ABSTRACT
We show how to take a regression function \( \hat{f} \) that is appropriately multicalibrated and efficiently post-process it into an approximately error minimizing classifier satisfying a large variety of fairness constraints. The post-processing requires no labeled data, and only a modest amount of unlabeled data and computation. The computational and sample complexity requirements of computing \( \hat{f} \) are comparable to the requirements for solving a single fair learning task optimally, but it can in fact be used to solve many different downstream fairness-constrained learning problems efficiently. Our post-processing method easily handles intersecting groups, generalizing prior work on post-processing regression functions to satisfy fairness constraints that only applied to disjoint groups. Our work extends recent work showing that multicalibrated regression functions are omnipredictors (i.e. can be post-processed to optimally solve unconstrained ERM problems) to constrained optimization problems.

1 INTRODUCTION
A now common technical framing of fair machine learning is that of constrained optimization. The goal is to solve an empirical risk minimization problem over some class of models \( \mathcal{H} \), subject to fairness constraints. For example, we might ask to find the best performing model \( h \in \mathcal{H} \) that equalizes false positive rates, false negative rates, overall error rates, or positive classification rates across some collection of groups \( \mathcal{G} [7, 13] \). For each of these notions of fairness, there is a continuum of relaxations to consider: rather than asking that (e.g.) false positive rates be exactly equalized across groups, we could ask that they differ by no more than 5%, or 10%, or 15%, etc. Because these relaxations trade off with model accuracy (tracing out Pareto frontiers), it is common to explore the range of tradeoffs for a family of fairness constraints\(^1\) (see e.g. [1, 18]).

Each of these are distinct problems that seemingly require training distinct models from scratch based on training data. Moreover, each of these problems can be computationally expensive to solve: for example, the approach of [1] requires solving roughly \( \log |\mathcal{G}| \epsilon^2 \) empirical risk minimization problems over \( \mathcal{H} \) to produce an \( \epsilon \)-approximately optimal solution to any one of them, and the computation of one solution is not used to reduce the cost of the others. The goal of our work is to understand when we can precompute a single regression model \( \hat{f} \) which is sufficient to find optimal solutions to all of the fair machine learning problems described above, each as only a computationally easy post-processing of \( \hat{f} \).

1.1 Our Results in Context
The idea of post-processing a trained model \( \hat{f} \) in order to satisfy fairness constraints is not new. For example, [13] propose a simple post-processing of a regression function \( \hat{f} \) to derive a classifier subject to false positive or negative rate constraints, and a number of more recent works have refined this approach (see our discussion in the related work). However, the conditions under which such post-processing approaches work are still not yet fully understood. The original work of [13] handles the case in which the groups \( \mathcal{G} \) are disjoint, by finding a different thresholding of \( \hat{f} \) for each group \( g \in \mathcal{G} \)—but this approach does not scale well to intersecting groups, since it would naively require tuning a different threshold to each of the possibly \( 2^k \) intersections of \( k \) underlying groups. [13] and [4] show that this post-processing yields the Bayes Optimal fair classifier if \( \hat{f} \) is the true conditional label distribution—a very strong assumption. In this work, we show how to efficiently post-process a regression function to obtain a variety of “fair” classifiers, even when the groups \( \mathcal{G} \) in question intersect, and give accuracy guarantees under substantially weaker assumptions on \( \hat{f} \) than that it correctly encodes the true conditional label distribution.

Post Processing for Intersecting Groups. Suppose we have \( k = |\mathcal{G}| \) groups that are intersecting (e.g. divisions of a population by race, gender, income, nationality, etc.) A naive reduction to the post-processing approach of [13] would consider all \( 2^k \) (now disjoint) intersections of groups, and find a separate thresholding of \( \hat{f}(x) \) for each one. We show that even when groups intersect, for a variety of fairness constraints, the optimal post-processing \( \hat{h} \) remains a

\(^1\)Hereafter, we refer to such constraints as “fairness constraints”, and models which satisfy the constraint or constraints of interest as “fair”; we will specify precisely which set of constraints we handle in Section 2. Assume fairness constraints are defined with respect to a collection of groups \( \mathcal{G} \).
thresholding that depends on only $k$ parameters $\lambda_g$, one for each group $g$. The value at which to threshold $\hat{f}(x)$ now depends only on these $k$ parameters and the subset of groups that $x$ is contained in. We give a simple, efficient algorithm to compute these optimal post-processings. The algorithm is efficient in the worst case — i.e. it does not have to call any heuristic “learning oracle” as direct learning approaches do [1, 18], and requires access only to a modest amount of unlabeled data from the underlying distribution.

Accuracy Guarantees from Multicalibration. As in [13] when given the Bayes optimal regression function $\hat{f}$ (i.e. $\hat{f}(x)$ is the expected value of $y$ given $x$), our post-processing $\tilde{h}$ yields the Bayes optimal classifier. However, one generally cannot hope to learn the Bayes optimal regression function $\hat{f}$ given a polynomial amount of data and computation. Fortunately, we show one can inexpen-
sively compute the most accurate fair classifier in a class $H$ from a much weaker regression function, namely, from a model $\tilde{f}$ which is multicalibrated with respect to a class of models $H$, a class of groups $G$, and a simple class of functions derived from $H$ and $G$. Learning such a multicalibrated predictor with respect to these classes can be done with polynomial sample complexity in an oracle-efficient manner whenever $H$ and $G$ have polynomial VC dimension — and so both the sample and computational complexity of computing $\hat{f}$ are comparable to what would be required to directly solve a single instance of a fairness constrained optimization problem over $H$.

Taken together, our results suggest that even when a downstream task requires a fairness notion which approximately equalizes statistical loss across groups, this is not necessarily what should be trained. Aiming instead for group-wise fidelity in the form of multi-
calibration provides the flexibility to deploy an optimal downstream model subject to a variety of fairness constraints without destroy-
ing information that would be needed to later relax or tighten those constraints, to remove them or to add more, or to change their type.

1.2 Additional Related Work

There are a number of other papers that study the problem of converting a regression (or “score”) function into a classification rule in the context of fair machine learning. For example, [21] shows that post-processing a learned binary classification model to satisfy fairness constraints can be substantially suboptimal even when the hypothesis class under consideration contains the Bayes optimal predictor, which motivates a focus on post-processing regression functions instead. [22] study the structure of the Bayes optimal fair classifier for several notions of fairness when groups are intersecting, under a continuity assumption on the underlying distribution; they do not consider utility guarantees for post-processing a regression function that does not completely represent the under-
lying probability distribution. [20] and [2] give post-processing algorithms that transforms a score function into a classification function that optimizes different measures of accuracy subject to a variety of fairness constraints using a similar primal/dual perspective that we use in this paper. But these papers do not address the two main questions we raise in our work: intersecting groups, and efficiently learnable conditions on the score function that lead to utility guarantees (they assume that in the limit the true conditional label distribution is learnable and given as input to their algorithm).

In proving our accuracy bounds, we draw on a recent line of work on multicalibration [8, 12, 14, 16, 19]. In particular, [11] showed that regression functions that are multicalibrated with respect to a class of models $H$ are omnipredictors with respect to $H$, which means that they can be post-processed to perform as well as the best model in $H$ with respect to any convex loss function satisfying mild technical conditions. The results in our paper can be viewed as being a constrained optimization parallel to [11], which studies unconstrained optimization.

Several other papers also use multicalibration of intermediate statistical products to argue for the utility of downstream models. [23] consider the problem of calibrating a model to the utility function of a downstream utility maximizing decision maker to preserve the usefulness of the model for the decision-maker. [5] show that a proxy model for a protected attribute can be useful in enforcing fairness constraints on a downstream model when the real protected attribute is not available if the proxy is appropriately multicalibrated. [10] study how refining a regression function affects the fairness and accuracy of downstream models derived from it; they propose in their discussion that multicalibration might provide a means to provide guarantees for overlapping populations; our work can be seen as carrying out this proposal.

[15] independently study a problem similar to ours. Our two papers derive a closely related but incomparable set of results. [15] tackles a more general problem, and studies a richer set of objective functions and constraints (whereas we restrict attention to the classification error objective and fairness motivated constraints). In contrast, in our paper, we are able to take advantage of the additional structure of our problem to derive improved bounds. In particular, we can handle intersecting groups (with running time and sample complexity depending polynomially on the number of groups), whereas [15] requires taking all of the exponentially many group intersections to recover disjoint groups—which leads to an exponential (in the number of groups) loss in the running time and sample complexity. Similarly, they require more precise multicalibration as more groups are added, whereas we derive results from a multicalibrated predictor with parameter that is independent of the number of groups.

1.3 Limitations

This work (and the literature to which it contributes) explores algo-
rithmic approaches that reduce complex and ambiguous social ideas of fairness to mathematical formalisms (such as equality of false positive rates between coarse-grained groups of individuals). Our work can be applied only when evaluating the membership of an individual in a group is well-defined, when consideration of group membership is legal\footnote{Note that in some contexts such as consumer lending in the United States, direct consideration of membership in protected groups such as race is illegal. However, demographic information can be used when designing and auditing a decision-making process, so long as those characteristics are not part of the real-time lending decisions.}, and when the training data is repre-
sentative of the underlying population. There will be contexts in which these assumptions are either false, overly simplistic, or bypass larger questions. As an example, an application might be fair in its performance but fundamentally unethical in the first place, or groups may be systematically underrepresented in datasets. In the latter case, the guarantees of our work cannot be interpreted as
guarantees relative to the optimal predictor for the true distribution over groups.

It is worth noting that while the assumption that we can define group membership of individuals simplifies the complexities of personal identity, this work does improve on the existing literature on post-processing approaches to fairness in that it allows for non-disjoint, or intersectional, group membership. In general, this work (and all work in algorithmic fairness) should not be assumed to "solve" fairness. Instead it should be taken as a tool in a larger system to evaluate and remediate issues of fairness and ethics in machine learning.

2 PRELIMINARIES

We study regression and binary classification problems. Let $X$ be an arbitrary feature space and $\mathcal{Y} = \{0, 1\}$ be a binary label space. A classification problem is defined by an underlying data distribution $D \in \mathcal{D}(X \times \mathcal{Y})$. In general we will not have direct access to the data distribution, but rather only samples drawn i.i.d. from $D$. We let $D$ denote a dataset of size $n$, drawn i.i.d. from $D: D \sim D^n$.

We will study both regression functions $f : X \to \mathbb{R}$ and classification functions (classifiers) $h : X \to \{0, 1\}$. In general we will use $f$ and variants ($f^*, \hat{f}$, etc.) when speaking of regression functions and $h$ and variants ($h^*, \hat{h}$, etc.) when speaking of classification functions. Our interest will be in regression functions used to estimate conditional label expectations in binary prediction problems, and so the natural range of our regression functions will be (discrete subsets of) $[0, 1]$. When discussing classification error, we will use $\ell$ to denote the 0-1 loss function.

**Definition 1 (Bayes Optimal Regression Function).** We let $f^*$ denote the Bayes optimal regression function $f^* = \arg\min_f \mathbb{E}_{(x, y) \sim D}(f(x) - y)^2$ which takes value:

$$f^*(x) = \mathbb{E}_{(x', y') \sim D}[y'|x' = x]$$

**Remark 1.** $f^*$ encodes the true conditional label expectations. We use property of Bayes optimality going forward. We do not use that $f^*$ also minimizes squared error.

Let $D_\mathcal{X}$ denote the marginal distribution on features induced by projecting $D$ onto $\mathcal{X}$. Note that we can equivalently sample a pair $(x, y) \sim D$ by first sampling $x \sim D_\mathcal{X}$ and then sampling $y = 1$ with probability $f^*(x)$ and $y = 0$ otherwise.

Given a classifier $h : X \to \mathcal{Y}$, and a data distribution $D$, we can refer to various notions of error. We will be interested in both overall error and on subsets of the data that we call groups (which we might think of as demographic groups when the data represents people). We will represent groups by group indicator functions:

**Definition 2.** Let $\mathcal{G}$ denote a collection of groups, each represented by a group indicator function $g : X \to \{0, 1\}$. If $g(x) = 1$ we call $x$ a member of group $g$. Let $\ell$ denote the group containing all elements $(\ell(x) = 1 \text{ for all } x)$. We will always assume that $\ell \in \mathcal{G}$.

We allow $\mathcal{G}$ to contain arbitrarily intersecting groups. We now use this notation to denote the error rates and false positive rates a classifier has over these groups.

**Definition 3.** The error of a classifier $h : X \to \mathcal{Y}$ on a group $g$ as measured over distribution $D$ is:

$$err(h, g, D) = \mathbb{E}_{(x, y) \sim D}[\ell(h(x), y) | g(x) = 1]$$

The false positive rate (FPR) of a classifier $h : X \to \mathcal{Y}$ on a group $g$ is:

$$\rho(h, g, D) = \mathbb{P}_{(x, y) \sim D}[h(x) = 1 | y = 0, g(x) = 1]$$

When $h$ is a randomized classifier, the probabilities are computed over the randomness of $h$ as well. For convenience, we write $err(h) = err(h, l, D)$, $\rho_g(h) = \rho(h, g, D)$, and $\rho(h) = \rho(h, l, D)$.

**Definition 4.** We say that classifier $h : X \to \mathcal{Y}$ satisfies $\gamma$-False Positive Fairness with respect to $D$ and $\mathcal{G}$ if for all $g \in \mathcal{G}$,

$$w_g \left| \rho_g(h) - \rho(h) \right| \leq \gamma.$$  

where $w_g = \mathbb{P}_{(x, y) \sim D}[y = 0, g(x) = 1]$.

**Remark 2.** In the above definition, we include a multiplicative factor that provides slack in the fairness guarantee for groups with small weight over the distribution. This approximation parameter is necessary for learning from a finite sample, as statistical estimation over small groups is inherently more difficult. An equivalent alternative would be to remove the $w_g$ term in our constraints and provide guarantees only for groups for whom $w_g$ is sufficiently large.

**Remark 3.** We will find it convenient to work with an equivalent formulation of error and false positive rates which do not explicitly condition on $g(x) = 1$, but instead multiply by $g(x)$:

$$err(h, g) = \mathbb{E}_{(x, y) \sim D}[\ell(h(x), y) | g(x) = 1]$$

$$\rho_g(h) = \mathbb{P}_{(x, y) \sim D}[h(x) = 1 | y = 0, g(x) = 1]$$

$$\rho(h) = \mathbb{P}_{(x, y) \sim D}[h(x) = 1 | y = 0]$$

For the sake of brevity, in the main body of this paper we prove all results in the context of $\gamma$-False Positive Fairness. We discuss the modifications necessary to extend the results to other fairness notions in Appendix A.

We will study how to derive classifiers with optimal error properties, subject to fairness-motivated constraints on group-wise error rates, from regression functions satisfying multicalibration constraints [14]. Informally, if $\hat{f}$ is multicalibrated with respect to a class of functions $C$, then $\hat{f}(x)$ takes values equal to $f^*(x)$ in expectation, even conditional on both the value of $\hat{f}(x)$ and on the value of $c(x)$ for each $c \in C$. We use two variants. The first (multicalibration in expectation) was defined and studied in [11]:

**Definition 5 (Multicalibrated in Expectation [11, 14]).** Fix a distribution $D$ and $C$ a collection of functions $c : X \to [0, 1]$. Fix a predictor $\hat{f} : X \to R$ where $R$ is some discrete domain $R \subset [0, 1]$.
We say \( \hat{f} \) is \( \alpha \)-approximately multicalibrated with respect to \( C \) if for every \( c \in C \):
\[
\sum_{x \in R} P[\hat{f}(x) = o] \mathbb{E}[(\hat{f} - f^*)(x) \cdot c(x, o) | \hat{f}(x) = o] \leq \alpha.
\]

We will require this notion of multicalibration with respect to the set of groups \( G \) with which we define our fairness constraints, for the classifiers \( h \in \mathcal{H} \), and for the intersection of these classes \( G \times H = \{g(x) \cdot h(x) | g \in G, h \in \mathcal{H} \} \). We will also need a variant of multicalibration that is tailored to two-argument functions \( c : \mathcal{X} \times \mathcal{R} \rightarrow \{0, 1\} \) in order to argue about the properties of thresholding functions, which take both a value \( x \in \mathcal{X} \) and a threshold in a discrete domain \( R \subseteq \{0, 1\} \), and threshold predictions to \{0, 1\}.

In this definition, when we condition on \( \hat{f}(x) = o \), we also condition on the second argument of \( c \) taking the same value \( o \). We call this joint multicalibration. It is only a modest generalization of multicalibration: we verify in Appendix C that existing algorithms for obtaining multicalibrated predictors easily extend to our definition of joint multicalibration.

**Definition 6 (Joint Multicalibration in Expectation).** We say that a predictor \( f : \mathcal{X} \rightarrow \mathcal{R} \) where \( \mathcal{R} \) is some discrete domain \( R \subseteq \{0, 1\} \) is \( \alpha \)-approximately jointly multicalibrated with respect to a class \( C \) of functions \( c : \mathcal{X} \times \mathcal{R} \rightarrow \{0, 1\} \) if for every \( c \in C \):
\[
\sum_{x \in \mathcal{X}} P[\hat{f}(x) = o] \mathbb{E}[(\hat{f} - f^*)(x) \cdot c(x, o) | \hat{f}(x) = o] \leq \alpha.
\]

### 3 THE STRUCTURE OF AN OPTIMAL POST-PROCESSING

In this section, we consider a fairness-constrained optimization problem of finding a model (or distribution over models) in \( \mathcal{H} \) that minimizes error subject to a constraint on group-wise false positive rates:

\[
\min_{h \in \mathcal{A} \mathcal{H}} \text{err}(h), \quad \text{s.t.} \quad w_g \rho_g(h) - \rho(h) \leq \gamma \quad \text{for each } g \in G,
\]

where \( w_g, \rho_g(h), \) and \( \rho(h) \) are defined as in Definition 3.

We now rewrite this error minimization optimization in a more convenient and more general form below. First, we describe the error of \( h \) with respect to an arbitrary regression function \( f \), which is just how far \( h \) is from matching \( f \)'s conditional label distribution. This error total error can be broken down into events \( h(x) = 1 \), of which an \( 1 - f(x) \) fraction should be 0; and where \( h(x) = 0 \), of which an \( f(x) \) fraction should be 1. We rewrite the constraint in a similar fashion, switching from conditioning on group membership to multiplying by the indicator function as described in Remark 3.

**Definition 7.** Let \( f : \mathcal{X} \rightarrow \mathcal{R} \subseteq \{0, 1\} \) be some regression function and let \( \gamma \in \mathbb{R}_+ \). Define \( \psi(f, \gamma, \mathcal{H}) \) to be the following optimization problem:

\[
\min_{h \in \mathcal{A} \mathcal{H}} \mathbb{P}_{x \sim \mathcal{D}} \left[ (1 - f(x)) \cdot \ell(h(x), 0) \right] + \mathbb{E} \left[ f(x) \cdot \ell(h(x), 1) \right] \\
\text{s.t. for each } g \in G : \quad \mathbb{E}[\{(1 - f(x)) \cdot \ell(h(x), 0) \cdot g(x)\} - \beta_g \mathbb{E}[\{(1 - f(x)) \cdot \ell(h(x), 0)\}] \leq \gamma,
\]

where \( \beta_g = \mathbb{P}[g(x) = 1 | y = 0] \).

For \( f = f^* \), this definition is equivalent to 1:

**Lemma 1.** OptimalFair. Let \( f^* \) be the Bayes optimal regression function over \( D \). Then optimization problem \( \psi(f^*, \gamma, \mathcal{H}) \) is equivalent to the fairness-constrained optimization problem (1).

The proof is in Appendix B. We will be interested in the properties of the optimal solution to \( \psi(f, \gamma, \mathcal{H}) \), which will be elucidated via its Lagrangian. Note that the optimization problem has 2|G| linear inequality constraints. Let \( \lambda = (\lambda^+_g)_{g \in G} \) denote the vector of 2|G| dual variables corresponding to those constraints, and write \( \lambda_g = \lambda^+_g - \lambda^-_g \).

**Definition 8 (Lagrangian).** Given any regression function \( f \), we define a Lagrangian of the optimization problem \( \psi(f, \gamma, \mathcal{H}) \) as \( L_f : \mathcal{H} \times \mathbb{R}^{|G|} \rightarrow \mathbb{R} \):

\[
L_f(h, \lambda) = \mathbb{P}_{x \sim \mathcal{D}} \left[ f(x) \ell(h(x), 1) + (1 - f(x)) \ell(h(x), 0) \right] + \sum_{g \in G} \lambda^+_g (1 - f(x)) \cdot \ell(h(x), 0) g(x) - \beta_g (1 - f(x)) \ell(h(x), 0) - \gamma
\]

\[
+ \sum_{g \in G} \lambda^-_g \beta_g \ell(h(x), 0) (1 - f(x)) - (1 - f(x)) \ell(h(x), 0) g(x) - \gamma \right].
\]

For convenience, given a Bayes optimal regressor \( f^* \), we write \( L^* = L_{f^*} \). Given any regressor \( \hat{f} \), we write \( \hat{L} = L_{\hat{f}} \).

Let \( \mathcal{H}_A = 2^\mathcal{X} \) be the set of all Boolean functions \( h : \mathcal{X} \rightarrow \{0, 1\} \). We will consider solving our optimization problem over \( \mathcal{H}_A \).

### 3.1 Computing the optimally post-processed classifier

To approximate \( h \) given \( f \), we need to compute an approximately optimal solution to the linear program \( \psi(f, \gamma, \mathcal{H}_A) \). We accomplish this by playing a no-regret vs. best response algorithm over the primal and dual variables of the linear program [9]. The dual player is playing gradient descent over the set of dual variables \( \lambda \) and the primal player best responds by updating their current hypothesis.

To implement the gradient step in practice, we need to estimate the losses of \( h \) with respect to \( f \) from a finite sample. We can do this using a sample of unlabelled data of size which scales logarithmically in the number of constraints and linearly in the number of rounds \( T \) of the no-regret dynamics.

The full formulation of the optimization problem as a zero-sum game and the main algorithm, Algorithm 4, is in Appendix B.

We now introduce notation to describe the structure of the functions output by Algorithm 4, which will be useful when we discuss the necessary multicalibration requirements in the following subsection.

**Definition 9 (Set of Thresholding functions \( \mathcal{B}(C) \)).** Let \( x_g \in \{0, 1\}^{|G|} \) denote the group membership indicator vector of some point \( x \). Define the function:

\[
d(x) = \frac{2 \gamma - 1}{1 - \alpha}.
\]
Then, let for any \( \lambda, x, \beta \)
\[
s_k(x, v) := 1\{ (\lambda, x_G - \beta) \geq d(v) \}.
\]
Define \( \mathcal{B}(C) = \{ s_k | \lambda \in \Lambda(C), \beta = \beta_\gamma, \ldots, \beta_\gamma \} \), where \( \Lambda(C) = \{ \lambda \in \mathbb{R}^{2^G} | ||\lambda||_1 \leq C \} \) and \( \beta_\gamma = \mathbb{P}_{(x,y) \sim \mathcal{D}} [g(x) = 1 | y = 0] \), as defined in Definition 7.

Informally, these functions take an example, and map it to a vector of its group membership, indicating whether a \( \lambda \)-weighting of the example’s group membership is larger than some threshold \( d(v) \). In the following section, we use joint multicalibration with respect to such functions in order to relate the estimated error to the approximate LP solution to its true error. These thresholding functions \( \mathcal{B}(C) \) have a natural relationship to the deterministic thresholded models that we compute at each round of algorithm: we show in Appendix B that the solution at each iteration of Algorithm 4 is exactly a function belonging to \( \mathcal{B}(C) \).

**Theorem 1.** algmain Let \( \text{OPT} \) be the objective value of the optimal solution to \( \psi(f, \gamma, \mathcal{H}_A) \). Then, for any \( C \in \mathbb{R} \), after \( T = \frac{1}{4} \cdot C^2 \cdot (C^2 + 4|\mathcal{G}|)^2 \) iterations, Algorithm 4 outputs a randomized hypothesis \( h \) with the following properties:
- the error of the output satisfies \( \text{err}(h) \leq \text{OPT} + \frac{2}{T} \).
- the constraint violation of the output satisfies \( w_g |\rho_g(h) - \rho(h)\| \leq 0 \).
- the output \( h \) is the uniform mixture over \( T \) constituent models, each of which are threshold functions belonging to \( \mathcal{B}(C) \), to give a final solution which approximately satisfies the desired constraints.

The full proof of Theorem 6 is in Appendix B.

**Remark 4.** Although we use the standard techniques of Freund and Schapire [9] to solve the LP formulation of the problem, we provide a full description of the techniques and our application of them in Appendix B. We do so to emphasize that this choice is crucial to our solution. The chosen approach allows us to fully specify a post-processing function by deterministically breaking ties in each individual round of the zero-sum game dynamics and then uniformly randomizing over these iterates, each of which are threshold functions belonging to \( \mathcal{B}(C) \), to give a final solution which approximately satisfies the desired constraints.

### 3.2 From a Multicalibrated Regression Function \( \hat{f} \)

Thus far, we have considered the optimization problem \( \psi(f, \gamma, \mathcal{H}_A) \) in the abstract, have characterized its optimal solution \( \hat{h} \), and have given a simple algorithm to find \( \hat{h} \), an approximately optimal solution. When \( f = f^* \), \( h = h^* \) is the Bayes optimal fair classifier, and \( \hat{h} \) is approximately Bayes optimal. But in practice, we will not have access to \( f^* \), but will instead only have some surrogate function, which we will call \( \hat{f}(x) \). We will argue that if \( \hat{f} \) is appropriately **multicalibrated**, then it is good enough for our purposes. We will compare the approximate solution \( \hat{h} \) produced by Algorithm 4 to the optimization problem \( \psi(f, \gamma, \mathcal{H}_A) \) which has corresponding Lagrangian \( \hat{L}(h, \lambda) \), as defined in Definition 8 to the optimal solution \( (h^*, \lambda^*) \) to the optimization problem \( \psi(f^*, \gamma, \mathcal{H}) \) for some constrained class \( \mathcal{H} \), and show conditions under which they are close.

In order to proceed, we first need to determine what our surrogate function ought to be multicalibrated with respect to. In addition to being \( \alpha \)-approximately multicalibrated in expectation with respect to \( \mathcal{G} \) and \( \mathcal{H} \), we will require that \( \hat{f} \) be \( \alpha \)-approximately multicalibrated with respect to \( \mathcal{G} \times \mathcal{H} = \{ g(x) \cdot h(x) | g \in \mathcal{G}, h \in \mathcal{H} \} \). Furthermore, we will need to require that \( \hat{f} \) be \( \alpha \)-approximately jointly multicalibrated in expectation with respect to the set of functions \( \mathcal{B}(C) \times \mathcal{G} \).

**Remark 5.** When the groups of interest are disjoint, joint multicalibration with respect to the class \( \mathcal{B}(C) \) is implied by multicalibration with respect to \( \mathcal{G} \). But when groups can intersect, this is not an implication, and satisfying joint multicalibration with respect to \( \mathcal{B}(C) \) adds new constraints on \( \hat{f} \).

With these preliminaries behind us, we can now state our main theorem, which shows that for any class of models \( \mathcal{H} \) and class of groups \( \mathcal{G} \), given an approximately multicalibrated \( \hat{f} \) (with multicalibration requirements depending on \( \mathcal{H}, \mathcal{G}, \) and \( \mathcal{B}(C) \)), the model \( \hat{h} \) output by Algorithm 4 achieves an error rate and fairness guarantees comparable to the optimal solution to \( \psi(f^*, \gamma, \mathcal{H}) \).

**Theorem 2.** finalerror Set \( C = \sqrt{1/a} \). Let \( \hat{f} \) be \( \alpha \)-approximately multicalibrated in expectation with respect to \( \mathcal{G} \), and \( \mathcal{G} \times \mathcal{H} \) and \( \alpha \)-approximately jointly multicalibrated in expectation with respect to \( \mathcal{G} \times \mathcal{B}(C) \). Let \( \hat{h} \) be the result of running Algorithm 4 with input \( \hat{f} \) and \( C \). Then, \( \text{err}(\hat{h}) \leq \text{err}(h^*) + a(5 + 2\sqrt{1/a}) + 2\sqrt{a} \), and for all \( g \in \mathcal{G}, w_g |\rho_g(h) - \rho(h^*)| \leq w_g |\rho_g(h^*) - \rho(h^*)| + 2a. \)

**Proof Sketch.** Generalizing notation from the previous sections, let \( \text{err}(h) = \mathbb{E}_{x \sim \mathcal{D}} \left[ f^*(x)l(h(x), 1) + (1 - f^*(x))l(h(x), 0) \right] \) denote the true error of \( h \) on the distribution (i.e. as measured according to the true conditional label distribution \( f^* \)), and let \( \text{err}(h) = \mathbb{E}_{x \sim \mathcal{D}} \left[ \hat{f}(x)l(h(x), 1) + (1 - \hat{f}(x))l(h(x), 0) \right] \) denote the error of \( h \) as estimated using the surrogate function \( \hat{f} \). At a high level, the proof of Theorem 7 will proceed as follows:

\[
\text{err}(h^*) = L^*(h^*, \lambda^*) \geq L^*(h^*, \hat{\lambda}) \approx L(\hat{h}, \hat{\lambda}) \geq \text{err}(\hat{h}) \approx \text{err}(\hat{h}) \approx \text{err}(h).
\]

Each of these steps takes a lemma (presented in full in the appendix) to justify, but the logic is at a high level as follows: The equalities on Lines 2 and 6 follow from complimentary slackness: at the optimal solution \( (h^*, \lambda^*) \) it must be that for each constraint \( g \) either the constraint is exactly tight so that its ‘violation’ term in the Lagrangian evaluates to 0, or its corresponding dual variable \( \lambda^*_g = 0 \). Thus, all terms in the Lagrangian other than the objective evaluate to 0. The inequality in Line 3 follows from the dual optimality condition that \( \lambda^* \in \arg \max \lambda^*L^*(h^*, \lambda) \) and similarly the inequality in Line 5 follows from the primal optimality condition that \( h \in \arg \min_w \mathbb{E}_{\mathcal{D}} L(h, \hat{\lambda}) \). Line 7 follows from the fact that \( \hat{h} \) is an approximately optimal solution to \( \psi(\hat{f}, \gamma, \mathcal{H}_A) \). Lines 4 and 8
follow from our multicalibration guarantees, the former from multicalibration with respect to groups and our hypothesis class, and the latter from joint multicalibration with respect to the functions $B(C) \times G$.

We provide the proof of Lines 4 and 8 below. The first to demonstrate how we use multicalibration to show closeness of the Lagrangian with respect to the Bayes optimal regressor $f^*$ and the Lagrangian with respect to the multicalibrated regressor $\hat{f}$; the second to demonstrate how we use joint multicalibration to show that the error of the solution output by Algorithm 4 with respect to the multicalibrated regressor $\hat{f}$ is close to its error with respect to the Bayes optimal regressor $f^*$. The remainder of the proof is in Appendix B.

**Lemma 2. lagrclose(Bounding Equation 3 by Equation 4)** Fix any $\lambda$. If $\hat{f}$ is $\alpha$-multicalibrated with respect to $G$, $H$, and $G \times H = \{g(x) \cdot h(x) \mid g \in G, h \in H\}$, then then we have

$$\left| \hat{L}(h^*, \lambda) - L^*(h^*, \lambda) \right| \leq \alpha(3 + 2\|\lambda\|_1).$$

**Proof.** Define $\kappa = 1 + \sum_{g \in G} \lambda_g (g(x) - \beta_g)$ Observe that we can write:

$$\hat{L}(h, \lambda) = L_1(h, \lambda) - \gamma \sum_{g \in G} \lambda^+_g + \lambda^-_g - L_2(h, \lambda),$$

where

$$L_1(h, \lambda) = \mathbb{E}_{x \sim D_x} \left[ \ell(h(x), 0) \cdot \kappa \right],$$

$$L_2(h, \lambda) = \mathbb{E}_{x \sim D_x} \left[ \hat{f}(x)(\ell(h(x), 1) + \ell(h(x), 0) \cdot \kappa) \right].$$

Similarly, we can write:

$$L^*(h, \lambda) = L_1(h, \lambda) - \gamma \sum_{g \in G} \lambda^+_g + \lambda^-_g - L_2^*(h, \lambda),$$

where

$$L^*_2(h, \lambda) = \mathbb{E}_{x \sim D_x} \left[ f^*(x)(\ell(h(x), 0) \cdot \kappa - \ell(h(x), 1)) \right].$$

Observe that the $L_1$ term does not depend on $\hat{f}$ or $f^*$ and so is common between $\hat{L}$ and $L^*$. We can bound $L_2$ as follows:

$$L_2(h^*, \lambda) = \mathbb{E}_{x \sim D_x} \left[ \hat{f}(x) \left( -\ell(h^*(x), 1) + \ell(h^*(x), 0) \cdot \kappa \right) \right]$$

$$= \mathbb{E}_{x \sim D_x} \left[ \hat{f}(x) \left( -1 - h^*(x) + h^*(x) \cdot \kappa \right) \right]$$

$$= \sum_{v \in R} \mathbb{P}[\hat{f}(x) = v] \mathbb{E}_{x \sim D_x} \left[ \hat{f}(x)(-1 - h^*(x)) + h^*(x)(\kappa) \cdot \hat{f}(x) = v \right]$$

$$\leq \sum_{v \in R} \mathbb{P}[\hat{f}(x) = v] \mathbb{E}_{x \sim D_x} \left[ f^*(x)(-1 - h^*(x)) + h^*(x)(\kappa) \cdot \hat{f}(x) = v \right]$$

$$+ \gamma \sum_{g \in G} \lambda_g (1 + \beta_g) \leq L^*_2(h^*, \lambda) + \alpha(3 + 2\|\lambda\|_1),$$
We can compute:

\[
\leq 3\alpha + \sum_{g \in \mathcal{G}} \lambda_g (1 + \beta_g) \alpha \\
\leq 3\alpha + \alpha \sum_{g \in \mathcal{G}} \lambda_g (1 + \max_{g' \in \mathcal{G}} \beta_{g'}) \\
\leq 3\alpha + \alpha \sum_{g \in \mathcal{G}} \lambda_g (1 + 1) \\
\leq 3\alpha + 2\|\lambda\|_1 \alpha.
\]

Similarly, we can show that \(L'(h^*, \lambda) - \tilde{L}(h^*, \lambda) \leq \alpha (3 + 2\|\lambda\|_1)\).

Putting everything together, we get that:

\[
|\tilde{L}(h^*, \lambda) - L'(h^*, \lambda)| \leq \alpha (3 + 2\|\lambda\|_1).
\]

This concludes the proof. \(\square\)

We now provide the proof of Line 8.

**Lemma 3 (Bound of Equation 7 by Equation 8).** Let \(\hat{f}\) be \(\alpha\)-approximately jointly multicalibrated with respect to \(B(\mathcal{C}) \times \mathcal{G}\). Then,

\[
|\bar{err}(\hat{h}) - err(\hat{h})| \leq 2\alpha.
\]

**Proof.** Since \(\hat{h}\) is a randomized model that mixes uniformly over model \(\hat{h}_t\) for \(t \in [T]\), it suffices to show that for every \(t \in [T]\),

\[
|\bar{err}(\hat{h}_t) - err(\hat{h}_t)| \leq 2\alpha.
\]

We can compute:

\[
\bar{err}(\hat{h}_t) = \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \ell(\hat{h}_t(x), 1) + (1 - \hat{f}(x)) \ell(\hat{h}_t(x), 0) \right],
\]

\[
= \sum_{v \in \mathcal{R}} \mathbb{P}[\hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 0] \cdot \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \ell(\hat{h}_t(x), 1) + (1 - \hat{f}(x)) \right],
\]

\[
\ell(\hat{h}_t(x), 0) | \hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 0 \right] = \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \ell(\hat{h}_t(x), 1) + (1 - \hat{f}(x)) \right],
\]

\[
+ \sum_{v \in \mathcal{R}} \mathbb{P}[\hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 1] \cdot \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \ell(\hat{h}_t(x), 1) + (1 - \hat{f}(x)) \right],
\]

\[
\ell(\hat{h}_t(x), 0) | \hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 1 \right] = \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \ell(\hat{h}_t(x), 0) f(x) = v, s_{\lambda_{t-1}}(x, v) = 1 \right].
\]

By Lemma 16, \(\hat{h}_t(x) = s_{\lambda_{t-1}}(x, \hat{f}(x))\), and so in particular conditioning on \(\hat{f}(x) = v\) and \(s_{\lambda_{t-1}}(x, v)\) fixes the value of \(\hat{h}_t(x)\). So, we can rewrite the above as

\[
\bar{err}(\hat{h}_t) = \sum_{v \in \mathcal{R}} \mathbb{P}[\hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 0] \cdot \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) | \hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 0 \right] + \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) | \hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 1 \right] + \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) | \hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 1 \right] + \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) | \hat{f}(x) = v, s_{\lambda_{t-1}}(x, v) = 0 \right] + 2\alpha.
\]

where the inequality comes from our \(\alpha\)-approximate joint multicalibration guarantee. The same argument yields the opposite direction, completing the proof. \(\square\)

**4 CONCLUSION**

We describe a post-processing method that takes as input a regression function and, using a reasonable amount of unlabeled data, outputs an approximately optimal classifier which satisfies a variety of fairness constraints over intersecting demographic groups. The main contribution we make is answering two questions about understanding post-processing methods for fairness constrained optimization: how should we post-process a base regressor to obtain a valuable downstream classifier and for what (weak) conditions of the base regressor (weaker than Bayes optimality, for example) can we give provable guarantees of the post-processing? We show that the algorithmic description of an error-minimizing and fair post-processing is a simple linear threshold function and that beginning with a multicalibrated base regressor results in an approximately optimal and fair classifier.

**REFERENCES**

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*. PMLR, 60–69.

[2] Ibrahim M. Alabdulmohsin and Mario Lucic. 2021. A near-optimal algorithm for debiasing trained machine learning models. *Advances in Neural Information Processing Systems* 34 (2021), 8072–8084.

[3] Rael Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. 2016. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 1046–1059.

[4] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*. 797–806.

[5] Emily Diana, Wesley Gill, Michael Kearns, Krishnamurthi Raghavan, Aaron Roth, and Saeid Sharifi-Malvajerdi. 2021. Multiaccurate proxies for downstream fairness. *arXiv preprint arXiv:2107.04423* (2021).

[6] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. 2015. Preserving statistical validity in adaptive data
analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. 117–126.

[7] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[8] Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. 2021. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 1095–1108.

[9] Yoav Freund and Robert E. Schapire. 1996. Game Theory, on-Line Prediction and Boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory* (Desenzano del Garda, Italy) (COLT ’96). Association for Computing Machinery, New York, NY, USA, 325–332. https://doi.org/10.1145/238061.238163

[10] Sumegha Garg, Michael P Kim, and Omer Reingold. 2019. Tracking and improving information in the service of fairness. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 809–824.

[11] Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. 2022. Omnipredictors. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[12] Varun Gupta, Christopher Jung, Georgy Noarov, Mallesh M Pai, and Aaron Roth. 2022. Online Multivalid Learning: Means, Moments, and Prediction Intervals. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[13] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[14] Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. 2018. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*. PMLR, 1939–1948.

[15] Lunjia Hu, Inbal Livni-Navon, Omer Reingold, and Chutong Yang. 2022. Omnipredictors for Constrained Optimization. In *Manuscript*.

[16] Christopher Jung, Chang-Jua Lee, Mallesh Pai, Aaron Roth, and Balek Vohra. 2021. Moment multicalibration for uncertainty estimation. In *Conference on Learning Theory*. PMLR, 2634–2678.

[17] Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. 2021. A new analysis of differential privacy’s generalization guarantees. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 9–9.

[18] Michael Kearns, Seth Neel, Aaron Roth, and Zhimei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*. PMLR, 2564–2578.

[19] Michael P Kim, Amirat Ghorbani, and James Zou. 2019. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 247–254.

[20] Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flavio P Calmon. 2021. Optimized Score Transformation for Consistent Fair Classification. *J. Mach. Learn. Res.*, 22 (2021), 258–1.

[21] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. 2017. Learning non-discriminatory predictors. In *Conference on Learning Theory*. PMLR, 1920–1953.

[22] Forest Yang, Mouhamedou Cisse, and Sanni Koyejo. 2020. Fairness with overlapping groups; a probabilistic perspective. *Advances in neural information processing systems* 33 (2020), 4067–4078.

[23] Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. 2021. Calibrating Predictions to Decisions: A Novel Approach to Multi-Class Calibration. *Advances in Neural Information Processing Systems* 34 (2021).

[24] Martin Zinkevich. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*. 928–936.
A GENERALIZATION TO OTHER FAIRNESS NOTIONS

We find it convenient to have notation for two quantities which appear repeatedly in the following exposition. Let \( \kappa = 1 + \sum_{g \in G} \lambda_y (g(x) - \beta_g) \) and \( \mu = 1 + \kappa, \) and \( \kappa, \mu \) the same quantities for \( \lambda_{g,t}, \) respectively.

A.1 False Negative (FN) Fairness

**Definition 10.** The false negative rate of a classifier \( h : X \rightarrow Y \) on a group \( g \) is:

\[
\rho_{\text{FN}}(h, g, D) = \mathbb{P}_{(x,y) \sim D} (h(x) \neq y \mid g(x) = 1)
\]

When \( h \) is a randomized classifier, the probabilities are computed over the randomness of \( h \) as well, \( \rho_{\text{FN}}(h, g, D) = \rho(h, g, D), \) and \( \rho_{\text{FN}}(h) \equiv \rho(h, 1, D). \)

**Definition 11.** We say that classifier \( h : X \rightarrow Y \) satisfies \( \gamma \)-False Negative (FN) Fairness with respect to \( D \) and \( G \) if for all \( g \in G, \)

\[
\omega^\text{FN}_g |\rho^\text{FN}_g(h) - \rho_{\text{FN}}(h)| \leq \gamma.
\]

where \( \omega^\text{FN}_g = \mathbb{P}_{(x,y) \sim D} (g(x) = 1, y = 1). \)

We consider the following fairness-constrained optimization problem:

\[
\begin{aligned}
\min_{h \in \mathcal{H}} & \quad \text{err}(h) \\
\text{s.t. for each } & \quad g \in G : \omega^\text{FN}_g |\rho^\text{FN}_g(h) - \rho_{\text{FN}}(h)| \leq \gamma.
\end{aligned}
\]

**Definition 12.** Let \( f : X \rightarrow \mathbb{R} \subseteq [0, 1] \) be some regression function and let \( \gamma \in \mathbb{R}_+. \) Define \( \psi_{\text{FN}}(f, \gamma, \mathcal{H}) \) to be the following optimization problem:

\[
\begin{aligned}
\min_{h \in \mathcal{H}} & \quad \mathbb{E}_{x \sim \mathcal{D}_X} [f(x)\ell(h(x),1) + (1 - f(x))\ell(h(x),0)] \\
\text{s.t. for each } & \quad g \in G : \mathbb{E} [\ell(h(x),1)g(x)f(x)] - \beta^\text{FN}_g \mathbb{E} [\ell(h(x),1)f(x)] \leq \gamma,
\end{aligned}
\]

where \( \beta^\text{FN}_g = \mathbb{P}[g(x) = 1|y = 1]. \)

**Lemma 4.** Let \( f^* \) be the Bayes optimal regression function over \( \mathcal{D}. \) Then optimization problem \( \psi_{\text{FN}}(f^*, \gamma, \mathcal{H}) \) is equivalent to the fairness-constrained optimization problem \( 9. \)

**Proof.** Note that the objective function is equivalent to that of Equation 1, and hence proof of the objectives being equivalent is identical to that of Lemma 12. For the constraints, note that

\[
\omega^\text{FN}_g |\rho^\text{FN}_g(h) - \rho_{\text{FN}}(h)| = \mathbb{P}[g(x) = 1, y = 1] \mathbb{P}[h(x) = 0|g(x) = 1, y = 1] - \mathbb{P}[h(x) = 0|g(x) = 1, y = 1] = \mathbb{P}[g(x) = 1, y = 1] \mathbb{P}[h(x) = 0|g(x) = 1, y = 1] - \mathbb{P}[h(x) = 0|g(x) = 1, y = 1] \mathbb{P}[g(x) = 1, y = 1] = \mathbb{E} [\ell(h(x),1)g(x)f^*(x)] - \mathbb{P}[g(x) = 1, y = 1] \mathbb{E} [\ell(h(x),1)f^*(x)] = \mathbb{E} [\ell(h(x),1)g(x)f^*(x)] - \mathbb{E} [\ell(h(x),1)f^*(x)] - \beta^\text{FN}_g \mathbb{E} [\ell(h(x),1)f^*(x)].
\]

The result follows.

**Definition 13 (Lagrangian).** Given any regression function \( f, \) we define a Lagrangian of the optimization problem \( \psi_{\text{FN}}(f, \gamma, \mathcal{H}) \) as

\[
L^\text{FN}_f(h, \lambda) = \mathbb{E}_{x \sim \mathcal{D}_X} [f(x)\ell(h(x),1) + (1 - f(x))\ell(h(x),0)] + \sum_{g \in G} \lambda^+_g [\ell(h(x),1)g(x)f(x) - \beta_g \ell(h(x),1)f(x)] + \sum_{g \in G} \lambda^-_g [\beta_g \ell(h(x),1)f(x) - \ell(h(x),1)g(x)f(x) - \gamma].
\]
Lemma 5.

\[ L^\text{FN}(h, \lambda) = \mathbb{E}_{x \sim D_x} \left[ \ell(h(x), 0) - \gamma \sum_{g \in G} (\lambda^+_g + \lambda^-_g) 
+ f(x) \left( -\ell(h(x), 0) + \ell(h(x), 1) \left( 1 + \sum_{g \in G} \lambda_g (g(x) - \beta^\text{FN}_g) \right) \right) \right] \]

Proof. Distributing out like terms in the expression for the Lagrangian in Definition 13 gives us

\[ L_f(h, \lambda) = \mathbb{E}_{x \sim D_x} \left[ \ell(h(x), 0) - \gamma \sum_{g \in G} (\lambda^+_g + \lambda^-_g) 
+ f(x) \left( \ell(h(x), 1) - \ell(h(x), 0) + \ell(h(x), 1) \sum_{g \in G} (\lambda^+_g (g(x) - \beta_g) + \lambda^-_g (\beta_g - g(x))) \right) \right] 
= \mathbb{E}_{x \sim D_x} \left[ \ell(h(x), 0) - \gamma \sum_{g \in G} (\lambda^+_g + \lambda^-_g) 
+ f(x) \left( -\ell(h(x), 0) + \ell(h(x), 1) \left( 1 + \sum_{g \in G} (\lambda^+_g - \lambda^-_g) (g(x) - \beta_g) \right) \right) \right]. \]

Recall that \( \lambda_g = \lambda^+_g - \lambda^-_g \), so we are done. \( \square \)

Lemma 6. The optimal post-processed classifier \( h \) of \( \psi(f, \gamma, \mathcal{H}_A) \) for some regressor \( f \) takes the following form:

\[ h(x) = \begin{cases} 
1, & \text{if } f(x) > \frac{1}{p} \text{ and } \mu > 0, \\
0, & \text{if } f(x) < \frac{1}{p} \text{ and } \mu > 0, \\
1, & \text{if } f(x) < \frac{1}{p} \text{ and } \mu < 0, \\
0, & \text{if } f(x) > \frac{1}{p} \text{ and } \mu < 0.
\end{cases} \]

In the edge case in which \( f(x) = \frac{1}{p} \), \( h(x) \) could take either value and might be randomized.

Proof. Note that since we are optimizing over the set of all binary classifiers, \( h \) optimizes the Lagrangian objective pointwise for every \( x \). In particular, we have from Lemma 5 that:

\[ h(x) = \arg \min_p \left[ \ell(p, 0) + f(x) \left( -\ell(p, 0) + \ell(p, 1) \left( 1 + \sum_{g \in G} \lambda_g (g(x) - \beta_g) \right) \right) \right]. \]

In order to determine the threshold, we need to check when setting \( p = 1 \) leads to a value less than setting \( p = 0 \). In other words, we need to solve for \( f(x) \) when

\[ 1 - f(x) < f(x) \left( 1 + \sum_{g \in G} \lambda_g (g(x) - \beta_g) \right) \Rightarrow f(x) > \frac{1}{2 + \sum_{g \in G} \lambda_g (g(x) - \beta_g)}. \]

Thus,

\[ h(x) = \begin{cases} 
1, & \text{if } f(x) > \frac{1}{p} \text{ and } \mu > 0, \\
0, & \text{if } f(x) < \frac{1}{p} \text{ and } \mu > 0, \\
1, & \text{if } f(x) < \frac{1}{p} \text{ and } \mu < 0, \\
0, & \text{if } f(x) > \frac{1}{p} \text{ and } \mu < 0.
\end{cases} \]

\( \square \)
From Lemma 6, we can now define a best-response model and use Algorithm 1 to generate an optimally post-processed model that preserves $\gamma$-False Negative fairness. The algorithm’s error bounds may be derived using symmetric arguments to sections 3.1 and 3.2, where $\tilde{f}$ is required to be $\alpha$-approximately jointly multicalibrated in expectation with respect to $s_j(x, \phi) := 1[\lambda x g - \beta \geq (1 - 2\alpha)/\nu]$ following the same arguments as used in Lemma 16.

**Algorithm 1** Projected Gradient Descent Algorithm for $\gamma$-False Negative Fairness

1. **Input:** $D$: dataset, $f: X \rightarrow [0, 1]$: regression function, $G$: groups, $\gamma$: tolerance on fairness violation, $C$: bound on dual ($\|\lambda\|_1 \leq C$), $\eta$: learning rate
2. **Initialize** dual vector $\lambda^0 = 0$ and set $T = \frac{1}{\eta} \cdot C^2 \cdot (C^2 + 4|G|)^2$.
3. **for** $t = 1, \ldots, T$ **do**
   4. **Primal player updates** $h_t$
      
      $$h_t(x) = \begin{cases} 
      1, & \text{if } f(x) \geq \frac{1}{C} \text{ and } \mu_t > 0, \\
      0, & \text{if } f(x) < \frac{1}{C} \text{ and } \mu_t > 0, \\
      1, & \text{if } f(x) < \frac{1}{C} \text{ and } \mu_t < 0, \\
      0, & \text{if } f(x) \geq \frac{1}{C} \text{ and } \mu_t < 0, \\
      0 & \mu_t = 0 
      \end{cases}$$

5. **Compute**
   $$\hat{\beta}_g = \mathbb{E}_{(x, y) \sim D} [f(h_t(x), 1)g(x) f(x)] \text{ for all } g \in G,$$
   $$\hat{\rho}_t = \mathbb{E}_{(x, y) \sim D} [\beta_g f(h_t(x), 1)f(x)], \text{ where } \beta_g = \mathbb{P}[g(x) = 1|y = 0]$$

6. **Dual player updates**
   $$\lambda^t_{g, +} = \max(0, \lambda^t_{g, +} + \eta \cdot (\hat{\beta}_g - \hat{\rho}_t - \gamma)), \quad \lambda^t_{g, -} = \max(0, \lambda^t_{g, -} + \eta \cdot (\hat{\rho}_t - \hat{\beta}_g - \gamma)).$$

7. **Dual player sets**
   $$\lambda^t = \sum_{g \in G} \lambda^t_{g, +} - \lambda^t_{g, -}$$
8. **if** $\|\lambda^t\|_1 > C$ **then**
9. **set** $\lambda^t = \arg \min_{\lambda \in \mathbb{R}^{|G|}, \|\lambda\|_1 \leq C} \|\lambda^t - \tilde{\lambda}\|_2^2$
10. **end if**
11. **end for**
12. **Output:** $\tilde{h} := \frac{1}{T} \sum_{t=1}^T \hat{h}_t$, a uniformly random classifier over all rounds’ hypotheses.

### A.2 Error Fairness

**Definition 14.** We say that classifier $h: X \rightarrow Y$ satisfies $\gamma$-Error (E) Fairness with respect to $D$ and $G$ if for all $g \in G$,

$$w_g^E[err(h, g, D) - err(h, \bar{D})] \leq \gamma,$$

where $w_g^E = \mathbb{P}_{(x, y) \sim \bar{D}}[g(x) = 1]$. We consider the following fairness-constrained optimization problem:

$$\min_{h \in \mathcal{X}} \mathbb{E}_{(x, y) \sim D \times \mathcal{X}} [f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0)]$$

**Definition 15.** Let $f: X \rightarrow R \subseteq [0, 1]$ be some regression function and let $\gamma \in \mathbb{R}_+$. Define $\psi_{\gamma}(f, \gamma, \mathcal{H})$ to be the following optimization problem:

$$\min_{h \in \mathcal{X}} \mathbb{E}_{(x, y) \sim D \times \mathcal{X}} [f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0)]$$

s.t. for each $g \in G$:

$$\mathbb{E}[(\ell(h(x), 1)g(x)f^+(x) + \ell(h(x), 0)g(x)(1 - f^+(x))) - w_g^E(\ell(h(x), 1)f^+(x) - w_g^E\ell(h(x), 0)(1 - f^+(x)))] \leq \gamma.$$
where \( w_\theta^F = P_{(x,y) \sim D}[g(x) = 1] \) as in the previous definition.

**Lemma 7.** Let \( f^* \) be the Bayes optimal regression function over \( D \). Then optimization problem \( \psi_E(f^*, y, \mathcal{H}) \) is equivalent to the fairness-constrained optimization problem 10.

**Proof.** Note that the objective function is equivalent to that of Equation 1, and hence proof of the objectives being equivalent is identical to that of Lemma 12. For the constraints, note that

\[
\begin{align*}
\text{err}(h, \mathcal{D}) &= \mathbb{P}[g(x) = 1] \mathbb{P}[y = 1 | g(x) = 1] \mathbb{P}[h(x) = 1 | g(x) = 1, y = 1] \\
&\quad + \mathbb{P}[y = 0 | g(x) = 1] \mathbb{P}[h(x) = 1 | g(x) = 1, y = 0] \\
&\quad - (\mathbb{P}[y = 1 | h(x) = 0 | y = 1] + \mathbb{P}[y = 0 | h(x) = 1 | y = 0]) \\
&\quad = \mathbb{P}[g(x) = 1] \mathbb{P}[y = 1 | g(x) = 1] \mathbb{P}[h(x) = 0, g(x) = 1, y = 1] \\
&\quad + \mathbb{P}[y = 1 | g(x) = 1] \mathbb{P}[h(x) = 1, g(x) = 1, y = 0] \\
&\quad - \mathbb{P}[y = 1 | h(x) = 0, y = 1] \mathbb{P}[h(x) = 1, y = 1] \\
&\quad = \mathbb{E}[\ell(h(x), 1)g(x)f^*(x) + \ell(h(x), 0)g(x)(1 - f^*(x)) \\
&\quad - w_\theta^F \ell(h(x), 1)f^*(x) - w_\theta^F \ell(h(x), 0)(1 - f^*(x))]
\end{align*}
\]

**Definition 16 (Lagrangian).** Given any regression function \( f \), we define a Lagrangian of the optimization problem \( \psi_E(f, y, \mathcal{H}) \) as

\[
L_f^E : \mathcal{H} \times \mathbb{R}^{|\mathcal{G}|} \to \mathbb{R}:
\]

\[
L_f^E(h, \lambda) = \mathbb{E}_{x \sim \mathcal{D}_x} \left[ f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0) \\
+ \sum_{g \in \mathcal{G}} \lambda_g^+ \ell(h(x), 1)g(x)f(x) + \ell(h(x), 0)g(x)(1 - f(x)) \\
- w_\theta^F \ell(h(x), 1)f(x) - w_\theta^F \ell(h(x), 0)(1 - f(x)) - \gamma \\
+ \sum_{g \in \mathcal{G}} \lambda_g^- (w_\theta^F \ell(h(x), 1)f(x) + w_\theta^F \ell(h(x), 0)(1 - f(x)) \\
- \ell(h(x), 1)g(x)f(x) - \ell(h(x), 0)g(x)(1 - f(x)) - \gamma) \right].
\]

**Lemma 8.**

\[
L_f^E(h, \lambda) = \mathbb{E}_{x \sim \mathcal{D}_x} \left[ \ell(h(x), 0) \left( 1 + \sum_{g \in \mathcal{G}} \lambda_g (g(x) - w_\theta^F) \right) - \gamma \sum_{g \in \mathcal{G}} (\lambda_g^+ + \lambda_g^-) \\
+ f(x) \left[ - \ell(h(x), 0) \left( 1 + \sum_{g \in \mathcal{G}} \lambda_g (g(x) - w_\theta^F) \right) \\
+ \ell(h(x), 1) \left( 1 + \sum_{g \in \mathcal{G}} \lambda_g (g(x) - w_\theta^F) \right) \right] \right]
\]

**Proof.** Distribute out like terms as shown previously.

**Lemma 9.** The optimal post-processed classifier \( h \) of \( \psi(f, y, \mathcal{H}_\lambda) \) for some regressor \( f \) takes the following form:
Note that since we are optimizing over the set of all binary classifiers, \( h \) optimizes the Lagrangian objective pointwise for every \( x \). In particular, we have from Lemma 8 that:

\[
h(x) = \arg\min_p \left( \ell(p, 0) \left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right) + f(x) \left( -\ell(p, 0) \left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right) + \ell(p, 1) \left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right) \right) \right)
\]

Setting \( p = 0 \) makes the inner portion of the expression evaluate to

\[
f(x) \left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right),
\]

and setting \( p = 1 \) makes the inner portion of the expression evaluate to

\[
\left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right) - f(x) \left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right)
\]

In order to find the optimal \( h \), we want to find the threshold at which setting \( p = 1 \) minimizes the expression, and hence:

\[
\left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right) - f(x) \left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right) < f(x) \left( 1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) \right)
\]

\[
\frac{1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E)}{1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E) + 2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E)} < f(x)
\]

Thus,

\[
h(x) = \begin{cases} 
1, & \text{if } f(x) > \frac{1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E)}{2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E)} \text{ and } 2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E) > 0, \\
0, & \text{if } f(x) < \frac{1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E)}{2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E)} \text{ and } 2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E) > 0, \\
1, & \text{if } f(x) < \frac{1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E)}{2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E)} \text{ and } 2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E) < 0, \\
0, & \text{if } f(x) > \frac{1 + \sum_{g \in G} \lambda_g(g(x) - w_g^E)}{2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E)} \text{ and } 2 + 2 \sum_{g \in G} \lambda_g(g(x) - w_g^E) > 0. 
\end{cases}
\]

From Lemma 9, we can now define a best-response model and use Algorithm 2 to generate an optimally post-processed model that preserves \( \gamma \)-Error fairness. The algorithm’s error bounds may be derived using symmetric arguments to sections 3.1 and 3.2, where \( \hat{f} \) is \( \alpha \)-multicalibrated in expectation with respect to \( G, \mathcal{H}, \) and \( G \times \mathcal{H} \) and is jointly multicalibrated with respect to functions of the form:
Algorithm 2: Projected Gradient Descent Algorithm for $\gamma$-Error Fairness

1: Input: $D$: dataset, $f : X \to [0, 1]$: regression function, $G$: groups, $\gamma$: tolerance on fairness violation, $C$: bound on dual ($\|\lambda\|_1 \leq C$), $\eta$: learning rate
2: Initialize dual vector $\lambda^0 = 0$ and set $T = \frac{1}{\eta} \cdot C^2 \cdot (C^2 + 4|G|)^2$.
3: for $t = 1, \ldots, T$ do
4: Primal player updates

$$
\begin{align*}
1_{\lambda^{t-1}, x_G - w^E} \geq \frac{20 - 1}{1 - 2\alpha}
\end{align*}
$$

the proofs from section 3.2 may be modified to get its desired error bounds.

Theorem 9. The error bounds of Algorithm 2 are

$\Delta(t) \leq C_0 + C_1 \cdot \frac{1}{\lambda_t}$.  

The proof follows from a similar argument as in Theorem 8. 

A.3 Statistical Parity Fairness

**Definition 17.** We say that classifier $h : X \to Y$ satisfies $\gamma$-Statistical Parity (SP) Fairness with respect to $D$ and $G$ if for all $g \in G$,

$$
\mathbb{P}_{(x,y) \sim D}[g(x) = 1] - \mathbb{E}_{(x,y) \sim D}[h(x)|g(x) = 1] \leq \gamma.
$$

We consider the following fairness-constrained optimization problem:

$$
\min_{h \in \mathcal{H}} \text{err}(h) \quad \quad \text{s.t. for each } g \in G : \quad \quad \mathbb{P}[g(x) = 1] - \mathbb{E}_{(x,y) \sim D}[h(x)|g(x) = 1] \leq \gamma.
$$
Definition 18. Let \( f : X \rightarrow R \subseteq [0, 1] \) be some regression function and let \( \gamma \in \mathbb{R}_+ \). Define \( \phi_{SP}(f, \gamma, \mathcal{H}) \) to be the following optimization problem:

\[
\begin{align*}
\min_{h \in \mathcal{H}} & \quad \mathbb{E}_{x \sim \mathcal{D}_X} \left[ f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0) \right] \\
\text{s.t. for each } g & \in \mathcal{G} : \quad \mathbb{E}_{x \sim \mathcal{D}_X} \left[ h(x)g(x) - w_{\gamma}^{SP} \mathbb{E}_{x \sim \mathcal{D}_X} [h(x)] \right] \leq \gamma
\end{align*}
\]

where \( w_{\gamma}^{SP} = \mathbb{P}[g(x) = 1] \).

Lemma 10. Let \( f^\ast \) be the Bayes optimal regression function over \( \mathcal{D} \). Then optimization problem \( \psi_{SP}(f^\ast, \gamma, \mathcal{H}) \) is equivalent to the fairness-constrained optimization problem 11.

Definition 19 (Lagrangian). Given any regression function \( f \), we define a Lagrangian of the optimization problem \( \psi_{SP}(f, \gamma, \mathcal{H}) \) as \( L_f^{SP} : \mathcal{H} \times \mathbb{R}_{2^{|\mathcal{G}|}} \rightarrow \mathbb{R} \):

\[
L_f^{SP}(h, \lambda) = \mathbb{E}_{x \sim \mathcal{D}_X} \left[ f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0) \right] + \sum_{g \in \mathcal{G}} \lambda^+ g(h(x)(g(x) - 1) - \gamma) + \sum_{g \in \mathcal{G}} \lambda^- g(h(x)(1 - g(x)) - \gamma) \right]
\]

Lemma 11. The optimal post-processed classifier \( h \) of \( \psi_{SP}(f, \gamma, \mathcal{H}) \) for some regressor \( f \) takes the following form:

\[
h(x) = \begin{cases} 
1, & \text{if } f(x) > 1/2 + (1/2) \sum_{g \in \mathcal{G}} \lambda^+_g (g(x) - 1), \\
0, & \text{if } f(x) < 1/2 + (1/2) \sum_{g \in \mathcal{G}} \lambda^-_g (g(x) - 1).
\end{cases}
\]

In the edge case in which \( f(x) = 1/2 + (1/2) \sum_{g \in \mathcal{G}} \lambda^+_g (g(x) - 1), h(x) \) could take either value and might be randomized.

Proof. Note that we can rewrite our Lagrangian from Definition 19 as

\[
L_f^{SP}(h, \lambda) = \mathbb{E}_{x \sim \mathcal{D}_X} \left[ f(x)(\ell(h(x), 1) - \ell(h(x), 0)) + \ell(h(x), 0) + h(x) \sum_{g \in \mathcal{G}} \lambda^+_g (g(x) - 1) + \gamma \sum_{g \in \mathcal{G}} (\lambda^+ + \lambda^-) \right]
\]

and hence our optimal \( h \) will be optimal pointwise, i.e.

\[
h(x) \arg \min_p \left[ f(x)(\ell(p, 1) + \ell(p, 0)) - \ell(p, 0) + p \sum_{g \in \mathcal{G}} \lambda^+_g (g(x) - 1) \right]
\]

We can then find our threshold by comparing this expression when \( p = 0 \) and \( p = 1 \), i.e.

\[
-f(x) + 1 + \sum_{g \in \mathcal{G}} \lambda^+_g (g(x) - 1) < f(x) \\
\frac{1 + \sum_{g \in \mathcal{G}} \lambda^-_g (g(x) - 1)}{2} < f(x).
\]

Hence,

\[
h(x) = \begin{cases} 
1, & \text{if } f(x) > 1/2 + (1/2) \sum_{g \in \mathcal{G}} \lambda^+_g (g(x) - 1), \\
0, & \text{if } f(x) < 1/2 + (1/2) \sum_{g \in \mathcal{G}} \lambda^-_g (g(x) - 1). 
\end{cases}
\]

We can now define a best-response model and use Algorithm 3 to generate an optimally post-processed model that preserves \( \gamma \)-Statistical Parity fairness. Assuming that \( \hat{f} \) is \( \alpha \)-multicalibrated in expectation with respect to \( \mathcal{G}, \mathcal{H} \), and \( \mathcal{G} \times \mathcal{H} \) and is jointly multicalibrated with respect to functions of the form \( 1[\langle \lambda, x_G - \beta \rangle \geq 2v - 1] \), the proofs from section 3.2 may be modified to get its desired error bounds.
Algorithm 3 Projected Gradient Descent Algorithm for \( \gamma \)-Statistical Parity Fairness

Input: \( D \): dataset, \( f : X \rightarrow [0, 1] \): regression function, \( G \): groups, \( \gamma \): tolerance on fairness violation, \( C \): bound on dual (\( \| \lambda \|_1 \leq C \)), \( \eta \): learning rate

Initialize dual vector \( \lambda^0 = 0 \) and set \( T = \frac{1}{4} \cdot C^2 \cdot (C^2 + 4|G|)^2 \).

for \( t = 1, \ldots, T \) do
  Primal player updates \( h_t \)
  
  \[ h_t(x) = \begin{cases} 
  1, & \text{if } f(x) \geq 1/2 + (1/2) \sum_{g \in G} \lambda_{g}^{t-1} (g(x) - 1), \\
  0, & \text{if } f(x) < 1/2 + (1/2) \sum_{g \in G} \lambda_{g}^{t-1} (g(x) - 1). 
  \end{cases} \]

  Compute
  
  \[ \hat{\rho}_g^t = \left| \frac{1}{x \sim D} \sum_{x \sim D \setminus h_t(x)} [h_t(x)g(x)] - w_{SP}^{g} \right| \quad \forall g \in G, \]
  \[ \hat{\rho}^t = \frac{1}{T} \sum_{t=0}^{T-1} [f(x)\ell(h_t(x), 1) + (1 - f(x))\ell(h_t(x), 0)], \]

  where \( w_{SP}^{g} = P[g(x) = 1] \).

  Dual player updates
  
  \[ \lambda_{g}^{t+} = \max(0, \lambda_{g}^{t-} + \eta \cdot (\hat{\rho}_g^t - \hat{\rho}^t - \gamma)), \]
  \[ \lambda_{g}^{t-} = \max(0, \lambda_{g}^{t-} + \eta \cdot (\hat{\rho}_g^t - \hat{\rho}^t - \gamma)). \]

  Dual player sets \( \lambda^t = \sum_{g \in G} \lambda_{g}^{t+} - \lambda_{g}^{t-} \).

  if \( \| \lambda^t \|_1 > C \) then
    set \( \lambda^t = \arg \min_{\| \lambda \|_1 \leq C} \| \lambda^t - \lambda^t \|_2 \).
  end if

end for

Output: \( \hat{h} := \frac{1}{T} \sum_{t=1}^{T} \hat{h}_t \), a uniformly random classifier over all rounds’ hypotheses.

A.4 Achieving All Fairness Notions

Ideally, we would like our function to be multicalibrated so that we can achieve any fairness notion downstream. Putting everything together from the previous sections, we can do so.

Definition 20 (Set of thresholding functions \( \mathcal{B}(G) \)). Let \( x_g \in \{0, 1\}^{|G|} \) denote the group membership indicator vector of some point \( x \), and define the following functions:

\[
\begin{align*}
    d^{FP}(v) & := \frac{2v - 1}{1 - v} \\
    d^{FN}(v) & := \frac{1 - 2v}{v} \\
    d^{E}(v) & := \frac{2v - 1}{1 - 2v} \\
    d^{SP}(v) & := 2v - 1 .
\end{align*}
\]

Then, for any \( \lambda, x, \beta \), let

\[
\begin{align*}
    s_{FP}^{\lambda}(x, v) & := 1[\langle \lambda, x_g \rangle - \beta^{FP} \geq d^{FP}(v)], \\
    s_{FN}^{\lambda}(x, v) & := 1[\langle \lambda, x_g \rangle - \beta^{FN} \geq d^{FN}(v)], \\
    s_{E}^{\lambda}(x, v) & := 1[\langle \lambda, ax_g \rangle - \omega^{E} \geq d^{E}(v)], \\
    s_{SP}^{\lambda}(x, v) & := 1[\langle \lambda, x_g \rangle - 1 \geq d^{SP}(v)],
\end{align*}
\]

where

\[
\begin{align*}
    \beta^{FP} & := \{P((x_g) \rightarrow 0) | g(x) = 1, y = 0 \}_{g \in G}, \\
    \beta^{FN} & := \{P((x_g) \rightarrow 0) | g(x) = 1, y = 1 \}_{g \in G}, \\
    \omega^{E} & := \{P((x_g) \rightarrow 0) | g(x) = 1 \}_{g \in G}.
\end{align*}
\]
Define $\mathcal{B}(C) = \{s^{FP}_{\lambda} | \lambda \in \Lambda(C)\} \cup \{s^{PN}_{\lambda} | \lambda \in \Lambda(C)\} \cup \{s^{SP}_{\lambda} | \lambda \in \Lambda(C)\}$, where $\Lambda(C) = \{\lambda \in \mathbb{R}^2 \parallel \lambda \parallel_1 \leq C\}$, as defined in Equation 12.

Then, if $f$ is multicalibrated with respect to $\mathcal{B}(C)$, any of the projected gradient descent algorithms covered above (Algorithms 4 through 2) may be run to achieve the desired fairness notion.

**B EXPANDED PROOFS AND SECTION 3 DISCUSSION**

**Lemma 12.** Let $f^*$ be the Bayes optimal regression function over $\mathcal{D}$. Then optimization problem $\psi(f^*, y, \mathcal{H})$ is equivalent to the fairness-constrained optimization problem (1).

**Proof.** We confirm that the objective and constraints are both equivalent. First the objective:

$$
\text{err}(h) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(h(x), y)]
= \sum_{x,y} \mathbb{P}(X = x, Y = y) \ell(h(x), y)
= \sum_{x \in \mathcal{X}} \mathbb{P}(X = x, Y = 0) \ell(h(x), 0) + \mathbb{P}(X = x, Y = 1) \ell(h(x), 1)
= \mathbb{E}_{X \in \mathcal{X}} [(1 - f^*(x))\ell(h(x), 0) + f^*(x)\ell(h(x), 1)]
$$

For the constraints, note that

$$
w_g | p_g(h) - p(h)| = \mathbb{P}[g(x) = 1, y = 0] \mathbb{P}[h(x) = 1 | g(x) = 1, y = 0] - \mathbb{P}[h(x) = 1 | y = 0]
= \mathbb{P}[g(x) = 1, y = 0] \frac{\mathbb{P}[h(x) = 1, g(x) = 1, y = 0]}{\mathbb{P}[g(x) = 1, y = 0]} - \frac{\mathbb{P}[h(x) = 1, y = 0]}{\mathbb{P}[Y = 0]} - \frac{\mathbb{P}[h(x) = 1, g(x) = 1, y = 0]}{\mathbb{P}[Y = 0]} - \frac{\mathbb{P}[h(x) = 1, y = 0]}{\mathbb{P}[Y = 0]}
- \mathbb{E}[\ell(h(x), 0)g(x)(1 - f^*(x)) - \mathbb{P}[g(x) = 1, y = 0] \mathbb{E}[\ell(h(x), 0)(1 - f^*(x))] - \mathbb{P}[g(x) = 1 | Y = 0] \mathbb{E}[\ell(h(x), 0)(1 - f^*(x))]]
- \mathbb{E}[\ell(h(x), 0)g(x)(1 - f^*(x))] - \beta_g \mathbb{E}[\ell(h(x), 0)(1 - f^*(x))].
$$

The result follows. \(\square\)

**Lemma 13.**

$$
L_f(h, \lambda) = \mathbb{E}_{x \sim \mathcal{D}} \left[ \ell(h(x), 0) \left( x \right) - y \sum_{g \in \mathcal{G}} (\lambda_+^g + \lambda_-^g)
- f(x) \left( - \ell(h(x), 1) + \ell(h(x), 0) \right) \right].
$$
Proof. Distributing out like terms in the expression for the Lagrangian in Definition 8 gives us
\[
L_f(h, \lambda) = \mathbb{E}_{x \sim D_x} \left[ f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0) \right. \\
+ \sum_{g \in G} \lambda_g^+ \left( f(h(x), 0)g(x)(1 - f(x)) - \beta_g\ell(h(x), 0)(1 - f(x)) - \gamma \right) \\
+ \lambda_g^- \left( \beta_g\ell(h(x), 0)(1 - f(x)) - \ell(h(x), 0)g(x)(1 - f(x)) - \gamma \right) \\
= \mathbb{E}_{x \sim D_x} \left[ \ell(h(x), 0) \left( 1 + \sum_{g \in G} \lambda_g^+(g(x) - \beta_g) + \lambda_g^-(\beta_g - g(x)) \right) - \gamma \sum_{g \in G} (\lambda_g^+ + \lambda_g^-) \\
- f(x) \left( - \ell(h(x), 1) + \ell(h(x), 0)(1 + \sum_{g \in G} \lambda_g^+(g(x) - \beta_g) + \sum_{g \in G} \lambda_g^-(\beta_g - g(x))) \right) \\
= \mathbb{E}_{x \sim D_x} \left[ \ell(h(x), 0) \left( 1 + \sum_{g \in G} (\lambda_g^+ - \lambda_g^-)(g(x) - \beta_g) \right) - \gamma \sum_{g \in G} (\lambda_g^+ + \lambda_g^-) \\
- f(x) \left( - \ell(h(x), 1) + \ell(h(x), 0)(1 + \sum_{g \in G} (\lambda_g^+ - \lambda_g^-)(g(x) - \beta_g)) \right) \right].
\]

Recall that \( \lambda_g = \lambda_g^+ - \lambda_g^- \), so we are done. \( \square \)

Definition 21 (Optimal post-processed classifier). We say that a classifier \( h_f \) is an optimal post-processing of \( f \) if there exists a vector \( \lambda^f \) such that the following primal/dual optimality conditions are simultaneously met:

\[
h_f(x) = \arg\min_{h \in \mathcal{H}_A} L_f(h, \lambda^f) \quad \lambda^f = \arg\max_{\lambda \in \mathbb{R}^{|G|}} L_f(h_f, \lambda).
\]

For convenience, we write
\[
h^*(x) = h_f(x) \quad \text{and} \quad \lambda^* = \lambda^f
\]

where \( f^* \) is the Bayes optimal regressor and \( f \) is any other regressor. We will write \( \lambda^*_g \) and \( \hat{\lambda}_g \) to refer to the dual variable in \( \lambda^* \) and \( \lambda^f \) for group \( g \), respectively. We observe that as the optimal solution to the Lagrangian minimax optimization problem, \( h^*(x) \) is the Bayes optimal classifier subject to the fairness constraints in 1.

Lemma 14. The optimal post-processed classifier \( h \) of \( \psi(f, y; \mathcal{H}_A) \) for some regressor \( f \) takes the following form:

\[
h(x) = \begin{cases} 
1, & \text{if } f(x) > \frac{\kappa}{\mu} \text{ and } \mu > 0, \\
0, & \text{if } f(x) < \frac{\kappa}{\mu} \text{ and } \mu > 0, \\
1, & \text{if } f(x) < \frac{\kappa}{\mu} \text{ and } \mu < 0, \\
0, & \text{if } f(x) > \frac{\kappa}{\mu} \text{ and } \mu < 0.
\end{cases}
\]

where \( \kappa = 1 + \sum_{g \in G} \lambda_g(g(x) - \beta_g) \) and \( \mu = 1 + \kappa \).

In the edge case in which \( f(x) = \frac{\kappa}{\mu} \), \( h(x) \) could take either value and might be randomized.

Proof. Note that since we are optimizing over the set of all binary classifiers, \( h \) optimizes the Lagrangian objective pointwise for every \( x \). In particular, we have from Lemma 13 that:

\[
h(x) = \arg\min_p \left[ \ell(p, 0)(x) - f(x)(1 - \ell(p, 1) + \ell(p, 0)(x)) \right].
\]

Determining the optimal threshold is equivalent to determining when the above expression with \( \ell(p, 0) = 1 \) and \( \ell(p, 1) = 0 \) is less than \( f(x) \), i.e.

\[
\kappa - f(x)(x) < f(x) \quad \kappa < f(x) \left( 1 + (x) \right).
\]
Thus,

\[
h(x) = \begin{cases} 
1, & \text{if } f(x) > \frac{\mu}{\rho} \text{ and } \mu > 0, \\
0, & \text{if } f(x) < \frac{\mu}{\rho} \text{ and } \mu > 0 \\
1, & \text{if } f(x) < \frac{\mu}{\rho} \text{ and } \mu < 0, \\
0, & \text{if } f(x) > \frac{\mu}{\rho} \text{ and } \mu < 0
\end{cases}
\]

In Lemma 14, we can only describe the optimal post-processed classifier for cases where either \( f(x) \) is less than or greater than the threshold \( \frac{\mu}{\rho} \), \( h(x) \). In practice, our algorithm will need to update \( h \) at round \( t \) according to the current dual variables \( \lambda \) in a way that is well-defined for all values of \( f(x) \). Hence, we define our best response as follows, where ties between \( f(x) \) and the threshold are broken by rounding to 1.

**Definition 22 (Best Response Model).** Given regressor \( f \) and dual variables \( \lambda \), let the best response \( h \) be defined as

\[
h(x) = \begin{cases} 
1, & \text{if } f(x) \geq \frac{\mu}{\rho} \text{ and } \mu > 0, \\
0, & \text{if } f(x) < \frac{\mu}{\rho} \text{ and } \mu > 0 \\
1, & \text{if } f(x) < \frac{\mu}{\rho} \text{ and } \mu < 0, \\
0, & \text{if } f(x) \geq \frac{\mu}{\rho} \text{ and } \mu < 0
\end{cases}
\]

**Lemma 15.** For any regression model \( f \) and dual variables \( \lambda \), the classifier \( h \) defined in Definition 22 is a “best response” in the sense that:

\[
h \in \arg \min_{h \in \mathcal{H}_4} L_f(h, \lambda).
\]

**B.1 Proofs from Section 3.1**

*Game formulation.* We pose the optimization of our original linear program as a zero-sum game between a primal (minimization) player who plays over the set of hypotheses and a dual (maximization) player who plays over the set of dual variables. The utility function of the game is the Lagrangian of our linear program as stated in Definition 8. The value of this game is given by

\[
\min_{h \in \mathcal{H}} \max_{\lambda \in \mathbb{R}^{|2G|}} L_f(h, \lambda).
\]

*Constraining the linear program.* In order to compute an approximate minimax solution to this game, we need to constrain the strategy space of the dual player.

That is, we need to bound the dual space to a region \( \Lambda = \{ \lambda \in \mathbb{R}^{|2G|} \mid ||\lambda||_1 \leq C \} \). We call this constrained version of the problem the \( \Lambda \)-bounded Lagrangian problem, which has value

\[
\min_{h \in \mathcal{H}} \max_{\lambda \in \Lambda \mid ||\lambda||_1 \leq C} L_f(h, \lambda). \tag{12}
\]

We can apply the minimax theorem to this bounded game to see:

\[
\min_{h \in \mathcal{H}} \max_{\lambda \in \Lambda \mid ||\lambda||_1 \leq C} L_f(h, \lambda) \equiv \max_{\lambda \in \Lambda \mid ||\lambda||_1 \leq C} \min_{h \in \mathcal{H}} L_f(h, \lambda).
\]

We will only be able to achieve an approximate solution to the problem, which we define as follows.

**Definition 23.** We say that \((h, \lambda)\) is a \( \nu \)-approximate minimax solution to the \( \Lambda \)-bounded Lagrangian problem \( L_f \) if \( L_f(h, \lambda) \leq \min_{h' \in \mathcal{H}} L_f(h', \lambda) + \nu \) and \( L_f(h, \lambda) \geq \max_{h' \in \mathcal{H}} L_f(h', \lambda) - \nu \).

An approximate minimax solution to this bounded version of the problem is also an approximate solution to the original problem we described in Equation 1.

**Theorem 3 ([18]).** Let \((h, \lambda)\) be a \( \nu \)-approximate minimax solution to the \( \Lambda \)-bounded Lagrangian problem \( L_f \) and let \( \text{OPT} \) be the objective value of the optimal solution to \( \psi(f, y, \mathcal{H}_4) \). Then, \( \text{err}(h) \leq \text{OPT} + 2\nu \), and \( \forall y \in \mathcal{G}, w_y |p_y(h) - p_y(\text{OPT})| \leq \gamma + (1 + 2\nu)/\mathcal{C} \).

**Approximate equilibrium of the constrained game.** Now, we can proceed with no-regret play to find an approximate solution to the game. The dual player will play projected gradient descent over their vector \( \lambda \) and the primal player will best respond, as described in Algorithm 4.

**Theorem 4.** Algorithm 4 returns an \( \epsilon \)-approximate equilibrium solution to the zero-sum game defined by Equation 12 after \( T = \frac{1}{4\epsilon^2} \left( \frac{1}{\epsilon} + 4|\mathcal{G}| \right)^2 \) rounds.

To prove this, we will use the following result from Freund and Shapire.
Theorem 5 ([9]). (Approximately solving a game). If $\lambda_1, \ldots, \lambda_T \in \Delta_\lambda$ is the sequence of distributions over $\lambda$ played by the dual player and $h_1, \ldots, h_T \in \mathcal{H}$ is the sequence of best-response hypotheses played by the primal player satisfying regret guarantees

$$\frac{1}{T} \max_{\lambda \in \Lambda} \sum_{t=1}^{T} U(h_t, \lambda) - \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\lambda \sim \lambda_t'} [U(h_t, \lambda)] \leq \Delta_1$$

and

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\lambda \sim \lambda_t} [U(h_t, \lambda)] - \frac{1}{T} \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{E}_{\lambda \sim \lambda_t} [U(h, \lambda)] \leq \Delta_2$$

then the time-average of the two players’ empirical distributions is a $(\Delta_1 + \Delta_2)$-approximate equilibrium.

Proof of Theorem 4. We follow the regret analysis of [24]. To instantiate their result, we need a bound on the norm of the gradients of the loss function and on the diameter of the feasible set $F$. First, we see that at each step the gradient of the loss seen by gradient descent is bounded:

$$\|\nabla \ell\|^2 = \sum_{g \in G} w_g (\rho_g - \rho - \gamma)^2 + w_g (-\rho_g + \rho - \gamma)^2 \leq 2|G|.$$  

Second, we see that if we consider the feasible set such that $\|\lambda\| \leq \frac{1}{2}$, then $\|F\|^2 = \frac{1}{2}$. Thus we have that the regret of the dual player is bounded:

$$\mathcal{R}(T) \leq \frac{\|F\|^2 \sqrt{T}}{2} + (\sqrt{T} - \frac{1}{2})\|\nabla \ell\|^2$$

$$\mathcal{R}(T) \leq \frac{1}{\lambda} \left( \frac{1}{2} \sqrt{T} \right) + (\sqrt{T} - \frac{1}{2})|G| \leq \frac{\|F\|^2 + 4|G|}{2\sqrt{T}}.$$  

After $T = \frac{1}{\lambda^2} \left( \frac{1}{2} \sqrt{T} + 4|G| \right)^2$ rounds, by [9] the average over empirical distributions of play of the dual and primal players, $\hat{\lambda}$ and $\hat{h}$, respectively, form an $\epsilon$-approximate equilibrium solution to the zero-sum game defined by $\mathcal{G}$.

Lemma 16. Let $h_t$ be the best response to $\lambda_{t-1}$ described in Algorithm 4 at some round $t \in [T]$. Then,

$$h_t(x) = s_{\lambda_{t-1}}(x, f(x)).$$

Proof. Recall from Lemma 15 and Algorithm 4 that the best response to $\lambda$ that the primal player can make is to compute $h$ based on the thresholding of the expression

$$\tau = \frac{k_{t-1}}{\mu_{t-1}}.$$  

Setting this threshold to be greater than or equal to some value $v$, note the following is implied:

$$\sum_{g \in G} \lambda_{g, t-1}(g(x) - \beta_g) - v \sum_{g \in G} \lambda_{g, t-1}(g(x) - \beta_g) \geq 2v - 1,$$

$$\Rightarrow (1 - v)(\sum_{g \in G} \lambda_{g, t-1}(g(x) - \beta_g) \geq 2v - 1,$$

$$\Rightarrow \langle \lambda_{t-1}, x_G - \beta \rangle = \sum_{g \in G} \lambda_{g, t-1}(g(x) - \beta_g) \geq \frac{2v - 1}{1 - v}.$$  

Thus, taking the indicator of

$$[\langle \lambda_{t-1}, x_G - \beta \rangle \geq d(v)]$$

is equivalent to determining if the threshold $\tau$ is greater than or equal to some $v$, and hence by the definition of $s_{\lambda_{t-1}}(x, v)$ in Definition 9 and of the best response $h$ in Definition 22, if $v$ is set to $f(x)$ it follows that

$$h(x) = s_{\lambda_{t-1}}(x, f(x)).$$

Theorem 6. Let $\text{OPT}$ be the objective value of the optimal solution to $\psi(f, \gamma, \mathcal{H}_A)$. Then, for any $C \in \mathbb{R}$, after $T = \frac{1}{4} \cdot C^2 \cdot (\overline{C}^2 + 4|G|)^2$ iterations, Algorithm 4 outputs a randomized hypothesis $\hat{h}$ with the following properties:

- the error of the output satisfies $\text{err}(\hat{h}) \leq \text{OPT} + \frac{\overline{C}}{2}$
- the constraint violation of the output satisfies $w_g|\rho_g(\hat{h}) - \rho(\hat{h})| \leq \gamma + \frac{\overline{C}}{2} + \frac{\overline{C}^2}{2}$
the output \( \hat{h} \) is the uniform mixture over \( T \) constituent models, each of which belong to the set of threshold functions \( \mathcal{B}(C) \).

Proof of Theorem 6. Applying Theorems 3 and 4, we have that after \( T \) rounds \((\hat{h}, \hat{\lambda})\) is an \( \epsilon \)-approximate equilibrium to the zero-sum game of 12 and equivalently a minimax solution to the \( \Lambda \)-bounded Lagrangian. Taking \( \epsilon = 1/C \), the solution \((\hat{h}, \hat{\lambda})\) is a \( \frac{C + 2/C^2}{1/C} = 1/C + 2/C^2 \) approximate solution to the original linear program 1. The final condition follows from Lemma 16.

Algorithm 4 Projected Gradient Descent Algorithm

1. Input: \( D \): dataset, \( f: \mathcal{X} \rightarrow [0, 1] \): regression function, \( \mathcal{G} \): groups, \( \gamma \): tolerance on fairness violation, \( C \): bound on dual (\( \|\lambda\|_1 \leq C \)), \( \eta \): learning rate
2. Initialize dual vector \( \lambda^0 = 0 \) and set \( T = \frac{1}{\eta} \cdot C^2 \cdot (C^2 + 4|\mathcal{G}|)^2 \).
3. for \( t = 1, \ldots, T \) do
   4. Primal player updates \( h_t \)
      
      \[
      h_t(x) = \begin{cases} 
      1, & \text{if } f(x) \geq \frac{\nu}{\mu_t + 1} \text{ and } \mu_{t-1} > 0, \\
      0, & \text{if } f(x) < \frac{\nu}{\mu_t + 1} \text{ and } \mu_{t-1} > 0, \\
      1, & \text{if } f(x) \leq \frac{\nu}{\mu_t + 1} \text{ and } \mu_{t-1} < 0, \\
      0, & \text{if } f(x) > \frac{\nu}{\mu_t + 1} \text{ and } \mu_{t-1} < 0, \\
      1, & \text{if } \mu_{t-1} = 0
      \end{cases}
      \]
   5. Compute
      
      \[
      \hat{\beta}_g = \mathbb{E}_{(x,y) \sim D} [f(h_t(x), 0)g(x)(1 - f(x))] \quad \text{for all } g \in \mathcal{G},
      \]
      
      \[
      \hat{\lambda} = \mathbb{E}_{(x,y) \sim D} [f(h_t(x), 0)(1 - f(x))], \quad \text{where } \beta_g = \mathbb{P}[g(x) = 1|y = 0]
      \]
6. Dual player updates
   
   \[
   \lambda_{g,t,+} = \max(0, \lambda_{g,t,+} + \eta \cdot (\hat{\beta}_g - \hat{\lambda} - \gamma)),
   \]
   
   \[
   \lambda_{g,t,-} = \max(0, \lambda_{g,t,-} + \eta \cdot (\hat{\beta}_g - \hat{\lambda} - \gamma)).
   \]
7. Dual player sets \( \lambda^t = \sum_{g \in \mathcal{G}} \lambda_{g,t,+} - \lambda_{g,t,-} \).
8. if \( \|\lambda^t\|_1 > C \) then
   9. set \( \lambda^t = \arg \min_{\lambda \in \mathbb{R}^{|\mathcal{G}|}, \|\lambda\|_1 \leq C} \|\lambda^t - \hat{\lambda}\|_2^2 \).
10. end if
11. end for
12. Output: \( \hat{h} := \frac{1}{T} \sum_{t=1}^{T} h_t \), a uniformly random classifier over all rounds’ hypotheses.

B.2 Proofs from Section 3.2

Theorem 7. Set \( C = \sqrt{1/\alpha} \). Let \( \hat{f} \) be \( \alpha \)-approximately mlt calibrated in expectation with respect to \( \mathcal{G}, \mathcal{H} \), and \( \mathcal{G} \times \mathcal{H} \) and \( \alpha \)-approximately jointly calibrated in expectation with respect to \( \mathcal{G} \times \mathcal{B}(C) \). Let \( \hat{h} \) be the result of running Algorithm 4 with input \( \hat{f} \) and \( C \). Then,

\[
\text{err}(\hat{h}) \leq \text{err}(h^*) + \alpha(5 + 2\sqrt{1/\alpha}) + 2\sqrt{\alpha}, \quad \text{and for all } g \in \mathcal{G}, \|g(h) - \rho(h)\| \leq \|g(h^*) - \rho(h^*)\| + 2\alpha.
\]

In order to prove this, we will proceed through the specifics of each line of the Proof Sketch 3.2 in Section 3.2 through Lemmas 17 through 3.

Lemma 17 (Equality in Equation 2).

\[
\text{err}(h^*) = L^*(h^*, \lambda^*)
\]

Proof. Consider the optimal solution \((h^*, \lambda^*)\) to \( \psi(f^*, \gamma, \mathcal{H}) \), and recall that \( \text{err}(\hat{h}) = \mathbb{E}_{\mathcal{X} \sim \mathcal{D}} [f^*(x)\ell(h(x), 1) + (1 - f^*(x))\ell(h(x), 0)] \).

Since the solution is optimal, it follows from complementary slackness, for each group \( g \) one of the following must hold: Either the constraint is exactly tight and so its “violation” term in the Lagrangian evaluates to 0, or its corresponding dual variables \( \lambda^*_g = 0 \). Thus, \( L^*_f(h^*, \lambda^*) \) simplifies to
\[ L^*_\ell(h^*, \lambda^*) = \mathbb{E}_{x \sim D_X} \left[ f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0) \right. \\
+ \left. 0 + \sum_{g \in G} \lambda^+_g \left( \ell(h(x), 0)g(x)(1 - f(x)) - \beta_g \ell(h(x), 0)(1 - f(x)) - \gamma \right) \right] \\
+ \left. 0 + \sum_{g \in G} \lambda^-_g \left( \beta_g \ell(h(x), 0)(1 - f(x)) - \ell(h(x), 0)g(x)(1 - f(x)) - \gamma \right) \right] \\
= \mathbb{E}_{x \sim D_X} \left[ f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0) \right. \\
\left. = \text{err}(h^*) \right] \\
\]

**Lemma 18 (Bounding Equation 2 by Equation 3).**

\[ L^*(h^*, \lambda^*) \geq L^*_\ell(h^*, \lambda^*). \]

**Proof.** This follows from the dual optimality condition that \( \lambda^* \in \arg \max_\lambda L^*(h^*, \lambda). \)

**Lemma 19. (Bounding Equation 3 by Equation 4) Fix any \( \lambda \). If \( \hat{f} \) is \( \alpha \)-multicalibrated with respect to \( G, H \), and \( G \times H = \{g(x) \cdot h(x) | g \in G, h \in H\} \), then then we have

\[ \|\hat{L}(h, \lambda) - L^*(h^*, \lambda)\| \leq \alpha (3 + 2\|\lambda\|_1). \]

**Proof.** Observe that we can write:

\[ \hat{L}(h, \lambda) = L_1(h, \lambda) - \gamma \sum_{g \in G} (\lambda^+_g + \lambda^-_g) = \hat{L}_2(h, \lambda), \]

where

\[ L_1(h, \lambda) = \mathbb{E}_{x \sim D_X} \left[ \ell(h(x), 0) \right], \]

\[ \hat{L}_2(h, \lambda) = \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \left( -\ell(h(x), 1) + \ell(h(x), 0) \right) \right]. \]

Similarly, we can write:

\[ L^*(h, \lambda) = L_1(h, \lambda) - \gamma \sum_{g \in G} (\lambda^+_g + \lambda^-_g) - L^*_2(h, \lambda), \]

where

\[ L^*_2(h, \lambda) = \mathbb{E}_{x \sim D_X} \left[ f^*(x) \left( -\ell(h(x), 1) + \ell(h(x), 0) \right) \right]. \]

Observe that the \( L_1 \) term does not depend on \( \hat{f} \) or \( f^* \) and so is common between \( \hat{L} \) and \( L^* \). We can bound \( \hat{L}_2 \) as follows:

\[ \hat{L}_2(h^*, \lambda) = \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \left( -\ell(h^*(x), 1) + \ell(h^*(x), 0) \right) \right] \\
= \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \left( -\ell(h^*(x), 1) + \ell(h^*(x), 0) \right) \right] \\
= \sum_{v \in R} P[\hat{f}(x) = v] \mathbb{E}_{x \sim D_X} \left[ \hat{f}(x) \left( -\ell(h^*(x), 1) + \ell(h^*(x), 0) \right) \right] \\
\leq \sum_{v \in R} P[\hat{f}(x) = v] \mathbb{E}_{x \sim D_X} \left[ f^*(x) \left( -\ell(h^*(x), 1) + \ell(h^*(x), 0) \right) \right] \\
+ \alpha \left( 3 + \sum_{g \in G} \lambda_g (1 + \beta_g) \right) \\
\leq L^*_\ell(h^*, \lambda) + \alpha (3 + 2\|\lambda\|_1). \]
where the first inequality follows from the fact that \( h^* \in \mathcal{H} \) and \( \hat{f} \) is multicalibrated with respect to \( \mathcal{G}, \mathcal{H} \), and \( \mathcal{G} \times \mathcal{H} \), which we verify below:

\[
\sum_{v \in R} \mathbb{P}[\hat{f}(x) = v] \mathbb{E}_{x \sim D_x} \left[ f^*(x) - \hat{f}(x) \right] \cdot \left( 1 - h^*(x) + h^*(x)(x) \right) \hat{f}(x) = v
\]

\[
= \sum_{v \in R} \mathbb{P}[\hat{f}(x) = v] \mathbb{E}_{x \sim D_x} \left[ f^*(x) - \hat{f}(x) \right] \cdot \left( -1 + 2h^*(x) + h^*(x) \sum_{g \in \mathcal{G}} \lambda_g (g(x) - \beta_g) \right) \hat{f}(x) = v
\]

\[
= - \sum_{v \in R} \mathbb{P}[\hat{f}(x) = v] \mathbb{E}_{x \sim D_x} \left[ \hat{f}^*(x) - \hat{f}(x) \right] \hat{f}(x) = v
\]

\[
+ 2 \sum_{v \in R} \mathbb{P}[\hat{f}(x) = v] \mathbb{E}_{x \sim D_x} \left[ f^*(x) - \hat{f}(x) \right] h^*(x) \hat{f}(x) = v
\]

\[
+ \sum_{v \in R} \mathbb{P}[\hat{f}(x) = v] \sum_{g \in \mathcal{G}} \lambda_g \mathbb{E}_{x \sim D_x} \left[ (f^*(x) - \hat{f}(x)) h^*(x) g(x) \right] \hat{f}(x) = v
\]

\[
- \sum_{v \in R} \mathbb{P}[\hat{f}(x) = v] \sum_{g \in \mathcal{G}} \lambda_g \beta_g \mathbb{E}_{x \sim D_x} \left[ (f^*(x) - \hat{f}(x)) h^*(x) \right] \hat{f}(x) = v
\]

\[
\leq 3\alpha + \sum_{g \in \mathcal{G}} \lambda_g (1 + \beta_g) \alpha
\]

\[
\leq 3\alpha + \alpha \sum_{g \in \mathcal{G}} \lambda_g (1 + \max_{g' \in \mathcal{G}} \beta_{g'})
\]

\[
\leq 3\alpha + \alpha \sum_{g \in \mathcal{G}} \lambda_g (1 + 1)
\]

\[
\leq 3\alpha + 2\|\lambda\|_1 \alpha
\]

Similarly, we can show that \( L^*(h^*, \lambda) - \hat{L}(h^*, \lambda) \leq \alpha (3 + 2\|\lambda\|_1) \). Putting everything together, we get that:

\[
\left| \hat{L}(h^*, \lambda) - L^*(h^*, \lambda) \right| \leq \alpha (3 + 2\|\lambda\|_1).
\]

This concludes the proof. \( \square \)

**Lemma 20 (Bounding Equation 4 by Equation 5).**

\[ L(h^*, \hat{\lambda}) \geq \hat{L}(\hat{h}, \hat{\lambda}) \]

**Proof.** This follows from the primal optimality condition that \( \hat{h} \in \text{arg min}_{h \in \mathcal{H}} \hat{L}(h, \hat{\lambda}) \) and that \( \mathcal{H} \subseteq \mathcal{H}_A \).

**Lemma 21 (Equality of Equation 5 and Equation 6).**

\[ \hat{L}(\hat{h}, \hat{\lambda}) = \hat{err}(\hat{h}) \]

**Proof.** This follows the same complimentary slackness argument as the proof of Lemma 17. \( \square \)

**Lemma 22 (Bound of Equation 6 by Equation 7).** Consider \( \hat{h} \) output by algorithm 4 after \( T = \frac{1}{4} \cdot C^2 \cdot (C^2 + 4|\mathcal{G}|)^2 \) rounds. Then,

\[ \hat{err}(\hat{h}) + 2/C \geq \hat{err}(\hat{h}) \]

**Proof.** This follows directly from Theorem 6. \( \square \)

We now have the tools to prove our main theorem.

**Proof of Theorem 7.** Applying lemmas 17 through 3 gives us

\[
err(h^*) = L^*(h^*, \lambda^*) \quad \text{(Lemma 17)}
\]

\[
\geq L^*(\hat{h}, \hat{\lambda}) \quad \text{(Lemma 18)}
\]

\[
\geq \hat{L}(\hat{h}, \hat{\lambda}) - \alpha (3 + 2\|\lambda\|_1) \quad \text{(Lemma 19)}
\]

\[
\geq \hat{L}(\hat{h}, \hat{\lambda}) - \alpha (3 + 2\|\lambda\|_1) \quad \text{(Lemma 20),}
\]

281
and
\[
\hat{L}(\hat{h}, \hat{\lambda}) = \text{err}(\hat{h}) \quad \text{(Lemma 21)}
\]
\[
\geq \text{err}(\hat{h}) - 2/C \quad \text{(Lemma 22)}
\]
\[
\geq \text{err}(\hat{h}) - 2/C - 2\alpha \quad \text{(Lemma 3)}.
\]

Putting this all together gives us
\[
\text{err}(h^*) \geq \text{err}(\hat{h}) - \alpha(3 + 2\|\lambda\|_1) - 2/C - 2\alpha
\]
\[
= \text{err}(\hat{h}) - \alpha(5 + 2\|\lambda\|_1) - 2/C
\]
\[
\geq \text{err}(\hat{h}) - \alpha(5 + 2C) - 2/C
\]

We want to set C to minimize this discrepancy. Noting that the derivative of \(\alpha(5 + 2C) + 2/C\) with respect to C is \(2\alpha - 2/C^2\), we get a minimization at \(C = \sqrt{1/\alpha}\).

Setting C as such gives the desired bound:
\[
\text{err}(h^*) \geq \text{err}(\hat{h}) - \alpha(5 + 2\sqrt{1/\alpha}) - 2\sqrt{\alpha}.
\]

Following a similar analysis as Lemma 3, we can bound the fairness constraints on \(\hat{h}\) by bounding them for the model \(\hat{h}_t\) found at every round \(t \in [T]\) of Algorithm 4.

\[
w_g \cdot (\rho_g(\hat{h}_t) - \rho(\hat{h}_t)) = \mathbb{E}_{x \sim D} \left[ (1 - \tilde{f}(x)) \tilde{t}(\hat{h}_t(x), 0) g(x) \right] - \mathbb{E}_{x \sim D} \left[ (1 - \tilde{f}(x)) \tilde{t}(\hat{h}_t(x), 0) \right] \beta_g
\]
\[
\leq \sum_{x \in R} \mathbb{P}[\tilde{f}(x) = v, \alpha_{l_x} - (x, v) = 0] \mathbb{E}_{x \sim D} \left[ (1 - \tilde{f}(x)) \tilde{t}(\hat{h}_t(x), 0) \cdot (g(x) - \beta_g) \right] \tilde{f}(x) = v, s_{\alpha_{l_x}}(x, v) = 1
\]
\[
= \sum_{x \in R} \mathbb{P}[\tilde{f}(x) = v, s_{\alpha_{l_x}}(x, v) = 1] \mathbb{E}_{x \sim D} \left[ (1 - \tilde{f}(x)) \tilde{t}(\hat{h}_t(x), 0) \cdot (g(x) - \beta_g) \right] \tilde{f}(x) = v, s_{\alpha_{l_x}}(x, v) = 1
\]
\[
= \sum_{x \in R} \mathbb{P}[\tilde{f}(x) = v, s_{\alpha_{l_x}}(x, v) = 1] \mathbb{E}_{x \sim D} \left[ \tilde{t}(\hat{h}_t(x), 0) \cdot (g(x) - \beta_g - \tilde{f}(x)g(x) + \tilde{f}(x) \beta_g) \right] \tilde{f}(x) = v, s_{\alpha_{l_x}}(x, v) = 1
\]
\[
= \sum_{x \in R} \mathbb{P}[\tilde{f}(x) = v, s_{\alpha_{l_x}}(x, v) = 1] \mathbb{E}_{x \sim D} \left[ (1 - \tilde{f}(x)) \tilde{t}(\hat{h}_t(x), 0) \cdot (g(x) - \beta_g) \right] \tilde{f}(x) = v, s_{\alpha_{l_x}}(x, v) = 1 + 2\alpha
\]
\[
= \mathbb{E}_{x \sim D} \left[ (1 - \tilde{f}(x)) \tilde{t}(\hat{h}_t(x), 0) \cdot (g(x) - \beta_g) \right] + 2\alpha = w_g \rho_g(\hat{h}_t) - \rho(\hat{h}_t) + 2\alpha.
\]

Here, the inequality comes from our multicalibration guarantees: (1) because we assumed \(G\) contains the \(l\), we can swap \(\tilde{f}(x)\) with \(f^*(x)\) at the cost of \(\alpha\) additive error, and (2) because we have joint multicalibration with respect to \(G \times B(C)\), we can swap \(\tilde{f}(x)g(x)\) with \(f^*(x)g(x)\) at the cost of \(\alpha\) additive error. We can repeat the same argument in the opposite direction, and get that
\[
w_g \left| \rho_g(h^*) - \rho(h^*) \right| \geq w_g \left| \rho_g(\hat{h}) - \rho(\hat{h}) \right| - 2\alpha.
\]

\[\square\]

C ACHIEVING JOINT MULTICALIBRATION

In this section we give an algorithm that can take as input any model \(\tilde{f} : X \to [0, 1]\) and transform it into a new model \(\hat{f} : X \to R\) such that \(\hat{f}\) achieves multicalibration in expectation with respect to a class of functions \(\mathcal{C}_1 \subset \{0, 1\}^X\) and simultaneously, joint multicalibration in expectation with respect to a class of functions \(\mathcal{C}_2 \subset \{0, 1\}^{X \times R}\) where \(R = \{0, \frac{1}{m}, \frac{2}{m}, \ldots, 1\}\) for some \(m > 0\). Our algorithm can be viewed as a variant of the original multicalibration algorithm of [14] (our variant achieves the stronger guarantee of calibration in expectation, first defined in [11]), or a simplification of the split-and-marge algorithm of [11], which replaces the “merge” operation with simple per-update rounding.

First we observe that without loss of generality, we can focus on achieving joint multicalibration for a single class of functions. To see this, note that given \(\mathcal{C}_1 \subset \{0, 1\}^X\), we can transform it into an identical class of two argument functions that simply ignore their second argument:

\[\mathcal{C}_1' = \{c \text{ where } c(x, v) = c_1(x) \text{ for every } c_1 \in \mathcal{C}_1\}.\]
Note that if \( \hat{f} \) is \( \alpha \)-approximately joint-multicalibrated with respect to \( C' \), then it is \( \alpha \)-approximately multicalibrated with respect to \( C_1 \) and vice versa. In other words, in order to be simultaneously multicalibrated with respect to \( C_1 \) and joint-multicalibrated with respect to \( C_2 \), it is sufficient (actually equivalent) to be joint-multicalibrated with respect to \( C'_1 \cup C'_2 \). Therefore, we focus on enforcing joint-multicalibration with respect to arbitrary \( C \subset \{0,1\}^{X \times [0,1]} \).

Before we describe the algorithm, we define the round operation. Write \( [\frac{1}{m}] = \{0, \frac{1}{m}, \frac{2}{m}, \ldots, 1\} \) for any \( m > 0 \). We let \( f' = \text{Round}(f, m) \) to denote the function that simply rounds the output of \( f \) to the nearest grid point of \( [\frac{1}{m}] \). Similarly, we write \( \text{Round}(v, m) = \arg \min_{v' \in [\frac{1}{m}]} |v' - v| \) to denote the grid point of \( [\frac{1}{m}] \) closest to \( v \).

**Algorithm 5 Multicalibration algorithm**

1. Input: \( \alpha, f, C \)
2. \( m = \frac{1}{2^\alpha} \)
3. \( f_0 = \text{Round}(f, m) \)
4. \( t = 0 \)
5. While there exists a \( c \in C \) such that:
   
   \[
   \sum_{x \in R} \mathbb{P}_{x \sim D_X} \left[ f_t(x) = v, c(x, v) = 1 \right] \left( v - \mathbb{E}_{(x,y) \sim D} \left( y|f_t(x) = v, c(x, v) = 1 \right) \right)^2 \geq \alpha
   \]

7. Let
   
   \[
   (u_t, c_t) = \arg \max_{v \in C, c \in C} \sum_{x \in R} \mathbb{P}_{x \sim D_X} \left[ f_t(x) = v, c(x, v) = 1 \right] \cdot \left( v - \mathbb{E}_{(x,y) \sim D} \left( y|f_t(x) = v, c(x, v) = 1 \right) \right)^2
   
   \]

8. Let
   
   \[
   S_t = \{x \in X : f_t(x) = v, c_t(x, u_t) = 1\}
   
   \]

9. \( \tilde{v} = \mathbb{E}_{(x,y) \sim D} (y|x \in S_t) \)

10. \( v'_t = \text{Round}(\tilde{v}, m) \)

11. \( f_{t+1}(x) = \begin{cases} v'_t & \text{if } x \in S_t \\ f_t(x) & \text{otherwise} \end{cases} \)

12. \( t = t + 1 \)
13. EndWhile

**Theorem 8.** The output of Algorithm 5 \( f_T : X \to \{0, \alpha, 2\alpha, \ldots, 1\} \) is \( \sqrt{\alpha} \)-approximately jointly multicalibrated with respect to \( C \) where \( T \leq \frac{4}{\alpha^2} \).

**Proof.** By definition, the output of the algorithm \( f_T \) is such that

\[
\sum_{x \in R} \mathbb{P}_{x \sim D_X} \left[ f_T(x) = v, c(x, v) = 1 \right] \left( v - \mathbb{E}_{(x,y) \sim D} \left( y|f_T(x) = v, c(x, v) = 1 \right) \right)^2 < \alpha
\]

for every \( c \in C \), meaning it satisfies \( \sqrt{\alpha} \)-joint calibration:

\[
\sum_{x \in R} \mathbb{P}_{x \sim D_X} \left[ f_T(x) = v, c(x, v) = 1 \right] \left| v - \mathbb{E}_{(x,y) \sim D} \left( y|f_T(x) = v, c(x, v) = 1 \right) \right|^2 < \sqrt{\alpha}
\]

So it suffices to show that the algorithm halts in less than \( T \leq \frac{4}{\alpha^2} \) rounds. Define

\[
B(f) = \mathbb{E}_{(x,y) \sim D} \left( (y - f(x))^2 \right).
\]

We use \( B \) as a potential function and show that we decrease it in each round in the following lemma.

**Lemma 23.** For every \( t < T \), \( B(f_{t+1}) - B(f_t) \leq -\frac{\alpha^2}{4} \).
PROOF. Define $\tilde{f}_t$ such that
\[
\tilde{f}_t(x) = \begin{cases} 
\hat{f}_t & \text{if } x \in B_t \\
 f_t(x) & \text{otherwise.}
\end{cases}
\]

Bounding (**): 
\[
B(f_{t+1}) - B(f_t) = \left( B(f_{t+1}) - B(\tilde{f}_t) \right) + \left( B(\tilde{f}_t) - B(f_t) \right)
\]

Bounding (*): 
\[
B(f_{t+1}) - B(\tilde{f}_t) = \mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ (y - f_{t+1}(x))^2 - (y - \tilde{f}_t(x))^2 \right] | x \in S_t |
\]
\[
= \mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ ((y - \tilde{f}_t(x)) + (\tilde{f}_t - f_t))^2 - (y - \tilde{f}_t(x))^2 \right] | x \in S_t |
\]
\[
= \mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ 2(y - \tilde{f}_t)(\tilde{f}_t - f_t) + (\tilde{f}_t - f_t)^2 \right] | x \in S_t |
\]
\[
\leq \mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] \cdot \frac{1}{4m^2}
\]
where the last inequality follows from the fact that $\tilde{f}_t = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|x \in S_t]$ and $|\tilde{f}_t - f_t| \leq \frac{1}{2m}$.

Bounding (**): Because in round $t$,
\[
\sum_{v \in R} \mathbb{P}_{x \sim \mathcal{D}_x} \left[ f_t(x) = v, c(x,v) = 1 \right] \left( v - \mathbb{P}_{(x,y) \sim \mathcal{D}}[y|f_t(x) = v, c(x,v) = 1] \right)^2 \geq \alpha,
\]
we must have
\[
\mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] (\tilde{f}_t - f_t)^2 = \mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] \left( \tilde{f}_t - \mathbb{P}_{(x,y) \sim \mathcal{D}}[y|x \in S_t] \right)^2 \geq \frac{\alpha}{m+1}.
\]

Now, we show that
\[
B(\tilde{f}_t) - B(f_{t+1}) = \mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ (y - \tilde{f}_t(x))^2 - (y - f_{t+1}(x))^2 \right] | x \in S_t |
\]
\[
= \mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ (y - \tilde{f}_t(x))^2 - ((y - \tilde{f}_t(x)) + (\tilde{f}_t - f_t))^2 \right] | x \in S_t |
\]
\[
= \mathbb{P}_{x \sim \mathcal{D}_x} \left[ x \in S_t \right] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ -2(y - \tilde{f}_t)(\tilde{f}_t - f_t) - (\tilde{f}_t - f_t)^2 \right] | x \in S_t |
\]
\[
\leq \frac{-\alpha}{m+1}
\]
where the last inequality follows from the fact that $\mathbb{E}_{(x,y) \sim \mathcal{D}}[y|x \in S_t] = \tilde{f}_t$.

Combining them together, we get
\[
B(f_{t+1}) - B(f_t) \leq \frac{1}{4m^2} - \frac{\alpha}{m+1}
\]
\[
= \frac{\alpha^2}{4} - \frac{\alpha}{\alpha + 1}
\]
\[
\geq \frac{\alpha^2}{4} - \frac{\alpha}{2}
\]
\[
= -\frac{\alpha^2}{4}.
\]

\[
\square
\]

Iterating Lemma 23 over $T$ rounds, we have
\[
B(f_T) \leq B(f_0) - T\frac{\alpha^2}{4}.
\]

Also, because $B(f) \in [0,1]$ for any $f$, it must be that $T \leq \frac{4}{\alpha^2}$. 

284
D OUT OF SAMPLE GUARANTEES

In the body of the paper, we assumed that we had direct access to distributional quantities — in particular, we needed to evaluate expectations over the feature distribution. In this section, we show that it is possible to estimate these quantities from modest amounts of unlabeled data sampled from the underlying distribution, and that the guarantees of our algorithm carry over to the underlying distribution. In particular, our algorithm results in a solution to the linear program that approximately satisfies its constraints on the underlying distribution, and achieves objective value that is approximately optimal within its comparison class. The strategy we take is to analyze a slightly modified algorithm (Algorithm 6), which at every stage, uses a fresh sample of data to evaluate the necessary expectations empirically. In particular, it uses a new sample at every iteration, and so has sample complexity that scales linearly with the number of iterations. Using techniques from adaptive data analysis [3, 6, 17] similar to how they are used by [14] to prove sample complexity bounds, we could reduce our linear dependence on $T$ in our sample complexity bound by a quadratic factor by reusing data across rounds, but we settle for the conceptually simpler bound here.

Theorem 9. Fix any distribution $\mathcal{D}$, hypothesis class $\mathcal{H}$, class of group indicators $\mathcal{G}$, dual bound $C$, and $\epsilon, \delta > 0$. After $T$ rounds, with probability $1 - \delta$, Algorithm 6 outputs a randomized hypothesis $h$ such that $\text{err}(h) \leq \text{OPT} + \frac{\epsilon}{T} + \frac{8\epsilon}{T}$ and $\text{err}(h) - \text{opt}(h) \leq \epsilon + \frac{1}{T} + \frac{2\epsilon}{C^2} + \frac{8\epsilon}{T}$, where $\text{OPT}$ is the objective value of the optimal solution of $\psi(f, y; \mathcal{H}_A)$. It makes use of $m = O(T \log(\frac{2\log(T)}{2\epsilon^2}))$ samples of unlabeled data drawn i.i.d. from $\mathcal{D}_X$. Here $T$ is as specified in the algorithm: $T = \frac{1}{4} \cdot C^2 \cdot (C^2 + 4|\mathcal{G}|)^2$.

Lemma 24. Fix any distribution $\mathcal{D}$, hypothesis class $\mathcal{H}$, and class of group indicators $\mathcal{G}$. In a single round $t$ of Algorithm 6 with $S_t \sim \mathcal{D}^m$ for $m = O(\frac{\log(\frac{2\log(T)}{2\epsilon^2})}{2\epsilon^2})$, Algorithm 6 returns a hypothesis $h_t$ that with probability $1 - \delta$ satisfies for all $g \in \mathcal{G}$:

$|\text{err}(h_t, g, \mathcal{D}) - \text{err}(h_t, g, S_t)| \leq \epsilon$

$|\rho(h_t, g, \mathcal{D}) - \rho(h_t, g, S_t)| \leq \epsilon$.

Algorithm 6 Projected Gradient Descent Algorithm

1. Input: $\mathcal{D}$; data distribution, $f : \mathcal{X} \rightarrow [0, 1]$; regression function, $\mathcal{G}$: groups, $y$: tolerance on fairness violation, $C$: bound on dual $\langle \|\lambda\|_1 \leq C \rangle$, $\eta$: learning rate, $m = \frac{\log(\frac{2\log(T)}{2\epsilon^2})}{2\epsilon^2}$: batch size of fresh data for each round of gradient descent, $\epsilon$: per round estimation error, $\delta$: failure probability

2. Initialize dual vector $\lambda^0 = 0$ and set $T = \frac{1}{4} \cdot C^2 \cdot (C^2 + 4|\mathcal{G}|)^2$.

3. for $t = 1, \ldots, T$ do

4. Primal player updates $h_t$

$$h_t(x) = \begin{cases} 1, & \text{if } f(x) \geq \frac{\nu_{t-1}}{\mu_{t-1}} \text{ and } \mu_{t-1} > 0, \\ 0, & \text{if } f(x) < \frac{\nu_{t-1}}{\mu_{t-1}} \text{ and } \mu_{t-1} > 0, \\ 1, & \text{if } f(x) \leq \frac{\nu_{t-1}}{\mu_{t-1}} \text{ and } \mu_{t-1} < 0, \\ 0, & \text{if } f(x) > \frac{\nu_{t-1}}{\mu_{t-1}} \text{ and } \mu_{t-1} < 0. \end{cases}$$

5. Sample $S_t$ i.i.d. from $\mathcal{D}^m$

6. Compute

$$\hat{p}_{t,g} = \mathbb{E}_{(x,y) \sim S_t} [f(h_t(x), 0)g(x)(1 - f(x)) \mid \forall g \in \mathcal{G}]$$

$$\hat{p}_t = \mathbb{E}_{(x,y) \sim S_t} [\beta_{t}f(h_t(x), 0)(1 - f(x)) \mid \beta_{t} = \mathbb{P}[g(x) = 1|y = 0]]$$

7. Dual player updates

$$\lambda_{g,t,+} = \max(0, \lambda_{g,t,+} + \eta \cdot (\hat{p}_{t,g} - \hat{p}_t - \gamma))$$

$$\lambda_{g,t,-} = \max(0, \lambda_{g,t,-} + \eta \cdot (\hat{p}_t - \hat{p}_{t,g} - \gamma))$$

8. Dual player sets $\lambda_t = \sum_{g \in \mathcal{G}} \lambda_{g,t,+} - \lambda_{g,t,-}$.

9. if $\|\lambda_t\|_1 > C$ then

10. set $\lambda^T = \arg \min_{\{\lambda \in \mathbb{R}^d : \|\lambda\|_1 \leq C\}} \|\lambda_t - \lambda\|_2^2$.

11. end if

12. end for

13. Output: $\tilde{h} = \frac{1}{T} \sum_{t=1}^{T} \hat{h}_t$, a uniformly random classifier over all rounds’ hypotheses.
Theorem 10 (Chernoff-Hoeffding Bound). Let $X_1, X_2, \ldots, X_m$ be i.i.d. random variables with $a \leq X_i \leq b$ and $E[X_i] = \mu$ for all $i$. Then, for any $\alpha > 0$,

$$\Pr\left( \left| \frac{\sum X_i}{m} - \mu \right| > \alpha \right) \leq 2 \exp\left( -\frac{2\alpha^2 m}{(b-a)^2} \right).$$

Proof of Lemma 24. This claim follows by applying a Chernoff-Hoeffding bound with $m \geq \frac{\ln(2\tau G)}{2\epsilon^2}$.

Proof Sketch of Theorem 9. Taking $m > \frac{\log(2\tau G)}{2\epsilon^2}$, we have that in a single round $t$ of our algorithm we are able to estimate the true distributional classification and fairness constraint errors up to an additive error of $\epsilon$ with probability $1 - \delta/T$ and hence with probability $1 - \delta$, we estimate these quantities up to additive error $\epsilon$ uniformly over all $T$ rounds. We can then make one small modification to the analysis of Algorithm 4. First observe that since the primal player’s best response does not depend on any estimation of a distributional quantity based on the sample $S_t$, their regret is still zero, as it is in the analysis of Algorithm 4. The dual player, on the other hand, is given loss vectors that deviate from the versions that would have been computed on the underlying distribution by at most $2\epsilon$ in $\ell_\infty$ norm, and hence experience additional regret (to the true distributional quantities) larger than in the analysis of Algorithm 4 by up to an additional additive $4\epsilon$. Consequently, the equilibrium solution $(\bar{h}, \bar{\lambda})$ from Algorithm 6 is an $4\epsilon + 1/C$ approximate equilibrium to the zero-sum game of 12 which then, applying Theorem 3, yields a $\frac{2}{C} + 8\epsilon$ approximate solution to the objective of the original linear program.