Pushing the limits of fairness impossibility: Who's the fairest of them all?

Anonymous Author(s)
Affiliation
Address
email

Abstract

The impossibility theorem of fairness is a foundational result in the algorithmic fairness literature. It states that outside of special cases, one cannot exactly and simultaneously satisfy all three common and intuitive definitions of fairness - demographic parity, equalized odds, and predictive rate parity. This result has driven most works to focus on solutions for one or two of the metrics. Rather than follow suit, in this paper we present a framework that pushes the limits of the impossibility theorem in order to satisfy all three metrics to the best extent possible. We develop an integer-programming based approach that can yield a certifiably optimal post-processing method for simultaneously satisfying multiple fairness criteria under small violations. We show experiments demonstrating that our post-processor can improve fairness across the different definitions simultaneously with minimal model performance reduction. We also discuss applications of our framework for model selection and fairness explainability, thereby attempting to answer the question: *who's the fairest of them all?*

1 Introduction

While fairness in machine learning has received significant attention in recent years, most existing works focus on one of the many fairness criteria [3]. Consequently, practitioners are perhaps left with no choice but to use their best judgment to apply a single fairness criterion. We suspect that the conflicting nature of existing mathematical definitions of fairness might have led to this undesirable practice of narrowing down fairness-related measurement and mitigation to one chosen definition. This is somewhat analogous to the trade-off between precision and recall while evaluating model performance [7]. Instead of choosing one of precision or recall to evaluate the performance of a classification model, practitioners often evaluate the trade-off and choose models that can maintain a certain level of precision while optimizing recall or vice-versa [19]. In this paper, we provide a framework for explicitly addressing such trade-offs among multiple fairness criteria and model performance toward optimal model selection.

One of the most prolific examples of fairness in machine learning arose from the ProPublica recidivism study ([2]), in which a risk assessment tool called COMPAS was found to be biased against black defendants. But beyond the immediate implications in criminal justice, the study also prompted more general studies in algorithmic fairness and, in particular, led to a key result highlighted in [13], [21], and [25] which some colloquially refer to as the "impossibility theorem" in fairness [30, 24]. This theorem essentially states that three common definitions of algorithmic fairness - demographic parity [16], equalized odds [21], and predictive parity [3], cannot be simultaneously satisfied outside of pathological situations. Several works following these initial results have therefore focused on satisfying one metric ([21, 32]) or proposing adaptable methods for different metrics ([11, 37]). Yet, while we do not deny the conclusions of the impossibility theorem, we also believe there have not been sufficient efforts to reconcile the conflicting fairness definitions to the best extent possible. We

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

fill this gap in the literature by translating the trade-offs among multiple fairness criteria and model performance into a constrained optimization problem and propose a post-processing methodology for simultaneously achieving approximate fairness in the conflicting definitions simultaneously.

We believe our framework would alleviate the practitioner from making the hard choice in choosing a particular metric. Instead, if they have a partial ordering of importance amongst the metrics (which many possess), our framework would explicitly allow them evaluate such trade-offs. The application which we highlight in Section 4 discusses these in detail. Overall, we make three main contributions in this work:

1. We design a flexible optimization framework that returns a post-processing score transformation function that can make scores group-wise ϵ-fair along three definitions (demographic parity, equalized odds, and predictive rate parity) simultaneously. This framework can be applied to any binary classifier that produces a continuous score, can be configured for singular or multiple metrics of fairness, and also can account for fairness vs. performance trade-offs in terms of ROC-AUC.
2. We present a novel reformulation of this non-convex optimization problem as a Mixed Integer Linear Program (MILP) [41]. This reformulation allows us to find provably globally optimal solutions. We further show that in practice, we can consistently find better solutions through our global optimization method compared to local optimization methods in a reasonable time.
3. We discuss and extend our framework from a post-processing mechanism to a tool that can aid practitioners in better understanding their data and models' empirical fairness characteristics and trade-offs and compare these traits across models.

The rest of the paper is organized as follows. In Section 2 we mathematically define fairness metrics and the multiple fairness optimization problem. We discuss the optimal solution via MILP in Section 3. We try to answer the question of "who's the fairest of them all?" through our applications and experiments in Section 4. Finally, we conclude with a discussion in Section 5. We wrap up this section with a discussion of the related literature.

Related Work: Early works [13] exploring the conflicts between fairness definitions prove that for a binary predictor, predictive rate parity conflicts with equalized odds ([21]) unless base rates are equal or the model is perfectly predictive. Chouldechova [13] also considers trade-offs between the fairness definitions in the binary prediction case. Kleinberg et al. [25] generalizes this study by showing that statistical (i.e., demographic) parity is also inconsistent with predictive rate parity and equalized odds. The same paper studies these inconsistencies in a more general bin-wise prediction setting and shows that approximate fairness definitions (predictive rate parity, equalized odds) can simultaneously hold but only under ϵ-approximate equal base rates or ϵ-approximate perfect performance. This work further proves that there is an inherent trade-off between fairness and loss.

Beyond impossibility theorem results, several works have focused on trade-offs between fairness and model performance [21, 17, 37, 42]. They develop in-processing solutions aimed at reducing one metric while maintaining accuracy. A few authors have analyzed trade-offs or attempted to achieve multiple fairness. One example is Pleiss et al. [36], which shows that predictive rate parity and a relaxed form of equalized odds are reconcilable under a randomized prediction scheme. Another notable example is Celis et al. [11], which develops a flexible in-processing approach to achieve multiple types of fairness (potentially at the same time). Like Celis et al. [11], the framework we develop also aims for a flexible approach to focus on one or many fairness metrics simultaneously. However, our method is distinct in that it is a post-processing based solution and also more general as it works for continuous scores (rather than binary classification). Our work is most closely related to Nandy et al. [32] in terms of the underlying score transformation mechanism, and we leverage some of their methods. However, whereas [32] only targets equalized odds, we go further and include demographic parity and predictive rate parity in our framework—this posits computational challenges, which we address by proposing novel methodology based on integer programming.

As noted at the end of [25], some open questions are how to optimally assign scores to satisfy multiple criteria when base rates are equal and additionally, how to satisfy predictive rate parity and either equal TPR or TNR when one cost outweighs the other. To our knowledge, no prior work has attempted to reconcile all three fairness conditions (demographic parity, equalized odds, and predictive rate parity) simultaneously with model performance through a post-processing framework. Although the post-processing methods tend to be less flexible for fairness-performance trade-offs

than their in-processing counterparts, they can be much more easily added on top of any existing model training pipeline. This makes a post-processing approach modular, and in particular, more appealing in complex web-scale recommender systems that use (a combination of) certain prediction scores to rank a list of items.

2 Multiple Fairness Optimization

We consider a binary classification problem where the i-th observation is characterized by their label $y_i \in \{0, 1\}$, their group membership $g_i \in \mathcal{G}$, and model predicted probability (also known as a risk score [25]) $s_i \in [0, 1]$ for $i = 1, \ldots, N$. The corresponding random variables are denoted as Y, G and S respectively. To set up the problem, we discretize the scores into nonempty bins $b \in \mathcal{B} := \{1, \ldots, |\mathcal{B}|\}$ by using, for example, a quantile transformation (we will denote $|\mathcal{B}|$ as B). Additionally, let $N_{b+}^{[g]}$ denote the number of group g positive class ($y_i = 1$) instances in bin b, $N_b^{[g]}$ denote the total number of instances of group g instances in bin b, $N_+^{[g]}$ ($N_-^{[g]}$) be the total number of group g positive (negative) instances, and $N^{[g]}$ be the total number of group g instances. Lastly, our approach seeks to achieve fairness by moving instances from one bin to another bin: hence, we define variable $x_{bb'}^{[g]}$ as the probability of moving an instance of group attribute g and score in bin b into a new bin b' [1]. In other words, for every group g, the collection $\{x_{bb'}^{[g]}\}_{b,b'}$ can be represented as a $B \times B$ transition matrix (with additional constraints, as discussed next).

For the optimization framework and the remainder of this paper, we translate a single fairness definition (e.g. equal true positive rate) as a constraint that controls for the worst-case violations across all bins. Below, we discuss different fairness constraints that we consider in our framework, and describe how they can be represented in terms of the optimization variables $\{x_{bb'}^{[g]}\}$.

2.1 Fairness Constraints Under Binning Framework

Demographic Parity (DP): For simplicity, we assume that there are exactly two groups $g \in \{1, 2\}$ as we formulate the fairness metrics, though pairwise constraints can be added to enforce fairness for an arbitrary number of groups. Starting with demographic or statistical parity from [16], this condition states that the model's predicted score is independent of group membership. This is equivalent to

$$P(S = s \mid G = 1) = P(S = s \mid G = 2).$$

Our version of this constraint uses bins B to empirically approximate the probability $P(S = s \mid G = g)$ and we also relax the equality to an ϵ-approximate equality (for some pre-specified $\epsilon > 0$). Therefore, after transforming the scores using $\{x_{bb'}^{[g]}\}_{b,b'}$ the ϵ_{DP}-approximate DP can be expressed as the following as a linear constraint:

$$\left| \frac{1}{N^{[1]}} \sum_{b \in \mathcal{B}} x_{bb'}^{[1]} N_b^{[1]} - \frac{1}{N^{[2]}} \sum_{b \in \mathcal{B}} x_{bb'}^{[2]} N_b^{[2]} \right| \leq \epsilon_{DP} \qquad \forall \ b' \in \mathcal{B}, \tag{1}$$

where $N_b^{[g]}$ denote the number of observations from group g in bin b (before transformation), and $N^{[g]} = \sum_{b \in \mathcal{B}} N_b^{[g]}$. For reproducibility, ϵ_{DP} should be chosen to be larger than the approximation error $O(1/\sqrt{N^{[g]}})$ for replacing $P(S = s \mid G = g)$ with its empirical counterpart.

Equalized Odds (EOdds): The equalized odds condition for binary predictors given in Hardt et al. [21]. This is a balance condition where the groups must have equal true positive and false positive rates. For continuous scores, it translates to having equal score distributions for each group conditional on their true labels [32]:

$$P(S = s \mid Y = y, G = 1) = P(S = s \mid Y = y, G = 2) \qquad \text{for } y \in \{0, 1\}.$$

[1] In applications, we can discretize the scores into B bins with a quantile discretizer and consider how we can move them across bins. More bins allow for more granular interpretation of the transformed scores at the cost of us solving a harder problem and vice versa.

Like demographic parity, our empirical score bin version requires that the distribution of positive or negative instances be ϵ_{EOdds}-approximately equal between groups in the new bins b'. Both equal true positive rate and false positive rate can be expressed as linear constraints, respectively:

$$\begin{aligned} \left| \frac{1}{N_+^{[1]}} \sum_{b \in \mathcal{B}} x_{bb'}^{[1]} N_{b+}^{[1]} - \frac{1}{N_+^{[2]}} \sum_{b \in \mathcal{B}} x_{bb'}^{[2]} N_{b+}^{[2]} \right| &\leq \epsilon_{EOdds} \ \ \forall \ \ b' \in \mathcal{B} \\ \left| \frac{1}{N_-^{[1]}} \sum_{b \in \mathcal{B}} x_{bb'}^{[1]} N_{b-}^{[1]} - \frac{1}{N_-^{[2]}} \sum_{b \in \mathcal{B}} x_{bb'}^{[2]} N_{b-}^{[2]} \right| &\leq \epsilon_{EOdds} \ \ \forall \ \ b' \in \mathcal{B}. \end{aligned} \tag{2}$$

Predictive Rate Parity (PRP): Lastly, we examine the predictive rate parity condition popularized in [13]. This condition states that the probability of being a positive instance is independent of group membership when we condition on the score. Formally:

$$P(Y = 1 \mid S = s, G = 1) = P(Y = 1 \mid S = s, G = 2).$$

Using the empirical score bin framework, an approximate version of the above implies that the proportion of positive instances in each bin must be ϵ_{PRP}-approximately equal among groups:

$$\left| \frac{\sum_{b \in \mathcal{B}} x_{bb'}^{[1]} N_{b+}^{[1]}}{\sum_{b \in \mathcal{B}} x_{bb'}^{[1]} N_{b}^{[1]}} - \frac{\sum_{b \in \mathcal{B}} x_{bb'}^{[2]} N_{b+}^{[2]}}{\sum_{b \in \mathcal{B}} x_{bb'}^{[2]} N_{b}^{[2]}} \right| \leq \epsilon_{PRP} \quad \forall \ \ b' \in \mathcal{B}. \tag{3}$$

Unlike constraints (1) and (2), which can be expressed as a linear function of the optimization variables $\{x_{bb'}^{[g]}\}$, condition (3) yields bilinear terms and is in general a non-convex constraint. A main technical difficulty of our framework arises from this non-convex fairness constraint—Section 3 presents an integer programming framework to handle this non-convexity, ensuring we can obtain a globally optimal solution to the resulting optimization problem.

Remark. Our definition of fairness as the worst-case violation across all bins aims to resemble approximations of the respective probabilistic definitions but we have not found identical definitions in other works. We comment on the differences and discuss why it does not contradict the traditional impossibility theorem of [25] in Appendix E.

2.2 MFOpt: Multiple Fairness Optimization Framework

We use constraints developed in Section 2.1 to state the multiple fairness optimization (MFOpt) problem:

$$\begin{aligned} \underset{\{x_{bb'}^{[g]}\}_{b,b',g}}{\text{minimize}} \quad & \sum_{g \in \mathcal{G}} \sum_{b \in \mathcal{B}} \sum_{b' \in \mathcal{B}} \left| \frac{N_b^{[g]}}{N} (\bar{s}_b - \bar{s}_{b'}) x_{bb'}^{[g]} \right| && \text{(4a)} \\ \text{s.t.} \quad & \sum_{b \in \mathcal{B}} x_{bb'}^{[g]} = 1 \qquad \forall \ \ b' \in \mathcal{B}, \ g \in \mathcal{G} && \text{(4b)} \\ & x_{bb}^{[g]} \geq 1 - \xi \qquad \forall \ \ b \in b' \in \mathcal{B}, \ g \in \mathcal{G} && \text{(4c)} \\ & x_{bb'}^{[g]} = 0 \qquad \forall \ \ b' \ s.t. \ |b' - b| \geq w, \ \forall g \in \mathcal{G} && \text{(4d)} \\ & \text{Fairness Constraints:} \quad (1), (2), (3) && \text{(4e)} \\ & \frac{\sum_{b \in \mathcal{B}} x_{bb'}^{[g]} N_{b+}^{[g]}}{\sum_{b \in \mathcal{B}} x_{bb'}^{[g]} N_{b}^{[g]}} \leq \frac{\sum_{b \in \mathcal{B}} x_{b(b'+1)}^{[g]} N_{b+}^{[g]}}{\sum_{b \in \mathcal{B}} x_{b(b'+1)}^{[g]} N_{b}^{[g]}} \qquad \forall \ \ b' \in \{1, ..., B-1\}, \ g \in \mathcal{G} && \text{(4f)} \\ & 0 \leq x_{bb'}^{[g]} \leq 1, \ \ \forall \ \ b, b', g. && \text{(4g)} \end{aligned}$$

Above, the optimization variables are the group-specific movement probability $x_{bb'}^{[g]}$ terms and all remaining terms are problem data and/or configurable hyperparameters. Starting with (4a), we define $\bar{s}_b$ as the midpoint score in the bin and hence the objective is the product of the movement distance $\bar{s}_b - \bar{s}_{b'}$ weighed by the fraction of total samples moved $N_b^{[g]}/N$ and the amount of movement $x_{bb'}^{[g]}$. (4b) States that the total movement out of bin b, including the movement back to itself, must sum up to 1 and along with (4g) ensures that $x_{bb'}^{[g]}$ represent probabilities in a transition matrix. (4c) states that the

total movement from bin b back to itself must be lower bounded by hyperparameter ξ. This parameter controls how far we allow the new scores to stray from the original and is necessary to prevent zero denominators in 3 and 4f. Constraint (4d) represents window constraints to restrict extreme movements of scores beyond w bins. Constraint (4e) are the fairness constraints in Section 2.1. (4f) ensures that we preserve the rank-ordering of the scores and expected values, which is desirable for comparing bins against each other. The predictive rate parity constraint (3) and (4f) both introduce non-convexities into Problem (4). These two constraints also require us to assume overlap, or that each bin contains at least one member from each group. Without overlap, predictive rate parity is undefined since it is not possible to compare expectations across groups for a given bin and demographic parity is likely violated as it means one group has 0 probability of landing in a given bin.

We close this section with two major benefits of this framework compared to inprocessing solutions such as adding fairness regularization ([42], [17], [37],) or using an entirely different fairness-based model ([11], [43]). First, the number of optimization variables scales in the order of $\mathcal{O}(|\mathcal{G}||\mathcal{B}|^2)$. This is significant as it entails that all of our solutions (solving the optimization once, or over a grid of settings for applications in Section 4) can be applied in arbitrarily large settings as long as score-based binning is possible. Applying the transformation is equally tractable, as it only requires binning observations and making independent draws from a multinomial distribution with B possible outcomes.

The second benefit of our framework is that it returns a highly interpretable solution as it returns one optimized $B \times B$ transition matrix per group. Hence given a newly scored instance, several facts can be easily read from the corresponding row of the matrix such as the likelihood of moving to a specific bin b, moving into any higher or lower bin, etc. These probabilities can also be finely controlled via constraints in the optimization as we demonstrate with the window constraints (4d) and max movement constraints (4c). This interpretability is not present in model regularization frameworks, where it is hard to gauge the amount of regularization needed to achieve a certain fairness effect and also difficult to know how an individual's score might change when switching from the base model to a fair model.

3 Finding Optimal Solutions via Mixed Integer Programming (MIP)

The primary difficulty of the above optimization problem are the predictive rate parity constraint (3) and rank-order constraint (4f) which turn the problem non-convex. Non-convex constrained optimization is generally NP-hard and traditional methods that seek locally optimal solutions include gradient-based interior point optimization ([40]), sequential quadratic programming ([18]), or algorithms specific to quadratically-constrainted-quadratic-programs (QCQPs) such as operator splitting methods, semidefinite relaxations, etc. (see [33] for an overview).

Rather than pursuing a locally optimal solution, we propose a novel reformulation of the problem into a tractable mixed-integer-linear-program (MILP) which can be solved to global optimality ([10], [9]). Our reformulation grants two benefits over traditional locally optimal solvers. First, global strategies theoretically enable us to find the best possible solution. Second, since our problem primarily scales with the number of bins, it is computationally tractable and allows us to utilize the power of modern MIP solvers.

3.1 Tractable reformulations for computational efficiency

We first observe that a direct reformulation of the fractional terms into bilinear terms in constraints (3) and (4f) will lead to bilinear terms in the order of $\mathcal{O}(B^3)$. We show that we can reduce this to $\mathcal{O}(B)$ bilinear terms through a substitution that exploits the problem structure. Next, we take advantage of the vastly reduced number of bilinear terms to apply the normalized multiparametric disaggregation technique (NMDT, [1]) which we explain in Section 3.1.2. This allows us to approximate products of continuous variables as products of integer variables, which can be easily linearized and handled by MIP solvers. Importantly, this transformation of xy terms requires upper and lower bounds for x and y and we propose a method in 3.1.2 for generating and tightening these bounds by solving fractional linear program subproblems ([12]). Taken together, the reduction of bilinear terms combined with the bound-tightening procedure enable us to effectively apply the NMDT methodology and transform the problem from a non-convex QCQP to an MILP that can be solved to global optimality.

3.1.1 Step 1: Reducing number of bilinear variables

We reduce the number of bilinear variables in our problem by making a substitution for the fraction term by introducing new optimization variables $v_b^{[g]} \geq 0$ (to represent a sum) and $t_b^{[g]} \geq 0$ (to represent the fractional quantity) as additional variables (see Appendix C for more details). We can then use them to write equivalent constraints with only $\mathcal{O}(B)$ bilinear terms. Let,

$$v_{b'}^{[g]} = \sum_{b \in \mathcal{B}} x_{bb'}^{[g]} N_b^{[g]} \qquad \text{and} \qquad t_{b'}^{[g]} v_{b'}^{[g]} = \sum_{b \in \mathcal{B}} x_{bb'}^{[g]} N_{b+}^{[g]} \qquad \forall \ b' \in \mathcal{B}.$$

Then we have the following:

$$\text{Constraint (3)} \iff \left| t_{b'}^{[1]} - t_{b'}^{[2]} \right| \leq \epsilon_{PRP} \ \ \forall \ \ b' \in \mathcal{B},$$

$$\text{Constraint (4f)} \iff t_{b'}^{[g]} \leq t_{b'+1}^{[g]} \ \ \forall \ \ b' \in \{1, \ldots, B-1\}$$

3.1.2 Step 2: NMDT and bound tightening through fractional LP subproblems

Linearizing bilinear terms (NMDT): We show how we can model each bilinear term $t_b^{[g]} v_b^{[g]}$ by using a binary expansion for the continuous variables $t_b^{[g]}, v_b^{[g]}$, and by observing that the product of binary variables can be modeled via integer programming (see [27, 39] for reference). To this end, we make use of the NMDT transformation [1]. We recap this method below as formulated in [1].[2] Given any bounded optimization variable $x \in [x_L, x_U]$, and precision factor p, a negative integer, we can represent this variable exactly as $x = (x_U - x_L)\lambda + x_L$ where

$$\lambda = \sum_{l \in \{-p, \ldots, -1\}} 2^l z_l + \Delta\lambda$$

where $0 \leq \Delta\lambda \leq 2^p$ is a remainder term and $z_l \in \{0, 1\}$ are binary optimization variables. Dropping the remainder term $\Delta\lambda$ gives us the approximate form and product forms of xy become dot products of several integer variables, which can be effectively handled via modern MIP solvers, such as [20]. Although we are solving an approximation (e.g. precision of $1e^{-4}$) this is not a practical problem since it is precise enough for reasonable choices of ϵ and we do not expect the constraints to hold exactly when we apply the post-processor on the testing data anyways.

Bounds on $v_b^{[g]}, t_b^{[g]}$ via Fractional LPs: A key requirement to apply NMDT is that all optimization variables in the bilinear terms ($t_b^{[g]}$ and $v_b^{[g]}$ in our case) must be bounded and we need to be able to accurately estimate these bounds (i.e., tighter bounds leads to faster runtimes [1]). We propose obtaining these lower and upper by minimizing and maximizing respectively, the sub-problems while keeping all fairness constraints except the quadratic constraints. Firstly, note that bounds on $v_b^{[g]}$ can be solved as a simple LP (details in Appendix C). Meanwhile, obtaining a bound on $t_b^{[g]}$ requires solving a nonlinear problem due to fact that $t_b^{[g]}$ represents a fractional objective (ratio of two affine terms of optimization variables). However, we observe that we can apply the Charnes-Cooper transformation [12] to reformulate the nonlinear problem into a simple LP (details in Appendix C).

3.2 Choice of algorithm: QCQP (heuristic) vs MIP (optimal solution)

In this section we show the results and benefits of our reformulation from a non-convex QCQP to an MILP. In Table 1, we take each dataset and train a grid-searched random forest model and score the training data. Next, we discretize the scores into bins, parameterize the problem (# bins, ϵ, max movement, window size, solve time) and compare our MILP solution solved by Gurobi ([20]) against the QCQP problem solved by IPOPT ([28]), which is a generic interior-point log-barrier penalty method for nonlinear constrained optimization. We discuss the datasets and problem parameters for all experiments in the Appendix D. For each metric such as AUC, we use AUC_{INT} and AUC_{IP} to denote the average result of applying the interior-point (INT, for short) or integer programming (IP) method, respectively. The metrics used are the objective value, optimality gap ($\%\Delta$),[3] and

[2] The open-source implementation can be found in `https://github.com/joaquimg/QuadraticToBinary.jl` (MIT License) which we utilize.

[3] The optimality gap is defined $\%\Delta = \frac{UpperBound - LowerBound}{UpperBound}$ where the upper bound is the best feasible solution and the lower bound is produced by the branch-and-bound method. The INT method does not have the benefit of providing lower bounds, hence we use the bound produced by the IP method to compute this.

AUC[4]. We also report the statistical significance of the improvement based the p-value from the Wilcoxon signed-rank test to determine if $\%\Delta_{IP} \leq \%\Delta_{INT}$ is a consistent result. Bold figures indicate statistical significance w.r.t. 1 standard deviation[5].

Table 1: Interior Point Solution vs. MIP Solution

Dataset	Obj_{INT}	Obj_{IP}	$\%\Delta_{INT}$	$\%\Delta_{IP}$	p-value	AUC_{INT}	AUC_{IP}
ACS Income	2.0809	1.9682	15.076 ± 6.461	10.621 ± 3.402	**0.0029**	0.9041	0.9044
ACS Insurance	0.9769	0.9599	3.432 ± 0.225	**1.715 ± 0.169**	**0.0010**	0.7411	0.7413
ACS Mobility	2.4580	2.3781	5.37 ± 0.803	**2.193 ± 0.138**	**0.0010**	0.7971	0.7973
ACS Poverty	2.0693	2.0526	3.756 ± 0.435	**2.972 ± 0.324**	**0.0010**	0.8440	0.8440
ACS Coverage	8.9361	1.9665	79.711 ± 0.782	**7.878 ± 2.207**	**0.0010**	0.5420	0.8149
ACS Travel	2.3935	2.3859	2.554 ± 0.254	2.242 ± 0.28	**0.0010**	0.7725	0.7725
Heart Disease	1.8871	1.3035	26.385 ± 17.401	**3.81 ± 0.864**	**0.0010**	0.8302	0.8629
COMPAS	7.4551	3.1300	62.88 ± 13.407	**17.055 ± 7.482**	**0.0010**	0.5143	0.7378

As the results show, the MIP reformulation consistently beats the interior point solver applied on the raw optimization problem, even when solving for only 10 minutes. We also observe that in most cases, regardless of the method we choose, we can quickly find near optimal solutions that are high performing in the sense of keeping an AUC close to the original model. This is a significant result as it means that even a locally optimal solution to our optimization problem can yield a practically useful post-processing result.

We conclude this section by reiterating two benefits of the reformulation. First, solving a MIP method yields lower bounds that can be used to prove optimality or otherwise gauge the quality of a feasible solution. Second, by framing the problem as a MIP, we can always theoretically continue improving the solution to optimality based on the acceptable time limits.

4 Applications

In this section, we illustrate a few methods of applying our framework and how it can be used to help model developers select and understand models from a fairness perspective. To apply these procedures, we first require developing an efficient frontier of fairness solutions to understand which ϵ configurations are feasible for a given model type and dataset. To generate this frontier, we solve the problem over a grid of parameters $\epsilon_{DP}, \epsilon_{EOdds}, \epsilon_{PRP}$. Each feasible solution will yield a point $s \in \{(AUC, \epsilon_{DP}, \epsilon_{EOdds}, \epsilon_{PRP})\}$ and the collection of non-dominated points from the solution set yields a efficient frontier. We show the 2-d profile shots of our 4-d fairness surface in Figure 1 as an example, where the color gradient represents AUC. In theory, we could obtain a true Pareto-optimal frontier since we have devised a method of obtaining globally optimal solutions. However, we generate this frontier using IPOPT due to practical limitations as we are solving a $7 \times 7 \times 7$ grid of ϵ parameters.

4.1 Understanding fairness tradeoffs

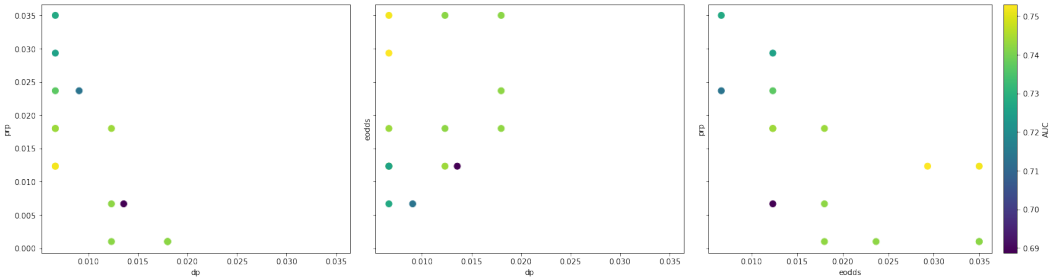
After the Pareto frontier is generated, the modeler can pick an operating point s based on the desired AUC and tolerable fairness violations ϵ. However, when communicating fairness properties to stakeholders and accounting for potential changes in strategy, it can be helpful to additionally understand the cost of further increasing fairness in terms of $\epsilon_{DP}, \epsilon_{EOdds}, \epsilon_{PRP}$.

This tradeoff can be understood by looking at the characteristics of points on the frontier near the operating point. Suppose we are at an operating point s. If we want to trade AUC for ϵ_{EOdds}, then we would find a point s' with at least as good $\epsilon_{DP}, \epsilon_{PRP}$ but worse AUC and better ϵ_{EOdds}. More generally, if we pick trade performance/fairness characteristic c (cost) for characteristic b (benefit),

[4] We describe how we computed the AUC for the bin-wise probabilities in Appendix B.

[5] We are limited in the number of trials we can run and actively chose to prioritize the variety of datasets we apply on rather than a large number of trials for a single dataset. As such, we expect relatively large standard errors but we reflect the consistency of our method through the p-value.

Figure 1: Efficient frontier of solutions for ACS West Insurance data



then we hold all other factors constant and find a point with better b and worse c. We illustrate this in the Table 2. The first row shows a hypothetical operating point while the following rows show other points on the efficient frontier that we could move to when we make a certain trade. Blank rows indicate that no such point was found that permits the desired trade-off. This trade-off perspective can enable developers to better understand and communicate the costs to performance or other fairness metrics when trying to close the disparity in one fairness metric.

Table 2: Performance Fairness Trade-off Analysis

Trade...	For...	s^*_{AUC}	$s^*_{\epsilon_{DP}}$	$s^*_{\epsilon_{EOdds}}$	$s^*_{\epsilon_{PRP}}$
Base	Base	0.7434	0.0123	0.018	0.0123
AUC	ϵ_{DP}	-	-	-	-
AUC	ϵ_{Eodds}	-	-	-	-
AUC	ϵ_{PRP}	0.7422	0.0123	0.0180	0.0067
ϵ_{DP}	ϵ_{PRP}	0.7436	0.0152	0.0123	0.0067
ϵ_{Eodds}	ϵ_{PRP}	-	-	-	-
ϵ_{Eodds}	ϵ_{DP}	0.7436	0.0152	0.0123	0.006

4.2 Performance Comparison

Lastly, we compare our framework against two methods and show that we can satisfy fairness constraint(s) just as well if not better, while yielding significantly better performance. When comparing against these methods, we first create 20 random splits of the dataset, perform some basic preprocessing, fit a grid-searched random forest model, and score the training and testing data to get the base scores $\hat{y}_0$. Next, we run the methodology that we are comparing against (i.e. build a model or apply the postprocessor) to get method scores $\hat{y}_m$. We then bin the outputs of the base model and compared method and compute the AUC along with the fairness metrics (ϵ_0, ϵ_m). Next, we solve our constrained optimization problem where we set the parameters ϵ to $\frac{1}{2}min(\epsilon_0, \epsilon_m)$. After optimizing, we can apply the optimal solutions $x^{[g]}_{bb'}$ to assign new bins for scored test instances. One method of assigning a group g instance with score $s \in b$ (denote as $s^{[g]}_b$) would be to make a random draw from a multinomial distribution parameterized by probabilities $(x^{[g]}_{b1}, x^{[g]}_{b2}, \ldots, x^{[g]}_{b|\mathcal{B}|})$. This is a coarse method of stochastic assignment and we also propose an alternative method in the Appendix A. Finally, we can compute the resulting AUC and fairness metrics on remapped bins for the testing data (and do the same for $\hat{y}_0$ and $\hat{y}_m$ on the testing data). These figures are shown in Table 3. We only show the results on the test set due to space constraints and have placed the results for the training set in Appendix D. Bold figures indicate that a metric is statistically significant to 1 standard deviation.

First, we compare our framework against the in-processing framework in Rezaei et al. [37], which is a robust optimization-based logistic regression model for reducing equality of opportunity violation. We found that this method works well compared to standard logistic regression and managed to decrease fairness violations while maintaining the similar performance. It improves in ϵ_{Eodds} compared to a random forest as well. However, its weakness is that the underlying model is still logistic regression and therefore has limited expressiveness. In comparison, MFOpt can be applied on top of any model class and thus the performance advantages of more flexible models are better maintained.

Table 3: Comparison with other fairness methods

Method	Metric	Base	Test Method	MF-Opt
Rezaei	AUC	0.7471 ± 0.003	0.6619 ± 0.0022	0.747 ± 0.003
	ϵ_{DP}	0.0117 ± 0.0014	0.0124 ± 0.0013	**0.0088 ± 0.001**
	ϵ_{EOdds}	0.0266 ± 0.007	0.0291 ± 0.0059	**0.0167 ± 0.0029**
	ϵ_{PRP}	0.109 ± 0.0145	0.1091 ± 0.0143	0.0986 ± 0.0133
Pleiss	AUC	0.8319 ± 0.0033	0.8149 ± 0.0087	0.831 ± 0.0032
	ϵ_{DP}	0.0212 ± 0.0016	0.0137 ± 0.0016	**0.0106 ± 0.0011**
	ϵ_{EOdds}	0.0329 ± 0.0042	0.023 ± 0.0038	**0.0142 ± 0.0028**
	ϵ_{PRP}	0.1465 ± 0.0178	0.4147 ± 0.1537	0.1547 ± 0.0293

Next, we compare our framework against the post-processing framework in Pleiss et al. [36]. In this method, the authors use randomization to maintain the model's calibration while simultaneously satisfying a relaxed equalized odds condition (whereby a linear combination of TPR and TNR are satisfied). Again, we see that this method maintains close performance as the base model, successfully shrinks equalized odds violations, and even decreases demographic parity violations too. However, it results in large violations of predictive rate parity based on our definition in contrast to our method.

4.3 Who's the fairest of them all?

In industry, machine learning model selection is guided by many factors including performance, speed, interpretability, among others. Yet, the fairness dimension is commonly overlooked unless the developers specifically induce it in their model. Even then, picking a specific fairness metric to optimize for can be a nebulous task. Rather than focus on a single metric, we propose a simple and intuitive method of gauging a model's efficiency in trading between different fairness definitions.

To do so, we propose constructing the frontier for the two models in question and then filtering all points on the efficient frontier with tolerable performance $AUC \geq AUC_{min}$. Next, find the point on the respective frontiers with minimum Euclidean distance to the origin. The model with the shortest distance to their frontier can then be declared as the model that has better tradeoff properties. We do not show an example due to lack of space, but remark that this procedure along with the trade-off analysis can be useful when a developer is iterating between models that were not designed for fairness, but wants a model that can be flexibly made more fair through the postprocessing framework that we propose. As fairness requirements may change over time, the model that can yield the best tradeoffs between different definitions can offer the most overall utility.

5 Discussion

In our study, we have devised a flexible, tractable, and interpretable post-processing method for making any binary classifier more fair. We then apply our methodology to push the limits of the impossibility theorem and show that while theoretical limitations remain undisputed, there is a path forward to practically reconciling the conflicting fairness definitions. Interestingly, our results extend the findings of [38], which finds that the trade-off of fairness and accuracy are negligible in practice. Our work reinforces this claim but also adds on that trade-offs between fairness definitions can be negligible as well. On the technical end, there are other optimization methodologies for solving the problem to global optimality such as spatial branch and bound. We considered these solutions during our investigation but ultimately opted for a MIP approach due to the maturity and availability of solvers (see Appendix F for details). Another potential avenue of further research is to improve the consistency of the PRP violation reduction, as we observed the largest standard error in reducing this metric. This could be due to us using random forests for all experiments, which is known to be an uncalibrated model [6]. One method to address this is therefore first calibrating the model through other methodologies such as Platt scaling ([35]) or binning-based calibration ([26]), as MFOpt appears to have no issue at least maintaining the level of calibration. We could also consider incorporating uncertainty in the training data through principled approaches such as stochastic or robust optimization.

References

[1] T. Andrade, F. Oliveira, S. Hamacher, and A. Eberhard. Enhancing the normalized multiparametric disaggregation technique for mixed-integer quadratic programming. *J. of Global Optimization*, 73(4):701–722, apr 2019. ISSN 0925-5001. doi: 10.1007/s10898-018-0728-9. URL https://doi.org/10.1007/s10898-018-0728-9.

[2] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias, May 2016. URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[3] S. Barocas, M. Hardt, and A. Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. http://www.fairmlbook.org.

[4] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, Oct. 2018. URL https://arxiv.org/abs/1810.01943.

[5] K. Bestuzheva, M. Besançon, W.-K. Chen, A. Chmiela, T. Donkiewicz, J. van Doornmalen, L. Eifler, O. Gaul, G. Gamrath, A. Gleixner, L. Gottwald, C. Graczyk, K. Halbig, A. Hoen, C. Hojny, R. van der Hulst, T. Koch, M. Lübbecke, S. J. Maher, F. Matter, E. Mühmer, B. Müller, M. E. Pfetsch, D. Rehfeldt, S. Schlein, F. Schlösser, F. Serrano, Y. Shinano, B. Sofranac, M. Turner, S. Vigerske, F. Wegscheider, P. Wellner, D. Weninger, and J. Witzig. The SCIP Optimization Suite 8.0. Technical report, Optimization Online, December 2021. URL http://www.optimization-online.org/DB_HTML/2021/12/8728.html.

[6] H. Boström. Calibrating random forests. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 121–126, 2008. doi: 10.1109/ICMLA.2008.107.

[7] M. Buckland and F. Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[8] S. Burer and A. Saxena. The milp road to miqcp. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 373–405, New York, NY, 2012. Springer New York. ISBN 978-1-4614-1927-3.

[9] P. M. Castro. Tightening piecewise mccormick relaxations for bilinear problems. *Computers & Chemical Engineering*, 72:300–311, 2015. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2014.03.025. URL https://www.sciencedirect.com/science/article/pii/S0098135414001069. A Tribute to Ignacio E. Grossmann.

[10] P. M. Castro. Normalized multiparametric disaggregation: An efficient relaxation for mixed-integer bilinear problems. *J. of Global Optimization*, 64(4):765–784, apr 2016. ISSN 0925-5001. doi: 10.1007/s10898-015-0342-z. URL https://doi.org/10.1007/s10898-015-0342-z.

[11] L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. FAT* '19, page 319–328, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287586. URL https://doi.org/10.1145/3287560.3287586.

[12] A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, 9(3-4):181–186, 1962. doi: https://doi.org/10.1002/nav.3800090303. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800090303.

[13] A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. doi: 10.1089/big.2016.0047. URL https://doi.org/10.1089/big.2016.0047.

[14] F. Ding, M. Hardt, J. Miller, and L. Schmidt. Retiring adult: New datasets for fair machine learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6478–6490. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/32e54441e6382a7fbacbbbaf3c450059-Paper.pdf.

[15] I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017. doi: 10.1137/15M1020575.

[16] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[17] B. Fish, J. Kun, and Ádám D. Lelkes. *A Confidence-Based Approach for Balancing Fairness and Accuracy*, pages 144–152. doi: 10.1137/1.9781611974348.17. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611974348.17.

[18] P. E. Gill and E. Wong. Sequential quadratic programming methods. 2012.

[19] M. Gordon and M. Kochen. Recall-precision trade-off: A derivation. *Journal of the American Society for Information Science*, 40(3):145–151, 1989.

[20] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022. URL https://www.gurobi.com.

[21] M. Hardt, E. Price, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3315–3323. Curran Associates, Inc., 2016.

[22] Q. Huangfu and J. A. J. Hall. Parallelizing the dual revised simplex method, 2015. URL https://arxiv.org/abs/1503.01889.

[23] A. Janosi, W. Steinbrunn, M. Pfisterer, R. Detrano, and D. Aha. Adult dataset, 1996. URL https://archive.ics.uci.edu/ml/datasets/adult.

[24] K. Karthik. The impossibility theorem of machine fairness - A causal perspective. *CoRR*, abs/2007.06024, 2020. URL https://arxiv.org/abs/2007.06024.

[25] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent Trade-Offs in the Fair Determination of Risk Scores. In C. H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:23, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-029-3. doi: 10.4230/LIPIcs.ITCS.2017.43. URL http://drops.dagstuhl.de/opus/volltexte/2017/8156.

[26] A. Kumar, P. S. Liang, and T. Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, 2019.

[27] H.-L. Li and C.-T. Chang. An approximate approach of global optimization for polynomial programming problems. In *European Journal of Operations Research*, pages 625–632. European Journal of Operations Research Volume 107, 1996.

[28] R. Lougee-Heimer. The common optimization interface for operations research: Promoting open-source software in the operations research community. *IBM Journal of Research and Development*, 47(1):57–66, 2003. doi: 10.1147/rd.471.0057.

[29] G. P. Mccormick. Computability of global solutions to factorable nonconvex programs: Part i – convex underestimating problems. *Math. Program.*, 10(1):147–175, dec 1976. ISSN 0025-5610. doi: 10.1007/BF01580665. URL https://doi.org/10.1007/BF01580665.

[30] T. Miconi. The impossibility of "fairness": a generalized impossibility result for decisions, 2017. URL https://arxiv.org/abs/1707.01195.

[31] H. Nagarajan, M. Lu, E. Yamangil, and R. Bent. Tightening mccormick relaxations for nonlinear programs via dynamic multivariate partitioning. In M. Rueher, editor, *Principles and Practice of Constraint Programming*, pages 369–387, Cham, 2016. Springer International Publishing. ISBN 978-3-319-44953-1.

[32] P. Nandy, C. Diciccio, D. Venugopalan, H. Logan, K. Basu, and N. E. Karoui. Achieving fairness via post-processing in web-scale recommender systems, 2020. URL https://arxiv.org/abs/2006.11350.

[33] J. Park and S. Boyd. General heuristics for nonconvex quadratically constrained quadratic programming, 2017. URL https://arxiv.org/abs/1703.07870.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[36] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5680–5689, 2017.

[37] A. Rezaei, R. Fathony, O. Memarrast, and B. Ziebart. Fairness for robust log loss classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5511–5518, apr 2020. doi: 10.1609/aaai.v34i04.6002. URL https://doi.org/10.1609%2Faaai.v34i04.6002.

[38] K. T. Rodolfa, H. Lamba, and R. Ghani. Empirical observation of negligible fairness–accuracy trade-offs in machine learning for public policy. *Nature Machine Intelligence*, 3(10):896–904, oct 2021. doi: 10.1038/s42256-021-00396-x. URL https://doi.org/10.1038%2Fs42256-021-00396-x.

[39] J. P. Teles, P. M. Castro, and H. A. Matos. Univariate parameterization for global optimization of mixed-integer polynomial problems. *European Journal of Operational Research*, 229(3): 613–625, 2013. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2013.03.042. URL https://www.sciencedirect.com/science/article/pii/S0377221713002750.

[40] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[41] L. A. Wolsey and G. L. Nemhauser. *Integer and combinatorial optimization*, volume 55. John Wiley & Sons, 1999.

[42] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi. Fairness Constraints: Mechanisms for Fair Classification. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 962–970. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/zafar17a.html.

[43] B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. AIES '18, page 335–340, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360128. doi: 10.1145/3278721.3278779. URL https://doi.org/10.1145/3278721.3278779.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In the discussion 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In Section D
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] We do not derive any theoretical results and all claims (e.g. NP-hardness of non-convex programs) are either well-known or a source is referenced.
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code will be uploaded within the week of paper submission as permitted by NeurIPS submission guidelines. Instructions and data used in all experiments are clearly outlined in D and 4
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Referenced in D
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Referenced in D
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] License and references can be found in the Appendix Section D
   (b) Did you mention the license of the assets? [Yes] License are mentioned alongside open-source package references
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Under the Experimental Data Section D
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]