\boldsymbol{\theta}, \boldsymbol{\lambda}) \leq 0, j = 1, \ldots, m, \tag{30}$$

$$h(\boldsymbol{\theta}, \boldsymbol{\lambda}) = 0, \tag{31}$$

$$\mu_j \geq 0, j = 1, \ldots, m \tag{32}$$

$$\mu_j f_j(\boldsymbol{\theta}, \boldsymbol{\lambda}) = 0, j = 1, \ldots, m, \tag{33}$$

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\lambda}, \boldsymbol{\mu}, \nu, \boldsymbol{\theta}) = 0, \tag{34}$$

The three equalities above constitute a linear system $F(\boldsymbol{\lambda}, \mu, \nu, \boldsymbol{\theta}) = [\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda}, \mu, \nu, \boldsymbol{\theta}); \mu_1 f_1(\boldsymbol{\theta}, \boldsymbol{\lambda}); \ldots; \mu_m f_m(\boldsymbol{\theta}, \boldsymbol{\lambda}); h(\boldsymbol{\theta}, \boldsymbol{\lambda})] = \mathbf{0} \in \mathbb{R}^{m+1}$. By the Implicit Function Theorem [2], there is a neighborhood around $(\boldsymbol{\lambda}, \boldsymbol{\mu}, \nu, \boldsymbol{\theta})$ and a function $s : \boldsymbol{\theta} \to (\boldsymbol{\lambda}, \boldsymbol{\mu}, \nu)$ that is continuously differentiable in a neighbor of $\boldsymbol{\theta}$, with Jacobian being:

$$\nabla_{\boldsymbol{\theta}} s(\boldsymbol{\theta}) = -\nabla_{\boldsymbol{\lambda}, \mu, \nu} F(\boldsymbol{\lambda}, \mu, \nu, \boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\lambda}, \mu, \nu, \boldsymbol{\theta}). \tag{35}$$

The function s further satisfies $F(s(\boldsymbol{\theta}), \boldsymbol{\theta}) = 0$, the equalities in the KKT conditions. □

The differentiable function $s(\boldsymbol{\theta})$ maps from a model $\boldsymbol{\theta}$ to the optimal dual variable values when solving the problem Eq. (22)-(23). It shows that $\boldsymbol{\lambda}$ is a continuous function of $\boldsymbol{\theta}$. As a result, $\boldsymbol{\lambda}$ converges to $\boldsymbol{\lambda}(\boldsymbol{\theta}^*)$ as $\boldsymbol{\theta}$ converges to $\boldsymbol{\theta}^*$.

Note: the convergence proof applies to multiple objectives defined on a training set only. Convergence on the unseen test data requires more assumptions, such as sufficiently large training sets and identical training and test distributions.

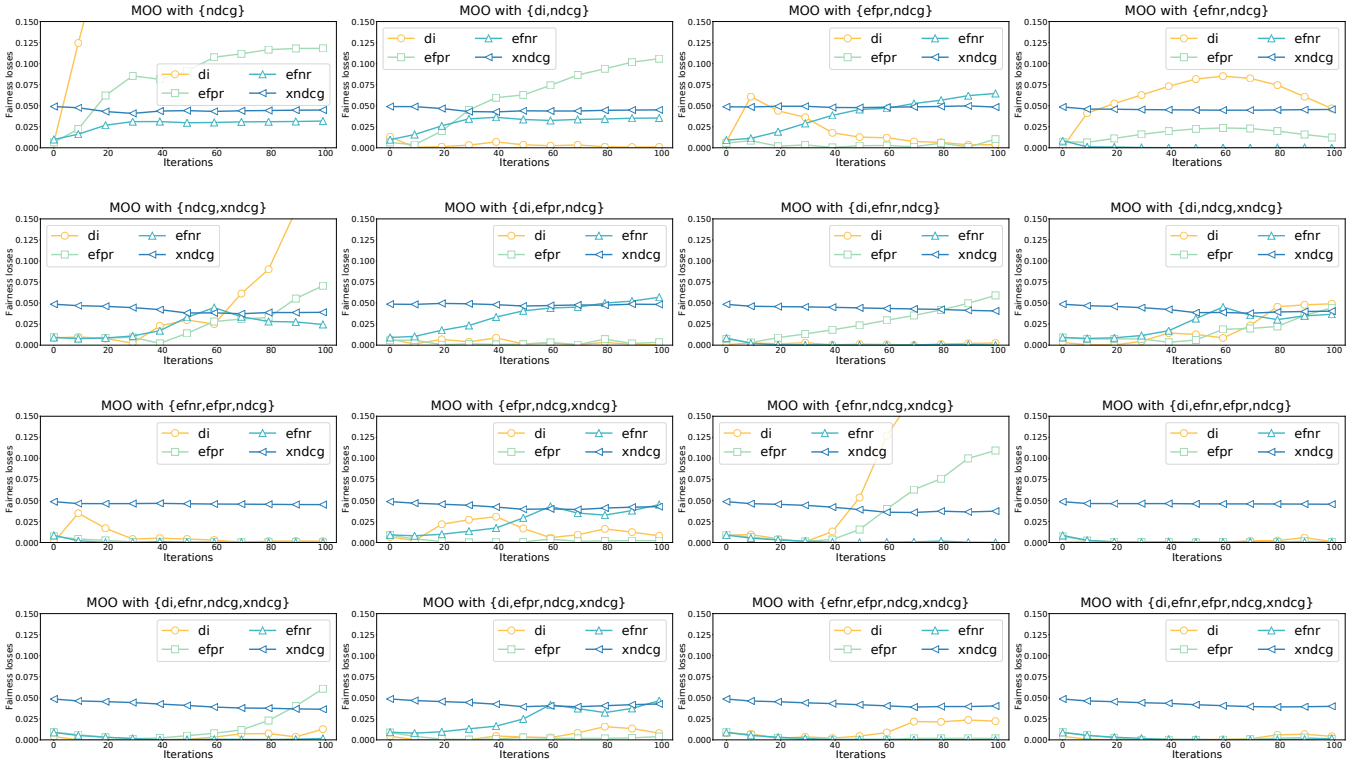


Figure 4: From top to bottom row, from left to right, we optimize NDCG along with every subset of the fairness objectives $\emptyset$, $\{\ell^{\text{DI}}\}$, $\{\ell^{\text{EFPR}}\}$, ..., $\{\ell^{\text{DI}}, \ell^{\text{EFNR}}, \ell^{\text{EFPR}}, \ell^{\text{XN}}\}$. Each subfigure tracks all fairness loss functions, even those not optimized by MOO, on the test set during training on YelpChi. Fairness loss not optimized can increased while other metrics are optimized.

Table 1: Review dataset statistics

Dataset	Dataset Statistics			$\frac{P(Y=1\|A=0)}{P(Y=1\|A=1)}$
	# Accounts	# Products	# Reviews (% spams)	
YelpChi	38063	201	67395 (13.23%)	0.0437
YelpNYC	160225	923	359052 (10.27%)	0.1446
YelpZip	260277	5044	608598 (13.22%)	0.0426

4 EXPERIMENTS

Experimental settings. We evaluate the trade-offs between multiple fairness metrics, and those between accuracy and fairness metrics, We adopt the three Yelp review datasets used previously for graph-based spam detection [37, 47] (see Table 1). We place in the favored group ($A = 0$) the reviewers who have the top 30% number of reviews, and the remaining reviewers in the protected group ($A = 1$). Reviews are grouped accordingly. The last column of the table shows the ratio of spams in the two groups ($P(Y = 1|A = 0)/P(Y = 1|A = 1)$). We can see that the class distributions are dependent on the attribute A: reviews from the favored group are less likely to be spams. We split the reviewers and their reviews into training (50%), validation (20%), and test (30%) sets, so that the ratio of spams and the bias are similar in the three sets. The isotropic normal distribution for sampling GNN parameters in Algorithm 2 has variance $\sigma^2 = 0.01$.

Evaluation metrics. Since the class distributions are imbalanced, we use NDCG to measure the detection accuracy. As we focus on Pareto efficiency, we measure how often the baselines' trained models are dominated by the models found by our algorithms. When there are only two objective functions, the dominance can be visualized. With all 5 objective functions, it is hard to visualize the dominance and we instead count the dominated solutions.

4.1 Experimental results

Do we need MOO? In Figure 2, we showed that optimizing one fairness metric may not guarantee the optimization of other metrics. Due to space limit, we only show the empirical results on the YelpChi dataset. We use Algorithm 1 to optimize the NDCG loss (ℓ_1) along with one of the 16 subsets of the 4 fairness losses. The smallest subset has no fairness loss and the algorithm only optimizes ℓ_1. The largest subset contains all 4 fairness losses and all fairness criteria are desired. By comparing the first two subfigures in the first row, one can see that disparate impact skyrocketed above the upper limit 0.14 in the first, but fell below 0.1 in the second subfigure when ℓ^{DI} is minimized explicitly. Interestingly, the fairness loss ℓ^{EFPR} slightly decreased but not much, while the other two fairness losses ℓ^{EFNR} and ℓ^{XN} remain the same. The third subfigure of the same row shows that, only when ℓ^{EFPR} is optimized by the algorithm will ℓ^{EFPR} be controlled. Similar effect on ℓ^{DI} can be observed by comparing the third and the last subfigures in the last row. In

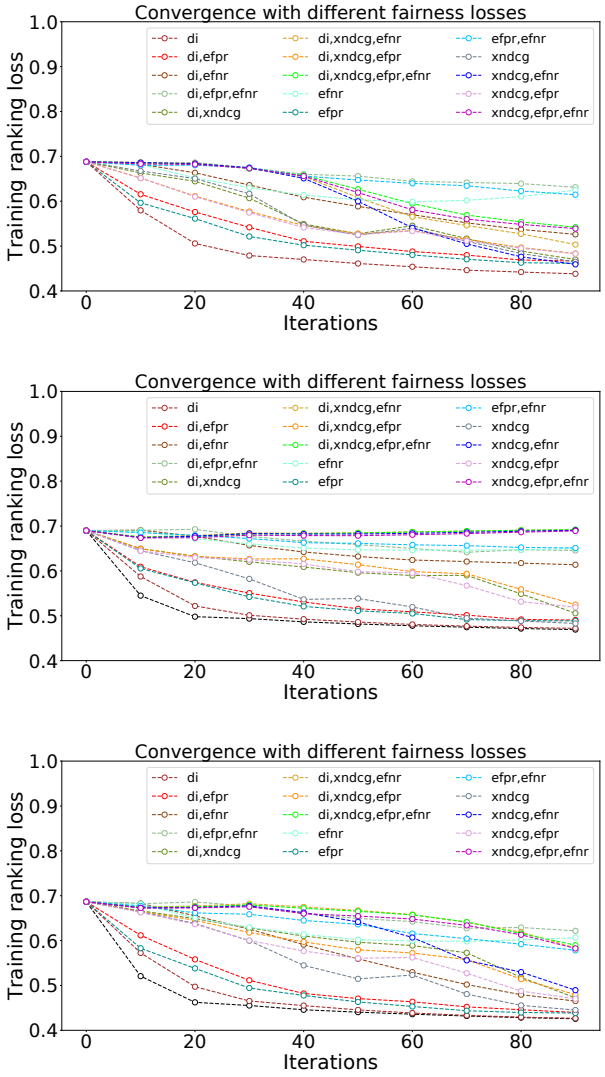


Figure 5: Convergence of ranking loss function, when optimized along with different sets of fairness loss functions. From top to bottom: YelpChi, YelpNYC, and YelpZip.

general, all fairness losses need to be minimized when all fairness criteria need to be met (the last subfigure).

Convergence of to a single Pareto efficient solution. We empirically demonstrate the convergence of Algorithm 1. Similar to Figure 4, on 3 datasets, we solve 45 MOO problems, each of which has the NDCG loss function ℓ_1 (Eq. (4)) and one of the 15 non-empty subsets of fairness losses. In Figure 5, we plot the convergence of ℓ_1 on the training set[6]. One can see that in 42 out of 45 MOO instances, the NDCG loss converges, with the three exceptions on the YelpNYC dataset when optimizing $[\ell_1, \ell^{\text{XN}}, \ell^{\text{EFNR}}]$, $[\ell_1, \ell^{\text{XN}}, \ell^{\text{EFPR}}, \ell^{\text{EFNR}}]$, and $[\ell_1, \ell^{\text{DI}}, \ell^{\text{XN}}, \ell^{\text{EFPR}}, \ell^{\text{EFNR}}]$. Note that the rates

[6]The convergence proof works on objective functions defined on the training set only, and generalization to test distribution requires some large-sample arguments.

Table 2: Number of models found by Algorithm 2 that are dominated by adversarial fairness learning. Sol's = Total solutions. Dom'd = Dominated.

	YelpChi		YelpNYC		YelpZip	
Epochs	# Sol's	#Dom'd	# Sol's	#Dom'd	# Sol's	#Dom'd
2	10	1	9	0	5	0
4	28	0	31	2	21	0
6	117	0	109	1	71	0
8	256	0	289	0	212	1
10	447	0	597	0	345	1

of convergences are different in different MOO problems. For example, the blue curve for optimizing $[\ell_1, \ell^{\text{XN}}, \ell^{\text{EFPR}}]$ only starts to decrease significantly at epoch 30 when it converges, while the brown and gray curves converge at around epoch 20. Such difference can be caused by the difference in the descent directions due to the different fairness objectives.

Converging to Pareto front. We optimize $[\ell_1, \ell^{\text{fair}}]$, where ℓ_1 is the NDCG loss and ℓ^{fair} is some fairness loss. In Figure 6, we plot the current dominating solutions in the list $\mathcal{L}_{t+1}$ in Algorithm 2 every 4 of the total of 20 epochs. It is clear that the fronts converge towards the lower-left corner of the 2 dimensional space $[\ell_1, \ell^{\text{fair}}]$.

We observe that the competitions between the ranking loss and each of ℓ^{DI}, ℓ^{EFNR}, and ℓ^{EFPR} are less severe, as the front closely approaches the lower-left corner where both losses are small. However, the trade-off between ℓ_1 and ℓ^{XN} is harder: pushing one loss down means the other loss will go up. This is probably because both losses are ranking-based and share the same battle ground.

Compare with baselines. We have two baselines:

- The fairness-regularized methods [23, 48, 49] need user-specified relative importance to balance the objectives. The number of balancing configurations grow exponentially in the number of objectives. To restrict the search space, we only optimize the scalar objective $\ell_1 + \lambda \ell^{\text{fair}}$ with $\lambda \in \{0.1, 1, 10\}$ only.
- The method in [9] used adversarial learning to obtain fair GNN without explicitly optimizing any fairness losses. Therefore, the learned representations can be evaluated against various combinations of fairness criteria.

Both baselines only find a single model that may not be Pareto efficient. In Figure 6, we compare Algorithm 2 with both baselines when optimizing only two objectives $[\ell_1, \ell^{\text{fair}}]$. In most cases, the baselines found solutions that are dominated by the Pareto fronts found by Algorithm 2. The only two exceptions happen on YelpNYC dataset, where both the regularization method and our algorithm are optimizing ℓ_1 along with ℓ^{EFNR} or ℓ^{XN}. The results of optimizing all 5 objectives are summarized in Table 2. Only the adversarial method is compared since the regularization method requires user-specified preference vectors. The Pareto fronts found by Algorithm 2 in different training epoch dominated the baseline.

5 RELATED WORK

Fairness on graphs. Fairness in graphs has been studied in several contexts. In [10, 50], fair Markov random fields structure learning

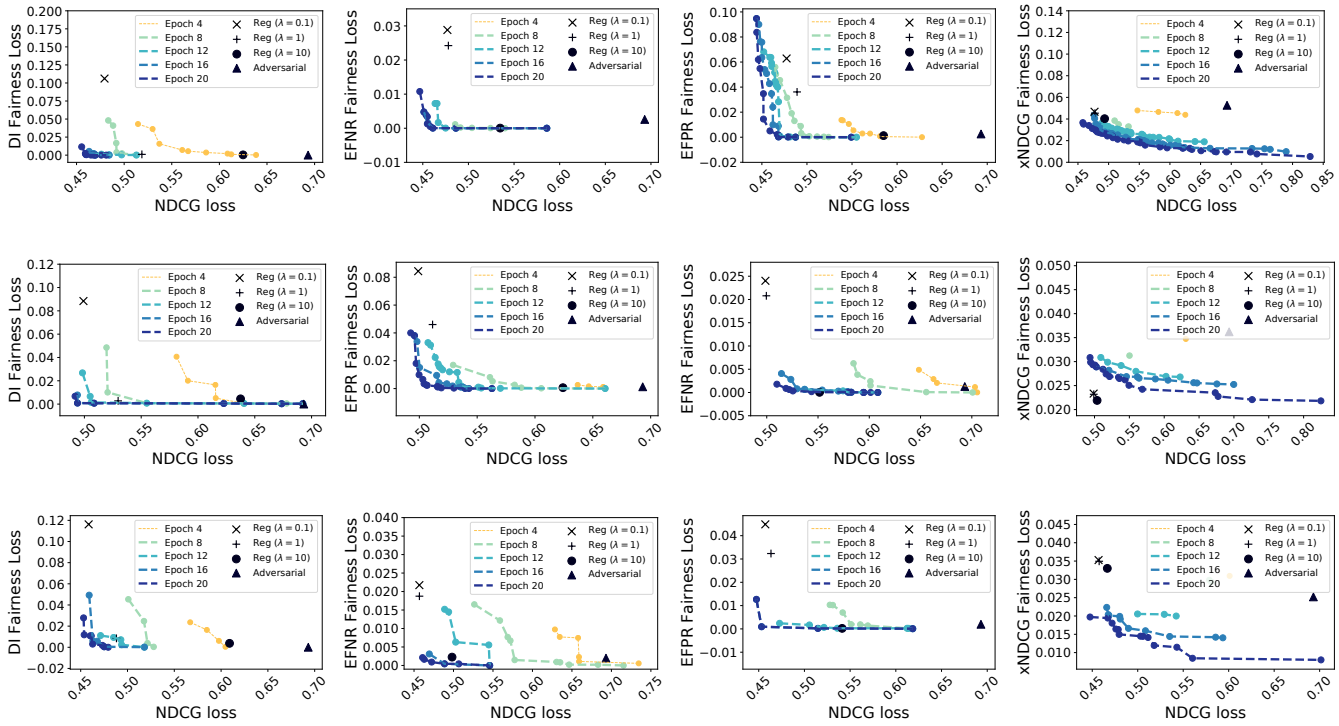


Figure 6: Convergence of Pareto fronts when running Algorithm 2. From top to bottom: YelpChi, YelpNYC, and YelpZip. We run the algorithm with NDCG loss (ℓ_1) and *one* of the four fairness losses (from left to right: ℓ^{DI}, ℓ^{EFNR}, ℓ^{EFPR}, and ℓ^{xNDCG}. The two baselines are compared and the fairness-regularized GNN has three different regularization hyperparameters ($\lambda = 0.1, 1, 10$).

and inference algorithms are proposed, respectively. In [5, 7, 36], they aimed to find a fair embedding of nodes on a graph. They assume sensitive attributes are available on the nodes to define the privileged and unprivileged groups, while we have node degree as the sensitive attribute. In [1, 9], the authors propose to train GNN using an adversarial opponent that tries to relate prediction or data representation to sensitive node attributes. In [31], adversarial objective function is added to a deep network so that the representation and the classification are both insensitive to the sensitive attribute. The advantage of these methods is that they are agnostic to the fairness criteria. As a result, however, the methods cannot optimize specific multiple objectives. Our MOO algorithms converge to Pareto optimal solutions that dominate the solutions found by adversarial fair learning. The targeted detection problem can be viewed as a ranking problem on bipartite graphs [4, 38]. The most relevant one promotes diversity and fairness [41], where a two-step and regularization approach was adopted.

Game theoretical fairness. The regularized optimization for achieving accuracy-fairness trade-offs has been proposed in [23]. An alternative formulation is to place the fairness regularization as a constraint of the optimization problem for model training [43, 49]. However, these methods only work with a single fairness metric and need to specify the strength of fairness regularization. Furthermore, these prior works did not explore the Pareto front consisting of multiple optimal trade-offs.

Multi-objective for fairness. This work is inspired by prior MOO works [12, 13, 39], which did not certificate and optimize multiple fairness criteria for GNN. There are work that address multiple fairness criteria using MOO [22, 32], however, their methods are preference-based and require user-specified relative objective importance. Our method uses stochastic search and is data-driven.

6 CONCLUSION

We studied the problem of meeting multiple fairness criteria when training a GNN to detect spams on a review graph. The challenge of certificating the compatibility of multiple fairness criteria is addressed by formulating linear systems in terms of graph structures and input features. When the certificate fails, it is then desirable to find Pareto efficient solutions, where improving one objective necessarily harm another. We propose algorithms with proof of convergence using the implicit function theorem to find Pareto optimal solutions, The proposed stochastic search is data-driven and without user-specified preference vectors. Our solutions dominate the baselines that use fairness-regularization and adversarial fair representation learning.

ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1931042, IIS-2008155, IIS-1909879 and CNS-1757787.

REFERENCES

[1] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. Towards a Unified Framework for Fair and Stable Graph Representation Learning. 2021.

[2] L. Dontchev Asen and R. Tyrrell Rockafellar. *Implicit Functions and Solution Mappings: A View from Variational Analysis*. 2014.

[3] R Berk, H Heidari, S Jabbari, Matthew Joseph, M Kearns, Jamie H Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression. *ArXiv*, abs/1706.02409, 2017.

[4] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. Fairness in recommendation ranking through pairwise comparisons. KDD '19.

[5] Avishek Joey Bose and William L. Hamilton. Compositional fairness constraints for graph embeddings. In *ICML*, 2019.

[6] Chris Burges, Tal Shaked, Erin Renshaw, A. Lazier, Matt Deeds, N. Hamilton, and Greg Hullender. Learning to rank using gradient descent. *Proceedings of the 22nd international conference on Machine learning*, 2005.

[7] Maarten Buyl and Tijl De Bie. Debayes: a bayesian method for debiasing network embeddings. *ArXiv*, abs/2002.11442, 2020.

[8] T. Calders and S. Verwer. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21:277–292, 2010.

[9] Enyan Dai and Suhang Wang. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. In *WSDM*, 2021.

[10] Golnoosh Farnadi, Behrouz Babaki, and Lise Getoor. Fairness in Relational Domains. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 108–114, New York, NY, USA, 2018.

[11] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and Removing Disparate Impact. KDD, 2015.

[12] J Fliege, A I F Vaz, and L N Vicente. Complexity of gradient descent for multiobjective optimization. *Optimization Methods and Software*, 34(5):949–959, 2019.

[13] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[14] C. Forman, A. Ghose, and B. Wiesenfeld. Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets. *ISR*, 2008.

[15] Sorelle A. Friedler, C. Scheidegger, and Suresh Venkatasubramanian. The (im)possibility of fairness. *Communications of the ACM*, 2016.

[16] Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying Real-world Goals with Dataset Constraints. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[17] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[18] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of Opportunity in Supervised Learning. NIPS'16, pages 3323–3331, 2016.

[19] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *KDD*, 2016.

[20] P. Kaghazgaran, M. Alfifi, and J. Caverlee. Wide-ranging review manipulation attacks: Model, empirical study, and countermeasures. In *CIKM*, 2019.

[21] P. Kaghazgaran, J. Caverlee, and A. Squicciarini. Combating crowdsourced review manipulators: A neighborhood-based approach. In *WSDM*, 2018.

[22] Mohammad Mahdi Kamani. *Multiobjective Optimization Approaches for Bias Mitigation in Machine Learning*. PhD thesis, PSU, 2020.

[23] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-Aware Classifier with Prejudice Remover Regularizer. In *Machine Learning and Knowledge Discovery in Databases*, pages 35–50, 2012.

[24] J. Kim, J. Chen, and Ameet Talwalkar. Fact: A diagnostic for group fairness trade-offs. In *ICML*, 2020.

[25] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[26] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. *ArXiv*, abs/1609.05807, 2017.

[27] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and VS Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *WSDM*, 2018.

[28] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. 2020.

[29] Michael Lohaus, Michael Perrot, and Ulrike Von Luxburg. Too Relaxed to Be Fair. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6360–6369. PMLR, 2020.

[30] M. Luca. Reviews, reputation, and revenue: The case of yelp. com. *HBS Working Paper*, 2016.

[31] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[32] Debabrata Mahapatra and Vaibhav Rajan. Multi-Task Learning with User Preferences: Gradient Descent with Controlled Ascent in Pareto Optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 6597–6607, 2020.

[33] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. Spotting opinion spammers using behavioral footprints. In *KDD*, 2013.

[34] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance. What yelp fake review filter might be doing? In *ICWSM*, 2013.

[35] M. Rahman, N. Hernandez, R. Recabarren, S. I. Ahmed, and B. Carbunar. The art and craft of fraudulent app promotion in google play. In *CCS*, 2019.

[36] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards Fair Graph Embedding. In *IJCAI*, 2019.

[37] S. Rayana and L. Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*, 2015.

[38] Ashudeep Singh and Thorsten Joachims. Fairness of Exposure in Rankings. In *KDD*, 2018.

[39] LIu Suyun and L N Vicente. Accuracy and Fairness Trade-offs in Machine Learning: A Stochastic Multi-Objective Approach. Technical report, 2020.

[40] J. Swearingen. Amazon Is Filled With Sketchy Reviews. Here's How to Spot Them, 2017.

[41] Lequn Wang and T. Joachims. Fairness and diversity for rankings in two-sided markets. *ICTIR*, 2021.

[42] X. Wang, K. Liu, and J. Zhao. Handling cold-start problem in review spam detection by jointly embedding texts and behaviors. In *ACL*, 2017.

[43] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. Learning Non-Discriminatory Predictors. In *CoLT*, 2017.

[44] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6861–6871, 2019.

[45] Yongkai Wu, Lu Zhang, and Xintao Wu. On convexity and bounds of fairness-aware classification. In *The World Wide Web Conference*, WWW '19, 2019.

[46] S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In *KDD*, 2012.

[47] Dou Yingtong, Guixiang Ma, Philip S. Yu, and Sihong Xie. Robust Detection of Adaptive Spammers by Nash Reinforcement Learning. In *KDD*, 2020.

[48] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness Beyond Disparate Treatment and Disparate Impact: Learning Classification Without Disparate Mistreatment. In *WWW*, 2017.

[49] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 2019.

[50] Yue Zhang and Arti Ramesh. Learning Fairness-aware Relational Structures. In *ECAI*, 2020.