Calibration for the (Computationally-Identifiable) Masses

Úrsula Hébert-Johnson*
Stanford University

Michael P. Kim†
Stanford University

Omer Reingold‡
Stanford University

Guy N. Rothblum§
Weizmann Institute

Abstract

As algorithms increasingly inform and influence decisions made about individuals, it becomes increasingly important to address concerns that these algorithms might be discriminatory. The output of an algorithm can be discriminatory for many reasons, most notably: (1) the data used to train the algorithm might be biased (in various ways) to favor certain populations over others; (2) the *analysis* of this training data might inadvertently or maliciously introduce biases that are not borne out in the data. This work focuses on the latter concern.

We develop and study *multicalbration* – a new measure of algorithmic fairness that aims to mitigate concerns about discrimination that is introduced in the process of learning a predictor from data. Multicalibration guarantees accurate (calibrated) predictions for every subpopulation that can be identified within a specified class of computations. We think of the class as being quite rich; in particular, it can contain many overlapping subgroups of a protected group.

We show that in many settings this strong notion of protection from discrimination is both attainable and aligned with the goal of obtaining accurate predictions. Along the way, we present new algorithms for learning a multicalibrated predictor, study the computational complexity of this task, and draw new connections to computational learning models such as agnostic learning.

*uhebertj@stanford.edu

†mpk@cs.stanford.edu. Supported in part by NSF grant CNS-122864. Part of this work was completed while the author was visiting VMWare Research Group.

‡reingold@stanford.edu. Supported in part by NSF grant CCF-1749750.

§rothblum@alum.mit.edu

1 Introduction

Fueled by rapidly growing data sets and by breakthroughs in machine learning, algorithms are informing decisions that affect all aspects of life. From news article recommendations to criminal sentencing decisions to healthcare diagnostics, increasingly algorithms are used to make predictions about individuals. Often, the predictions of an algorithm form the basis for deciding how to treat these individuals (suggesting a conservative Op-Ed, approving early parole, or initiating chemotherapy). A potential risk is that these algorithms might discriminate against groups of individuals that are protected by law or by ethics. This paper aims to mitigate such risks of algorithmic discrimination.

We consider algorithms that predict the probabilities of events occurring for individuals. For example, a financial institution may be interested in predicting the probability that an individual will repay a mortgage. The institution may have at its disposal a large array of information for each individual (as well as historic data and global information such as financial and political trends). But as thorough as the company may be, a significant level of uncertainty will remain. Just as we wouldn't expect to be able to predict with absolute certainty whether it will rain on a particular day a year from now, the financial institution wouldn't expect to predict with absolute certainty whether an individual will repay a loan. Thus, we consider algorithms that output, for every individual i, a prediction x_i of the probability that the event will occur for i; we call the mapping from individuals to probabilities a *predictor*.

Our focus in this paper is mitigating biases that may arise as an algorithm *analyzes* given data – specifically, as the algorithm learns a predictor from data. Continuing the above example, suppose that in a particular protected community S, on average, individuals are financially disadvantaged and are unlikely to repay a loan. A machine-learning algorithm that aims to optimize the institution's returns might devote resources to learning outside of S – where there is more opportunity for gains in utility – and assign a fixed, low probability to all $i \in S$. Such an algorithm would discriminate against the *qualified* members of S. If S is an underrepresented subpopulation, this form of discrimination has the potential to amplify S's underrepresentation by refusing to approve members that are capable of repaying the loan.

Focusing on such concerns, our primary contributions are as follows:

- We develop and study *multicalibration*, a new measure of algorithmic fairness aimed at mitigating concerns about discrimination that arises in the process of learning a predictor from given data. In a nutshell, multicalibration guarantees highly-accurate predictions for every group of individuals that can be identified by a specified (and often bounded) computational class $\mathcal{C}$. In the mortgage repayment example above, if the class of qualified members of S can be identified by a circuit $c \in \mathcal{C}$, then the predictions made on qualified members of S must be accurate, and the prediction algorithm cannot ignore / discriminate against these individuals. We emphasize that the class $\mathcal{C}$ can be quite rich and, in particular, can contain many overlapping subgroups of a protected group S.
- We present a general-purpose algorithm for learning a predictor that is multicalibrated with respect to *any* given class $\mathcal{C}$. The complexity of evaluating the predictor is only slightly larger than the complexity of evaluating circuits in $\mathcal{C}$. The learning algorithm's running time

depends linearly on the size of $\mathcal{C}$.

- We also study the *computational complexity* of learning multicalibrated predictors for more structured classes $\mathcal{C}$. We show a strong connection between the complexity of learning a multicalibrated predictor and agnostic learning [Hau92, KSS94]. In the positive direction, if there is an efficient (weak) agnostic learner [KMV08, Fel10] for a class $\mathcal{C}$, then we can achieve similarly efficient multicalibration with respect to sets defined by $\mathcal{C}$. In the other direction, we show that learning a multicalbirated predictor on sets defined by $\mathcal{C}$ is as hard as weak agnostic learning on $\mathcal{C}$. In this sense, the complexity of learning a multicalibrated predictor with respect to a class $\mathcal{C}$ is equivalent to the complexity of weak agnostic learning on $\mathcal{C}$.
- Finally, we demonstrate that multicalbration can go hand-in-hand with the goal of achieving high-utility predictors. In particular, given a predictor h, we can use post-processing to obtain a multicalibrated predictor x whose accuracy is no worse than that of h (accuracy is measured in ℓ_2^2 distance from the benchmark p^*). The complexity of evaluating the predictor x is only slightly larger than that of h.

High-level setting. For an individual i from the population $\mathcal{X}$, we denote i's outcome by $o_i \in \{0,1\}$. We take $p_i^* \in [0,1]$ to be the probability of outcome $o_i = 1$, conditioned on all the information which is available to the algorithm. We denote by p^* the vector of these probabilities for all individuals in $\mathcal{X}$. Our goal is to make a prediction x_i for the value of p_i^* for every individual i. As discussed above, we would like to avoid additional malicious or inadvertent discrimination (beyond the biases contained in the data). Thus, we refer to p^* as the *benchmark predictor* for measuring discrimination.

Organization. We begin by elaborating on our setting. The remainder of the introduction is structured as follows. In Section 1.1 we elaborate on the notion of multicalibration and on its relationship to other notions in the larger context of fairness. We outline our main results on learning multicalibrated predictors in Section 1.2. We further elaborate on related work and on future directions in Section 1.3. Finally, we provide a brief overview of techniques in Section 1.4.

1.1 Calibration, Multicalibration and Balance

Calibration vs. balance If we do not want a predictor x to downplay the fitness of a group S, we can require that it be (approximately) accurate in expectation over S; namely, that $\left|\mathbb{E}_{i\sim S}\left[x_i - p_i^*\right]\right| \leq \alpha$, where $\alpha \geq 0$ is small. This means that the expectation of x and p^* over S are almost identical. Calibration, introduced as a fairness concept in [KMR16], strengthens this requirement by essentially asking that for any particular value v, if we let $S_v = \{i \in S : x_i = v\}$ be the subset of S of individuals with predicted probability v, then $\left|\mathbb{E}_{i\sim S_v}\left[x_i - p_i^*\right]\right| = |v - \mathbb{E}_{i\sim S_v}[p_i^*]| \leq \alpha$.[1]

[1]Calibration is often defined in the literature with respect to the instantiations of the events rather than their probabilities. Namely, that for every v, the fraction of i in S_v with $o_i = 1$ is roughly v. As long as S_v is sufficiently large and the instantiations are sufficiently independent, the two definitions are equivalent (up to small sampling errors) by concentration bounds. Necessarily, the formal definition given in Section 2 will allow for a small fraction of the elements in S to be misclassified due to being in a small set S_v.

While calibration already precludes some forms of discrimination, as a group notion of fairness, it still allows for others (even if we assume that p^* is perfectly fair). Indeed, weaknesses of group notions of fairness were discussed in [DHP+12] (for a somewhat related notion called statistical parity), as a motivation for introducing an individual notion of fairness (see further discussion and comparisons below). A specific way to discriminate while satisfying calibration is to assign every member of S the value $\mathbb{E}_{i\sim S}[p_i^*]$. While being perfectly calibrated over S, the qualified members of S with large values p_i^* will be hurt.

With this motivation and additional considerations in mind, other notions of fairness have been studied, looking at the rate of false positives and false negatives of predictions. Several variants of such properties have been recently studied on their own and in connection to calibration [KMR16, Cho17,PRW+17,HPS16,WS17,CDPF+17]. Let us briefly consider the notions referred to as balance in [KMR16]: *balance for the positive class* — the expected prediction x_i for yes instances ($o_i = 1$) in group S equals the expected prediction for yes instances outside of S; and *balance for the negative class* — the expected prediction for no instances ($o_i = 0$) in group S equals the expected prediction for no instances outside of S.

While both calibration and balance (as well as other related variants) intuitively seem like good properties to expect in a fair predictor (even if they are a bit weak), it has been shown that calibration and balance are impossible to obtain together (in non-degenerate cases) [KMR16,Cho17, PRW+17]. In [HPS16] it is shown how to obtain equalized odds, a definition related to error-rate balance, as a post-processing step of "correcting" any other predictor. Additional study into such post-processing approaches was done in [WS17]. (See further discussion of "corrective discrimination" below.)

Our approach in this paper towards mitigating the conflict between calibration and balance is to strengthen the protections implied by calibration, rather than enforcing balance.

Multicalibration As mentioned, balance and calibration are often at odds. In our setting, the benchmark predictor p^* itself is unlikely to be balanced. Balance is therefore inconsistent with our goal of approaching p^*. Indeed, in our setting the fact that balance is not satisfied might simply be an artifact of the inherent randomness in the process of sampling the outcome o_i, and this motivates our definition of multicalibration.

To illustrate this point, consider the following (intentionally artificial) example: an algorithm is tasked with predicting the probability of rain during 10 days of winter, in two cities, a year from now. In city A the algorithm predicts rain on each day with probability 0.8; in city B it predicts rain with probability 0.2 (note that the certainty of the predictions is identical for both cities). A year passes and indeed in city A it rains on 8 of the days, whereas in city B it rains on only 2 days. What surprising accuracy! Nevertheless, the predictions violate balance and indeed, the mayor of city A complains that the predictor hurts tourism to her city: "our sunny days are just as sunny as the sunny days of city B, so why were you so much more pessimistic about ours?" The point we are making here is that the sunny days in A are not necessarily *a priori* different than the rainy days. Given the inherent uncertainty, it is unreasonable to expect accuracy on a subgroup that is *only identifiable a posteriori*. In this sense, different false negative (or false positive) rates between groups are not necessarily a sign of discrimination.

More generally, consider an algorithm that produces a predictor x. The values o_i are determined,

and then an auditor comes up with a set S' that over-performed compared with the predictions of x. Perhaps the learning algorithm was lazy and neglected to identify the higher potential in S'? Perhaps the individuals of S' were simply lucky? How can we tell? To answer these questions, we take the following perspective: on the one hand, we can only expect a learner to produce a predictor that is calibrated on sets that could have been identified *efficiently* from the data at hand; on the other hand, we expect the learner to produce a predictor that is calibrated on *every* efficiently-identifiable subset. This motivates our definition of multicalibration, which loosely says:

A predictor x is α-multicalibrated with respect to a family of sets $\mathcal{C}$ if it is α-calibrated with respect to every $S \in \mathcal{C}$.

In the spirit of the discussion above, we take $\mathcal{C}$ to be a family of (sufficiently large) sets of individuals, such that for every $S \in \mathcal{C}$, the predicate $i \in S$ can be evaluated from the individual's data within a particular family of computations (circuits of quadratic size, conjunctions of four attributes, or any other bounded complexity class). The more powerful the family, the stronger the guarantee becomes; no subpopulation that can be identified by the family will be overlooked. At the extreme, consider multicalibration with respect to the family of polynomial size circuits; in this case, every efficiently-identifiable subpopulation is protected! Note that the subpopulations in $\mathcal{C}$ can be overlapping, with complex relationships. In particular, they may well have no explicit dependence on sensitive attributes. In this sense, multicalibration goes far beyond calibration for several sensitive groups.

1.2 Our Results

Our study of multicalibration follows two major themes:

- We investigate the feasibility of obtaining multicalibration; specifically, we study the learnability of multicalibrated predictors, showing both positive and negative results.
- We investigate the properties of multicalibrated predictors. While multicalibration provides strong guarantees against forms of discrimination, we show that this protection can come at little cost in terms of complexity and utility.

We begin with a high-level overview of our setup. For a formal description of our model and assumptions, see Section 2. Suppose that for some universe of individuals $\mathcal{X}$, we wish to predict whether some event (ad click, loan repayment, cancer diagnosis, etc.) will occur for each individual $i \in \mathcal{X}$. We assume that for each individual, there is some true underlying probability p_i^* that the event will occur. We call any mapping from the universe to probabilities a *predictor*; formally, a predictor is a function[2] $x : \mathcal{X} \to [0, 1]$ that maps individuals from the universe to estimates of their true probabilities. We denote by p^* the benchmark predictor that gives the true probabilities.

The benchmark predictor is, itself, multicalibrated with respect to every collection of subsets $\mathcal{C}$. Thus, if we can efficiently learn a predictor from the data at hand with sufficient accuracy across the

[2] We will interchange between function and vector notation; generally we will denote the prediction that x assigns to an individual $i \in \mathcal{X}$ as x_i.

entire population (specifically, with small ℓ_1 distance from p^*), then the learned predictor will be multicalibrated. That said, in most interesting situations, p^* will be too complex to learn efficiency with such uniform accuracy (especially given that the values of p^* themselves will usually not be observable). Focusing on such settings, we aim for multicalibration (see above) as a notion of protection from discrimination.

The first question to address is whether multicalibration is feasible. For instance, it could be the case that the requirements of multicalibration are so strong that they would require learning and representing an arbitrarily complex function p^* exactly, which we've established can be infeasible. Our first result characterizes the complexity of representing a multicalbrated predictor. We demonstrate that multicalibration indeed can be achieved efficiently: for any p^* and any collection of large subsets $\mathcal{C}$, there exists a predictor that is α-multicalibrated on $\mathcal{C}$, whose complexity is only slightly larger than the complexity required to describe the sets of $\mathcal{C}$. For concreteness, we use circuit size as our measure of complexity in the following theorem.

Theorem 1. *Suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is collection of sets where for $S \in \mathcal{C}$, there is a circuit of size s that computes membership in S and $|S| \geq \gamma |\mathcal{X}|$. For any $p^* : \mathcal{X} \to [0,1]$, there is a predictor that is α-multicalibrated with respect to $\mathcal{C}$ implemented by a circuit of size $O(s/\alpha^4\gamma)$.*

As stated, this result claims the existence of multicalibrated predictors whose predictions are efficient to evaluate. The existence of such predictors, while interesting from a complexity-theoretic perspective, begs the more practical question of whether we can get our hands on such a predictor.

In fact, Theorem 1 is a corollary of our main result – an algorithm for learning α-multicalibrated predictors from labeled samples. While our model assumes the existence of some underlying true probabilities p^*, in most applications, these probabilities will not be directly observable. As such, we design algorithms that learn predictors from samples of individuals labeled with their *outcomes*; specifically, we assume access to labeled samples (i, o_i) of individual-outcome pairs, where i is sampled according to some distribution $\mathcal{D}$ on the universe and o_i is the realized value of a Bernoulli trial with probability p^*_i. Naturally, in this model, our goal is to give algorithms that are efficient in terms of running time and sample complexity.

In Section 3, we give an algorithm for learning a multicalibrated predictor from labeled samples, whose running time scales linearly with $|\mathcal{C}|$ and polynomially with α and γ. A consequence of our analysis is that naively, the sample complexity can be upper-bounded by $\log(|\mathcal{C}|)/\alpha^6\gamma^6$. We show how to improve the sample complexity over the naive approach by polynomial factors in both α and γ.

Theorem 2. *Suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is collection of sets such that for all $S \in \mathcal{C}$, $|S| \geq \gamma |X|$, and suppose set membership can be evaluated in time t. Then there is an algorithm that learns a predictor of $p^* : \mathcal{X} \to [0,1]$ that is α-multicalibrated on $\mathcal{C}$ from $O(\log(|\mathcal{C}|)/\alpha^{11/2}\gamma^{3/2})$ samples in time $O(|\mathcal{C}| \cdot t \cdot \text{poly}(1/\alpha, 1/\gamma))$.*

Observing the linear dependence in the running time on $|\mathcal{C}|$, it is natural to try and develop a learning procedure with subpolynomial, or even polylogarithmic, dependence on $|\mathcal{C}|$. Our next results aim to characterize when this optimistic goal is possible – and when it is not. We emphasize that the algorithm of Theorem 2 learns a multicalibrated predictor for *arbitrary* $p^* : \mathcal{X} \to [0,1]$ and $\mathcal{C}$. In the setting where we cannot exploit structure in p^* to learn efficiently, we might hope to

exploit structure, if it exists, in the collection of subsets $\mathcal{C}$. Indeed, we demonstrate a connection between our goal of learning a multicalibrated predictor and weak agnostic learning, introduced in the literature on agnostic boosting [BDLM01, KMV08, KK09, Fel10]. Our next result shows that efficient weak agnostic learning over $\mathcal{C}$ implies efficient learning of α-multicalibrated predictors on $\mathcal{C}$.

Theorem 3 (Informal)**.** *If there is a weak agnostic learner for $\mathcal{C}$ that runs in time T, then there is an algorithm for learning an α-multicalibrated predictor on $\mathcal{C}' = \{S \in \mathcal{C} : |S| \gamma |X|\}$ that runs in time $O(T \cdot \text{poly}(1/\alpha, 1/\gamma))$.*

Slightly more formally, we require a (ρ, τ)-weak agnostic learner in the sense first introduced by [KMV08] and generalized by [Fel10]. For the specifics of the requirements and parameters, see the formal statement in Section 4.

These results show that under the right structural assumptions on p^* or on $\mathcal{C}$, a multicalibrated predictor may be learned more efficiently than our upper bound for the general case. Returning to the general case, we may wonder if these structural assumptions are necessary; we answer this question in the positive. We show that for worst-case p^* learning a multicalibrated predictor on $\mathcal{C}$ is as hard as weak agnostic learning for the class $\mathcal{C}$.

Theorem 4 (Informal)**.** *If there is an algorithm for learning an α-multicalibrated predictor on a collection of sets $\mathcal{C}' = \{S \in \mathcal{C} : |S| \geq \gamma N\}$ that runs in time T, then there is an algorithm that implements a (ρ, τ)-weak agnostic learner in time $O(T \cdot \text{poly}(1/\tau))$ for any $\rho, \tau > 0$ such that $\tau \leq \min\{\rho - 3\alpha, \rho - 6\alpha\}$.*

In general, agnostic learning is considered a notoriously hard computational problem. In particular, under cryptographic assumptions [Val84, GGM84, BR17], this result implies that there is some constant $t > 0$, such that any algorithm that learns an α-multicalibrated predictor requires $\Omega(|\mathcal{C}|^t)$ time.

Finally, we return our attention to investigating the utility of multicalibrated predictors. Above, we have argued that multicalibration provides a strong protection of groups against discrimination. We show that this protection comes at (next to) no cost in the utility of the predictor. This result adds to the growing literature on fairness-accuracy trade-offs [FKL16, BHJ+17, CG17].

Theorem 5. *Suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is a collection of subsets of $\mathcal{X}$ and $\mathcal{H}$ is a set of predictors. There is a predictor x that is α-multicalibrated on $\mathcal{C}$ such that*

$$\mathop{\mathbb{E}}_{i \sim \mathcal{X}}[(x_i - p^*_i)^2] - \mathop{\mathbb{E}}_{i \sim \mathcal{X}}[(h^*_i - p^*_i)^2] < 6\alpha,$$

where $h^* = \text{argmin}_{h \in \mathcal{H}} \mathbb{E}_{i \sim \mathcal{X}}[(h - p^*)^2]$. Further, suppose that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$, and suppose that set membership for $S \in \mathcal{C}$ and $h \in \mathcal{H}$ are computable by circuits of size at most s; then x is computable by a circuit of size at most $O(s/\alpha^4\gamma)$.

We can interpret Theorem 5 in different ways based on the choice of $\mathcal{H}$. Suppose there is some sophisticated learning algorithm (say, a neural network) that produces some predictor h that obtains exceptional performance, but may violate calibration arbitrarily. If we take $\mathcal{H} = \{h\}$, then this result says: enforcing calibration on h after learning does not hurt the accuracy by much. Further,

our proof will demonstrate that if calibration changes the predictions of h significantly, then this change amounts to an *improvement* in accuracy. See Lemma 5.2 for the exact statement. Taking a different perspective, we can also think of $\mathcal{H}$ as a set of predictors that, say, are implemented by a circuit class of bounded complexity (e.g. conjunctions of k variables, halfspaces, circuits of size s). This theorem shows that for any such class of predictors $\mathcal{H}$ of bounded complexity, there exists a multicalibrated predictor with similar complexity that performs as well as the best $h^* \in \mathcal{H}$. In this sense, with just a slight overhead in complexity, multicalibrated predictors can achieve "best-in-class" predictions.

1.3 More on related work and future directions

Between populations and individuals In [DHP+12], an individual notion of fairness was defined, referred to as "fairness through awareness." This notion relied on a task-specific metric of similarity between individuals and formalized the idea that similar individuals (under this metric) are treated similarly. It is natural, in an array of applications, to view p^* as defining a metric – two individuals i and j will be assigned the distance $|p^*_i - p^*_j|$. In this work we consider cases in which figuring out p^* in its entirety is difficult. Thus, one can view our approach as a meaningful compromise between group fairness (satisfying calibration) and individual-calibration (closely matching p^*_i). It seems plausible that our approach could be applied to more general notions of similarity metrics, and we intend to explore this in future work.

Subgroup Fairness Contemporary independent work of [KNRW17] also investigates strengthening the guarantees of notions of group fairness by requiring that these properties hold for a much richer collection of sets. Unlike our work, their definitions require balance or statistical parity on these collection of sets. Their motivation is similar to ours, namely to bridge the gap between notions of individual fairness (powerful but hard to obtain) and population-level fairness (easy to work with but weak).

Despite similar motivations, the two approaches to subgroup fairness differ in substantial ways. As a concrete example, Theorem 5 demonstrates that achieving multicalibration is aligned with the incentives of achieving high-utility predictors; this is not necessarily the case with balance-based notions of fairness. Indeed, in the setting considered in this work, one of the motivations for multicalibration is a critique of balance that may only be heightened when considering "multi-balance". Consider the example in [DHP+12] where in a population S the strongest students apply to Engineering whereas in the general population T they apply to Business. Even if predictor p^* (for the probability of success in school) is balanced between S and T, forcing balance between the two populations within Business applicants and within Engineering applicants would be unfair to qualified applicants in both groups. (For more discussion, see the section below on "Corrective discrimination".)

On a technical level, both works draw connections between agnostic learning and the task of finding a group on which the fairness condition is violated ([KNRW17] refer to this as auditing). Leveraging this connection, we show how to use an agnostic learner to learn a multicalibrated predictor efficiently. [KNRW17] also leverage this connection: they derive an algorithm that converges to the best (most accurate) distribution over classifiers in a given class, given an oracle that solves the

learning (and auditing) tasks. While the convergence rate of their algorithm is not known to be polynomial, they implement it and show that it performs well in practice.

Preventing discrimination by algorithms is subtle; different scenarios will call for different notions of protection. As such, it is hard to imagine a universal definition of fairness. Nevertheless, these two independent works validate the need to investigate attainable approaches to mitigating discrimination beyond large protected groups. It will be interesting to further understand how these notions of subgroup fairness relate to one another, and when each approach is most appropriate.

Calibration for multiple protected sets vs. multicalibration The literature on fairness commonly considers more than a single protected set (cf. [CSV17] which studies fairness in ranking algorithms, with protected groups being defined by each attribute of interest). The major difference in our work is that we think of protected groups as any group that can be efficiently identified, rather than those defined by sensitive properties. One side benefit of the generality of our approach is that it does not single out groups based on sensitive values for special treatment (which can be illegal in some contexts).

Calibration, Bandits, and Regret There is a growing literature on designing fair selection policies in the multi-arm bandits setting [JKMR16, JKM$^+$17, LRD$^+$17]. Recently [LRD$^+$17] initiated the study of calibration in this context. Motivated by the aforementioned work of [DHP$^+$12], their notion of calibrated fairness aims to "treat similar individuals similarly" by designing sampling policies that have low *fairness regret*.

Causality and discrimination in the data Discrimination can occur at any stage throughout the algorithmic pipeline and in many forms. The most important aspect of developing a theory of algorithmic fairness that multicalibration does not address is "unfair" data (see more below). Kilbertus *et al.* [KRCP$^+$17] voice an important criticism of any notion of fairness based solely on observational criteria, as discrimination can occur through unresolved causal relationships. This criticism applies to the notions of balance, calibration, and multicalibration, which all depend only on the joint distribution of predictions, outcomes, and features.

Rather than attempting to provide a general-purpose definition of fairness, our work tackles a particular concern about discrimination that can occur as part of the process of learning a predictor from given data. In this context, we believe that multicalibration provides an important anti-discrimination guarantee.

Corrective discrimination Let us look deeper into the biases that may be present in the gathered data, by considering the mortgage example again: perhaps the number of members of S that received loans in the past is small (and thus there are too few examples for fine-grained learning within S); perhaps the attributes are too limited to identify the qualified members of S (taking this point to the extreme, perhaps the only available attribute is membership in S). In these cases, the data may be insufficient for multicalibration to provide meaningful guarantees. Further, even if the algorithm was given access to unlimited rich data such that refined values of p^* could be recovered, there are situations where preferential treatment may be in order: after all, the salaries

of members of S may be lower due to historical discrimination.

For these reasons, our concern that balance is inconsistent with p^* could be answered with: "yes, and purposely so!" Indeed, [HPS16] promotes enforcing a balance-related property, called equalized odds, as a form of "corrective discrimination." While this type of advocacy is important in many settings, multicalibration represents a different addition to the quiver of anti-discrimination measures, which we also believe is natural and desirable in many settings.

Consider a concrete example where multicalibration is appropriate, but equalizing error rates might not be: Suppose a genomics company offers individuals a prediction of their likelihood of developing certain genetic disorders. These disorders have different rates across different populations; for instance, Tay-Sachs disease is rare in the general population, but occurs much more frequently in the Ashkenazi Jewish population. We certainly do not want to enforce corrective discrimination on the Ashkenazi population by down-weighting the prediction that individuals would have Tay-Sachs (as they are endogenously more likely to have the disease). However, we also don't want the company to base its prediction solely on the Ashkenazi feature (either positively or negatively). Instead, enforcing multicalibration would require that the learning algorithm investigate both the Ashkenazi and non-Ashkenazi population to predict accurately in each group (even if this means a higher false positive rate in the Ashkenazi population). In this case, relying on p^* seems to be well-aligned with promoting fairness.

Finally, we consider the interplay between multicalibration and "corrective discrimination" to be an important direction for further research. For example, one can imagine applying corrective measures, such as the transformation of [HPS16], to a multi-calibrated predictor.

1.4 Our Techniques

Here, we give a high-level technical overview of our results. Our techniques draw from the literature on computational learning theory, online optimization, differential privacy, and adaptive data analysis.

Learning a multicalibrated predictor In Section 3.2, we describe an algorithm for learning α-multicalibrated predictors as stated in Theorem 2. Our algorithm is an iterative procedure. In particular, we will maintain a candidate predictor x, and at each iteration, the algorithm corrects the candidate values of some subset that violates calibration until the candidate predictor is α-multicalibrated. Recall that calibration over a set S requires that on the subsets $S_v = \{i \in S : x_i = v\}$ (which we will refer to throughout as *categories*), the expected value of the true probabilities $\mathbb{E}_{i \sim S_v}[p_i^*]$ on this set is close to v. As such, the algorithm is easiest to describe in the statistical query model, where we query for noisy estimates of the true statistics on subsets of the population and update the predictor based on these estimates. In particular, given a statistical query oracle that guarantees tolerance $\omega = O(\alpha\gamma)$, the estimates will be accurate enough to guarantee α-calibration on sets S with $|S| \geq \gamma |X|$.

When we turn to adapting the algorithm to learn from random samples, the algorithm answers these statistical queries using the empirical estimates on some random sample from the population. Standard generalization arguments [KV94] show that if the set of queries we might ask is fixed in advance, then we could bound the sample complexity needed to answer these *non-adaptive* queries

as $\tilde{O}(\log|\mathcal{C}|/\omega^2)$. Note, however, that the categories S_v whose expectations we query are selected *adaptively* (i.e. with dependence on the results of prior queries). In particular, the definition of the categories S_v depends on the current values of the predictor x; thus, when we update x based on the result of a statistical query, the set of categories on which we might ask a statistical query changes. In this case, we cannot simply apply concentration inequalities and take a union bound to guarantee good generalization without resampling every time we update the predictor.

To avoid this blow-up in sample complexity, we appeal to recently-uncovered connections between differential privacy and adaptive data analysis developed in [DFH+15c, DFH+15c, BNS+16, DFH+15b]. In Section 3.3, we show how to answer the statistical queries in a way that guarantees the learning algorithm is differentially private. This, in turn, allows us to argue that the predictor our algorithm learns from a small sample will be multicalibrated, not just for the observed individuals but also the unseen individuals. In particular, using a privacy-based approach with an analysis tailored to the goal of calibration, we obtain sample complexity that depends on $1/\alpha^{11/2}\gamma^{3/2}$ as opposed to the naive approach which results in $1/\alpha^6\gamma^6$.

As stated earlier, Theorem 1 can be seen as a corollary of Theorem 2. Bounding the number of iterations needed by the algorithm to converge not only upper-bounds the algorithm's running time and sample complexity, but also implies that the circuit complexity of the learned predictor is not much larger than the complexity of evaluating membership for $S \in \mathcal{C}$. We explain this implication in Section 3.5.

The complexity of multicalibration In Section 3.4, we discuss the complexity of learning a multicalibrated predictor and draw connections to agnostic learning [Hau92, KSS94]. We show that the algorithm learns a multicalibrated predictor in a bounded number of iterations; however, without additional assumptions about p^* or $\mathcal{C}$, each iteration could take $\Omega(|\mathcal{C}|)$ time. In the cases where $|\mathcal{C}|$ is large, we might hope to improve the dependence on $|\mathcal{C}|$ to polylogarithmic or perhaps subpolynomial. If p^* can be learned directly, then we can eliminate the dependence on $|\mathcal{C}|$ and instead, only depend on the minimum γ such that for all $S \in \mathcal{C}$, $|S| \leq \gamma N$.

In the case where p^* is arbitrary, we show that improving the dependence on $|\mathcal{C}|$ is possible if $\mathcal{C}$ is structured in a certain way, drawing a connection to the literature on agnostic learning [Hau92, KSS94,KMV08,Fel10]. Recall, in our algorithm for learning multicalibrated predictors, we maintain a candidate predictor x, and iteratively search for some set $S \in \mathcal{C}$ on which x is not calibrated. To solve this search problem more quickly, we frame the search as weak agnostic learning over a concept class derived from $\mathcal{C}$ and over the hypothesis class of $\mathcal{H} = \{h : \mathcal{X} \to [-1, 1]\}$.

Specifically, consider the concept class defined by the collection of subsets $\mathcal{C}$, where for each $S \in \mathcal{C}$, we include the concept $c_S : \mathcal{X} \to \{-1, 1\}$ where $c_S(i) = 1$ if and only if $i \in S$. We show how to design a "labeling" $\ell : \mathcal{X} \to [-1, 1]$ for individuals such that if x violates the calibration constraint on any $S \in \mathcal{C}$, then the concept c_S correlates nontrivially with the labels over the distribution of individuals, i.e. $\langle c_S, \ell\rangle \geq \rho$ for some $\rho > 0$.

Thus, if x is not yet multicalibrated on $\mathcal{C}$, then we are promised that there is some concept c_S with nontrivial correlation with the labels; we observe that this promise is exactly the requirement for a weak agnostic learner, as defined in [KMV08, Fel10]. In particular, given labeled samples $(i, \ell(i))$ sampled according to $\mathcal{D}$, if there is a concept c_S with correlation at least ρ with ℓ, then the weak agnostic learner returns a hypothesis h that is τ correlated with ℓ for some $\tau < \rho$. The catch is

that this hypothesis may not be in our concept class $\mathcal{C}$, so we cannot directly "correct" any $S \in \mathcal{C}$. Nevertheless, the labeling on individuals ℓ is designed such that given the hypothesis h, we can still extract an update to x that will make global progress towards the goal of attaining calibration. As long as τ is nontrvially lower bounded, we can upper bound the number of calls we need to make to the weak learner. The details of our choice of labels and how we track progress are given in Section 4.

Additionally, we show that our reduction to weak agnostic learning is unavoidable. In particular, we show that if it is possible to learn a multicalibrated predictor with respect to $\mathcal{C}$, then it is possible to weak agnostic learn on $\mathcal{C}$ (if we view $\mathcal{C}$ as a concept class). Specifically, we will show how to implement a weak agnostic learner for $\mathcal{C}$, given an algorithm to learn an α-multicalibrated predictor x with respect to $\mathcal{C}$ (in fact, we only need the predictor to be multicalibrated on $\mathcal{C}' = \{S \in \mathcal{C} : |S| \geq \gamma |X|\}$). The key lemma for this reduction says that if there is some $c \in \mathcal{C}$ that is nontrivially correlated with the labels, then x is also nontrivially correlated with c. As there are many natural classes $\mathcal{C}$ for which agnostic learning is conjectured to be hard, this gives a strong negative result to the question of whether we can obtain speedups in the general case.

In combination, these results show that the complexity of learning a multicalibrated predictor with respect to a class $\mathcal{C}$ is equivalent to the complexity of weak agnostic learning $\mathcal{C}$. Section 4 contains the formal statements and proofs that imply this equivalence.

"Best-in-class" prediction In Section 5, we turn to understanding how requiring a predictor to be multicalibrated affects the accuracy of the predictor. We show – in contrast to many other notions of fairness – multicalibration does not limit the utility of a predictor. In particular, given any collection of predictors $\mathcal{H}$, and any collection of subsets $\mathcal{C}$, we design a procedure for obtaining a predictor x that is α-multicalibrated on $\mathcal{C}$ and achieves expected squared prediction error less than or equal to the the best predictor in $\mathcal{H}$ (plus a small additive error on α). Further, leveraging Theorem 1 and Theorem 2, we show that the predictor x can be learned from samples and implemented by a circuit of comparable size to the predictors $h \in \mathcal{H}$. To prove that multicalibration does not negatively impact the utility, we in fact, show a much stronger statement: if applying multicalibration to some $h \in \mathcal{H}$ changes the predictions of h significantly (i.e. if $\|x - h\|$ is large), then this change represents an improvement in accuracy (i.e. $\|x - p^*\| < \|h - p^*\|$). In this sense, requiring multicalibration is aligned with the goals of learning a high-utility predictor.

Organization of the paper In Section 2, we provide a description of our model and the formal definitions related to multicalibration. In Section 3, we describe and analyze our algorithm for learning multicalibrated predictors. In Section 4, we investigate the complexity of obtaining multicalibration, showing a tight connection to the complexity of weak agnostic learning. Finally, in Section 5, we demonstrate how multicalibration achieves "best-in-class" prediction.

2 Preliminaries

Predictors Let $\mathcal{X}$ denote the universe of N individuals over which we wish to make predictions about some outcome $o \in \{0,1\}^N$. We assume that outcomes are the result of some underlying

random process, where for each i, o_i is sampled independently[3] as a Bernoulli random variable with success probability p^*_i. We aim to predict the underlying parameters of the process, rather than the realized outcome of the process. A *predictor* $x : \mathcal{X} \to [0, 1]$ of outcome o is some mapping from individuals $i \in \mathcal{X}$ to $[0, 1]$, where x_i is the prediction of p^*_i. We refer to p^* as the baseline predictor. When it is notationally convenient, we will sometimes treat predictors as vectors $x \in [0, 1]^N$ rather than as functions $x : \mathcal{X} \to [0, 1]$, where we assume a bijection from $\mathcal{X}$ to $[N]$.

Remark. *Often in learning theory, we think of learning functions $h : \mathcal{F} \to [0, 1]$ over the space of possible features $\mathcal{F}$. We find it preferable to reason about predictions about* individuals*; nevertheless, in our model, we can think of x_i as given by the composition of two separate functions, where $x_i = x'(\phi(i))$ for $x' : \mathcal{F} \to [0, 1]$ and $\phi : \mathcal{X} \to \mathcal{F}$. As ϕ will be fixed and assumed to be some simple function (say, mapping individuals to their $\langle$ age, height, ZIP code, $\ldots\rangle$) we drop the explicit reference to $\mathcal{F}$ and ϕ.*

Sampling from $\mathcal{X}$ Throughout, we will assume that our learning algorithms have the ability to efficiently obtain randomly sampled individuals from $\mathcal{X}$ (and by rejection sampling, large subsets of $\mathcal{X}$). Specifically, we will use $i \sim S$ to denote sampling i uniformly at random from $S \subseteq \mathcal{X}$.

Remark. *Our focus on the uniform distribution is mostly syntactic; as per our distinction between the identities of individuals $i \in \mathcal{X}$ and their features $\phi(i)$, the uniform distribution over individuals gives rise to a rich class of distributions over the features of individuals.*

We will further assume that sampling $i \sim \mathcal{X}$ is inexpensive compared to obtaining a corresponding outcome $o_i \in \{0, 1\}$ for $i \sim \mathcal{X}$ that is Bernoulli distributed with parameter p^*_i. As such, when measuring the sample complexity, we will only count the latter type of labeled samples. The learner has a good sense of the distribution over features, but not of the outcomes that arise from these features.

When measuring the accuracy of a predictor x, we will use the squared prediction error $\|p^* - x\|^2$, as this measure of divergence penalizes large individual deviations. The ℓ_2^2 distance will also be useful for measuring the similarity of predictors. While we elect to use ℓ_2^2, much of our analysis could be performed using any Bregman divergence; in particular, if we elected to work over arbitrary multinomial distributions over individuals in $\mathcal{X}$, we could measure accuracy in terms of the squared Mahalanobis distance from the optimal predictor, i.e. $(p^* - x)^T D(p^* - x)$, where D is the diagonal matrix in which D_{ii} is the probability of sampling individual $i \in \mathcal{X}$.

2.1 Calibration

Next, we give formal definitions of the criteria we use to measure algorithmic discrimination. The first notion captures the idea that, on average, we would like an algorithm's predictions to be unbiased.

[3] In many cases, complete independence may not hold; individuals' outcomes may be correlated in nontrivial ways. The only place we will use independence is to argue that the reliable statistics can be estimated from the observable data; our arguments can be applied to any model for which one can prove the appropriate tail inequalities. For further consideration, continue to our discussion on "observable" calibration after Claim 2.3.

Definition 2.1 (Accurate in expectation)**.** *For any $\alpha > 0$ and $S \subseteq \mathcal{X}$, a predictor x is* α-accurate in expectation (α-AE) *with respect to S if*

$$\left| \mathop{\mathbb{E}}_{i \sim S}[x_i - p_i^*] \right| \leq \alpha.$$

That is, the predictions, averaged over the set S, are accurate up to some additive α slack. This basic condition is necessary to ensure unbiased predictions, but it is too weak to guarantee discrimination hasn't occurred. In particular, suppose the underlying probabilities are such that $p_i^* = 1/2$ for all $i \in S$. A predictor that predicts 1 on half of the individuals in S and 0 on the other half is accurate in expectation, but it is arguably discriminatory; there is no difference between the individuals in S, but the predictor has artificially created two categories within this population. This example motivates the definition of calibration. Calibration mitigates this form of discrimination by considering the expected values over categories $S_v = \{i : x_i = v\}$ defined by the predictor x. Specifically, α-calibration with respect to S requires that for all but an α-fraction of a set S, the average of the true probabilities of the individuals receiving prediction v is α-close to v.

Definition 2.2 (Calibration)**.** *For any $v \in [0,1]$, $S \subseteq \mathcal{X}$, and predictor x, let $S_v = \{i : x_i = v\}$. For $\alpha \in [0,1]$, x is* α-calibrated *with respect to S if there exists some $S' \subseteq S$ with $|S'| \geq (1-\alpha)|S|$ such that for all $v \in [0,1]$,*

$$\left| \mathop{\mathbb{E}}_{i \sim S_v \cap S'}[x_i - p_i^*] \right| \leq \alpha.$$

Note that α-calibration with respect to S implies 2α-AE with respect to S. To see this, observe that calibration implies that on a $(1-\alpha)$-fraction of S, the average of the values are α-close to the expectation on this fraction; even if the other α-fraction is arbitrary, it can only introduce another additive α error.

Departures from prior definitions This notion of calibration departs from prior definitions in a few ways. Earlier definitions required exact calibration with respect to $\mathcal{X}$; we find it meaningful to consider approximate calibration. Introducing approximation into the definition of calibration has practical motivation. In particular, we don't expect to know the exact probabilities, nor can we observe the entire population. Given a desired accuracy and access to samples from the population, we can quantify how many samples are needed to guarantee calibration with good probability.

Note that in our definition of α-calibration, we require *relative* additive error that scales with the size of S_v. That is, for some S and category $S_v = \{i : x_i = v\} \cap S$, the magnitude of the sum of errors on S_v, $\left|\sum_{i \in S_v}(v - p_i^*)\right|$, scales with $|S_v|$, not N. This relative approximation model prevents certain "attacks" against approximate calibration. Specifically, one natural way a predictor might attempt to sidestep the anti-discrimination properties of approximate calibration would be to support many distinct values of $v \in [0,1]$, each with a very small number of individuals. If our notion of approximation allowed for *absolute* errors (i.e. errors whose sum over S_v scale with N instead of $|S_v|$), then this attack would be viable, leading to essentially no guarantees

Another subtle distinction is that we evaluate calibration with respect to the underlying probabilities $p^* \in [0,1]^N$, as opposed to the realized outcomes $o \in \{0,1\}^N$. We refer to the earlier notion as

observable calibration. Formally, a predictor is *observably α-calibrated* with respect to $S \subseteq \mathcal{X}$ and outcome $o \in \{0,1\}^N$ if for all $v \in [0,1]$,

$$\left| \mathop{\mathbb{E}}_{i \sim S_v \cap S'} [x_i - o_i] \right| \leq \alpha,$$

where S_v and S' are as in Definition 2.2. In our terminology, earlier references to calibration require that a predictor be *observably* 0-calibrated with respect to each protected group S. Our introduction of a non-zero α in these definitions allows us to relate our notion of calibration and the previous notion of observable calibration.

Claim 2.3. *Let $S \subseteq \mathcal{X}$, $\delta > 0$, and $\alpha > \sqrt{\frac{\log(1/\delta)}{2|S|}}$. Suppose the outcome $o \in \{0,1\}^N$ is drawn according to the true probabilities p^*. Then, with probability at least $1 - \delta$ over the draw of o, any predictor that is α-calibrated with respect to S will be* observably 2α*-calibrated with respect to S.*

Claim 2.3 follows directly from an application of Hoeffding's inequality over the random outcome o. The claim implies that for sufficiently large α, to guarantee *observable* 2α-calibration with high probability, it suffices to consider our notion of α-calibration.

Independence and observability We note that to prove Claim 2.3, we use the independence of o_i's in our application of Hoeffding's inequality. Throughout, the only times we invoke the independence of the realization of the o_i's are to prove that the empirical statistics over a sufficiently large sample of observations will be concentrated around the corresponding statistics of the true parameters. Thus, the independence assumption isn't strictly necessary; our results should hold for any model where the outcomes admit similar tail inequalities. We note, however, that if strong dependencies exist in the realization of the outcomes, our techniques will still achieve *observable* calibration from samples. Recall that in observable calibration, we compare the value v output by the predictor on a category S_v to the empirical average of outcomes over the category $\sum_{i \in S_v} o_i$. If we only need to guarantee closeness with respect to these observable outcomes, we do not need to ensure that sums over outcomes will be concentrated around their expectation (i.e. sums of the underlying true probabilities).

2.2 Learning model

In Section 3, we present algorithms to learn predictors satisfying accuracy-in-expectation and calibration. The algorithms can be viewed as statistical query algorithms [Kea98]. Specifically, the algorithms only require access to approximate statistical queries of the following form.

Definition 2.4 (Statistical Query [Kea98])**.** *For a subset of the universe $S \subseteq \mathcal{X}$, let $p^*_S = \sum_{i \in S} p^*_i$. For $\tau \in [0,1]$, a* statistical query with tolerance τ *returns some $\tilde{p}(S)$ satisfying*

$$p^*_S - \tau N \leq \tilde{p}(S) \leq p^*_S + \tau N.$$

Note that this query model guarantees *absolute* additive error τN. As discussed above, our notion of calibration with respect to S asks for *relative* additive error; however, if we know a lower bound

on $|S| \geq \gamma N$, then, asking a statistical query with tolerance $\tau = \alpha\gamma$ will guarantee relative error α on S.

In addition to giving algorithms that work in this statistical learning framework, we also address the question of learning a calibrated predictor from a set of samples of outcomes. Formally, we define the access to sampled outcomes as follows.

Definition 2.5 (Random sample)**.** *For $p^* \in [0,1]^N$, a* random sample $s(p^*)$ *returns an individual-outcome pair $(i, o_i) \in \mathcal{X} \times \{0,1\}$, where $i \sim \mathcal{X}$ is drawn uniformly at random and o_i is sampled according to the Bernoulli distribution with parameter p_i^*.*

We say an algorithm learns a predictor from samples if its only access to the true parameters p^* is through random samples of this form. It's easy to see that we can always implement a statistical query algorithm with access to enough random samples – for every query, we could sample a fresh set of outcomes to estimate the statistic accurately. Our goal will be to avoid resampling in order to prevent a blow-up in the sample complexity.

2.3 Multicalibration

We introduce the notion of *multicalibration*, which requires calibration to hold simultaneously on subsets of the population. We will show that multicalibration not only guarantees fairness across protected populations, but also helps us uncover more accurate predictions. To motivate multicalibration further, consider the following toy example: suppose p^* is such that there is some population S (possibly, a traditionally protected group) and a subpopulation $S' \subseteq S$ with $|S'| = |S|/2$, where for every $i \in S'$, $p_i^* = 1$, and for every $i \in S \setminus S'$, $p_i^* = 0$. The predictor x that predicts $x_i = 1/2$ for all $i \in S$ is calibrated on the population S, but clearly is suboptimal. Further, if S' was identifiable in advance, then this predictor is arguably discriminatory – there are two clearly identifiable groups within S, but we are treating them the same way. If, however, we insist on calibration with respect to S' in addition to S, then the predictor will be required to output accurate predictions for each group. Earlier approaches to using calibration to achieve fairness, as introduced in [KMR16], would prevent this form of discrimination for subsets S that are identified as a protected group (defined, for example, by race), but not for subpopulations of these groups – even if the subpopulations could be easily distinguished as outstanding.

For a collection of subsets $\mathcal{C}$, we say that a predictor is α-multicalibrated on $\mathcal{C}$ if it is α-calibrated simultaneously on all $S \in \mathcal{C}$.

Definition 2.6 (α-multicalibration)**.** *Let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be a collection of subsets of $\mathcal{X}$ and $\alpha \in [0,1]$. A predictor x is α-multicalibrated on $\mathcal{C}$ if for all $S \in \mathcal{C}$, x is α-calibrated with respect to S.*

We also define the corresponding definition for the weaker notion of α-AE.

Definition 2.7 (α-multi-AE)**.** *Let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be a collection of subsets of $\mathcal{X}$ and $\alpha \in [0,1]$. A predictor x is α-multi-AE on $\mathcal{C}$ if for all $S \in \mathcal{C}$, x is α-AE with respect to S.*

Discretization Even though α-calibration is a meaningful definition if we allow for arbitrary predictions $x_i \in [0,1]$, when designing algorithms to learn calibrated predictors, it will be useful to maintain some discretization on the values $v \in [0,1]$. Formally, we will use the following definition.

Definition 2.8 (λ-discretization)**.** *Let $\lambda > 0$. The* λ-discretization *of $[0,1]$, denoted by $\Lambda[0,1] = \left\{\frac{\lambda}{2}, \frac{3\lambda}{2}, \ldots, 1 - \frac{\lambda}{2}\right\}$, is the set of $1/\lambda$ evenly spaced real values over $[0,1]$. For $v \in \Lambda[0,1]$, let*

$$\lambda(v) = [v - \lambda/2, v + \lambda/2)$$

be the λ-interval centered around v (except for the final interval, which will be $[1-\lambda, 1]$).

At times, it will be convenient to work with a more technical variant of multicalibration, which implies α-multicalibration. In particular, this definition will allow us to work with an explicit discretization of the values $v \in [0,1]$. Throughout, for a predictor x, we refer to the "categories" $S_v(x) = \{i : x_i \in \lambda(v)\} \cap S$ for all $S \in \mathcal{C}$ and $v \in \Lambda[0,1]$.

Definition 2.9 ((α,λ)-multicalibration)**.** *Let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be a collection of subsets of $\mathcal{X}$. For any $\alpha, \lambda > 0$, a predictor x is* (α,λ)-calibrated *on $\mathcal{C}$ if for all $S \in \mathcal{C}$, $v \in \Lambda[0,1]$, and all categories $S_v(x)$ such that $|S_v(x)| \geq \alpha\lambda |S|$, we have*

$$\left| \sum_{i \in S_v(x)} x_i - p_i^* \right| \leq \alpha |S_v(x)| .$$

We claim that if learn a predictor that satisfies (α,λ)-multicalibration, we can easily transform this predictor into one that satisfies our earlier notion of α-multicalibration. In particular, let x^λ be the λ-discretization of a predictor x if for all $i \in S_v(x)$, $x_i^\lambda = \mathbb{E}_{i \sim S_v(x)}[x_i]$.

Claim 2.10. *For $\alpha, \lambda > 0$, suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is a collection of subsets of $\mathcal{X}$. If x is (α,λ)-multicalibrated on $\mathcal{C}$, then x^λ is $(\alpha+\lambda)$-multicalibrated on $\mathcal{C}$.*

Proof. Consider the categories $S_v(x)$ where $|S_v(x)| < \alpha\lambda |S|$. By the λ-discretization, there are at most $1/\lambda$ such categories, so the cardinality of their union is at most $(1/\lambda)\alpha\lambda |S| = \alpha |S|$. Thus, for each $S \in \mathcal{C}$, there is a subset $S' \subseteq S$ with $|S'| \geq (1-\alpha)|S|$ where for all $v \in \Lambda[0,1]$,

$$\left| \mathop{\mathbb{E}}_{i \sim S_v(x) \cap S'} [x_i - p_i^*] \right| \leq \alpha.$$

Further, λ-discretization will "move" the values of x_i by at most λ, so overall, x^λ will be $(\alpha+\lambda)$-calibrated. □

Typically, we will imagine $\lambda = \Theta(\alpha)$, but our results hold for any $\lambda \in (0,1]$. Choosing a smaller λ will allow the predictor to be more expressive, but will also increase the running time and sample complexity. Choosing a larger λ leads to a decay in the calibration guarantees.

Representing subsets of individuals When representing collections of subsets, we will assume that the subsets are represented implicitly. In particular, we will assume that $S \in \mathcal{C}$ is given as a circuit $c_S : \mathcal{X} \to \{0,1\}$, where $S = c_S^{-1}(1)$; that is, for $i \in \mathcal{X}$, $c_S(i) = 1$ if and only if $i \in S$. Using this implicit representation serves two purposes. First, in many cases, we may want to calibrate on a collection of subsets over a large universe; in these cases, assuming an explicit representation of each set is unreasonable. Second, associating a set S with a circuit that computes membership

in S allows us to quantify the complexity of the sets in $\mathcal{C}$. In particular, it seems natural to apply multicalibration to guarantee calibration with respect to a collection of *efficiently-identifiable* subsets (say, subsets defined by conjunctions of four attributes, or any other simple circuit class). It seems comparatively unreasonable to require calibration on, say, a random subsets, each of which would require $\Omega(|\mathcal{X}|)$ bits to describe.

3 Learning α-multicalibrated predictors

In this section, we prove Theorem 2; that is, we provide an algorithm for efficiently learning α-multicalibrated predictors. The algorithms we describe are iterative and fit into the online optimization framework [SS12,Haz16] as well as the statistical query framework [Kea98]. In Section 3.1, we give an algorithm that solves the simpler task of learning an α-multi-AE predictor, as a warm-up to introduce the main ideas. In Section 3.2, we describe the algorithm for learning α-multicalibrated predictors in full. Then, in Section 3.3, we will give a nontrivial implementation of the statistical query oracle that will imply nontrivial upper bounds on the running time and sample complexity of the learning algorithm. This implementation borrows ideas from the literature on differentially private query release and optimization [HR10, Ull15, HRRU14]. We conclude in Section 3.5 with the observation that our algorithm also has implications for the circuit complexity of calibrated predictors: for any collection of sets $\mathcal{C}$, there is an α-calibrated predictor whose circuit complexity is a small factor larger than the circuit complexity required to describe the sets in $\mathcal{C}$. This establishes Theorem 1.

3.1 α-multi-AE predictors

We begin our discussion with a simpler statistical query algorithm for learning an α-multi-AE predictor. This algorithm serves as a warm-up for the subsequent algorithm for learning an α-multicalibrated predictor.

Algorithm 3.1 – Learning an α-multi-AE predictor on $\mathcal{C}$

Let $\alpha, \gamma > 0$ and let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be such that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$.
For $S \subseteq \mathcal{X}$, let $\tilde{p}(S)$ be the output of a statistical query with tolerance $\tau < \alpha\gamma/4$.

- Initialize:
  - Let $x = (1/2, \ldots, 1/2) \in [0,1]^N$
- Repeat:
  - For each $S \in \mathcal{C}$:
    - Let $\Delta_S = \tilde{p}(S) - \sum_{i \in S} x_i$
    - If $|\Delta_S| > \alpha |S| - \tau N$:
      update $x_i \leftarrow x_i + \frac{\Delta_S}{|S|}$ for all $i \in S$ (projecting x_i onto $[0,1]$ if necessary)
  - If no $S \in \mathcal{C}$ updated: exit and output x

Algorithm 3.1 describes an iterative statistical query procedure for learning a α-multi-AE predictor on $\mathcal{C}$. Note that the problem of finding an α-multi-AE predictor for some collection of sets $\mathcal{C}$ can be written as a linear program; the algorithm presented can be viewed as an instance of projected subgradient descent (see [Haz16]). The algorithm iteratively updates a predictor x until it cannot find a set $S \in \mathcal{C}$ where the current estimate deviates significantly from the value reported by the statistical query. We claim that if no set violates this condition, then x is α-multi-AE on $\mathcal{C}$.

Claim 3.1. *If Algorithm 3.1 outputs a predictor x, then x is α-multi-AE on $\mathcal{C}$.*

Proof. Let $p_S = \sum_{i \in S} p_i^*$ and $x_S = \sum_{i \in S} x_i$. By the assumed tolerance of the statistical queries $\tilde{p}(S)$, we know that the queries are close to the p_S. Specifically, we know $|\tilde{p}(S) - p_S^*| \leq \tau N$ for some $\tau < \alpha\gamma$. By the termination condition and the triangle inequality, for all $S \in \mathcal{C}$ we get the estimate $|p_S^* - x_S| \leq |\tilde{p}(S) - x_S| + \tau N \leq \alpha |S|$; thus x is α-AE on $\mathcal{C}$. □

Thus, to show the correctness of the algorithm, it remains to show that the algorithm will, in fact, terminate; we show the algorithm can make at most $O(1/\alpha^2\gamma)$ updates.

Lemma 3.2. *Suppose $\alpha, \gamma > 0$ and $\mathcal{C} \subseteq 2^{\mathcal{X}}$ such that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$. Let $\tau = \alpha\gamma/4$. Then Algorithm 3.1 makes $O(1/\alpha^2\gamma)$ updates to x before terminating.*

Proof. We use a potential argument, tracking the progress the algorithm makes on each update in terms of the ℓ_2^2 distance between our learned predictor x and the true predictions p^*. Let x' be the predictor after updating x on set S and let $\pi : \mathbb{R} \to [0, 1]$ denote projection onto $[0, 1]$. We use the fact that the ℓ_2^2 can only decrease under this projection. For notational convenience, let $\delta_S = \dfrac{\Delta_S}{|S|} = \dfrac{1}{|S|}(\tilde{p}(S) - \sum_{i \in S} x_i)$. We have

$$\begin{aligned}
\|p^* - x\|^2 - \|p^* - x'\|^2 &= \sum_{i \in S} (p_i^* - x_i)^2 - \sum_{i \in S} (p_i^* - \pi(x_i + \delta_S))^2 \\
&\geq \sum_{i \in S} ((p_i^* - x_i)^2 - (p_i^* - (x_i + \delta_S))^2) \\
&= \sum_{i \in S} (2(p_i^* - x_i)\delta_S - \delta_S^2) \\
&= \left(2\delta_S \sum_{i \in S} (p_i^* - x_i) \right) - \delta_S^2 |S| \\
&\geq 2\delta_S (\delta_S |S| - \operatorname{sgn}(\delta_S)\tau N) - \delta_S^2 |S| \\
&\geq \delta_S^2 |S| - 2 |\delta_S| \tau N.
\end{aligned}$$

By setting $\tau = \alpha\gamma/4$ and by the bound $|\Delta_S| \geq \alpha |S| - \tau N \geq 3\alpha |S| /4$, the final quantity is at least $\Omega(\alpha^2 |S|)$. We also have

$$\begin{aligned}
\delta_S^2 |S| - 2 |\delta_S| \tau N &\geq \left(\frac{3\alpha}{4} \right)^2 |S| - 2 \left(\frac{3\alpha}{4} \right) \left(\frac{\alpha\gamma}{4} \right) N \\
&= \frac{3\alpha^2}{16} |S| .
\end{aligned}$$

The ℓ_2^2 distance between p^* and any other predictor (in particular, our initial choice for x) is upper-bounded by N. Thus, given that all $S \in \mathcal{C}$ have $|S| \geq \gamma N$, we make at least $\Omega(\alpha^2 \gamma N)$ progress in potential at each update, so the lemma follows. □

In combination, these statements show the correctness of Algorithm 3.1 and imply an upper bound on the number of statistical queries necessary.

Theorem 3.3. *For $\alpha, \gamma > 0$ and for any $\mathcal{C} \subseteq 2^{\mathcal{X}}$ satisfying $|S| \geq \gamma N$ for all $S \in \mathcal{C}$, there is a statistical query algorithm with tolerance $\tau = \alpha\gamma/4$ that learns a α-multi-AE predictor on $\mathcal{C}$ in $O(|\mathcal{C}|/\alpha^2\gamma)$ queries.*

Recall that, trivially, we could implement this statistical query algorithm from random samples by resampling for every query; however, in this case, we can easily improve the sample complexity exponentially over the trivial solution. Specifically, the queries we make are *non-adaptive* because, up front, we know a fixed collection of subsets whose expectation we might query. To guarantee accurate expectations on this fixed collection, we only need enough samples to guarantee that the sample is inaccurate on a fixed subset with very small probability, and then union bound over all $|\mathcal{C}|$ subsets. Appealing to a standard generalization argument [KV94], we can show the following theorem.

Corollary 3.4. *Suppose $\alpha, \gamma, \xi > 0$ and $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is such that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$. Then there is an algorithm that learns an α-multi-AE predictor on $\mathcal{C}$ with probability at least $1 - \xi$ from $n = O\left(\frac{\log(|\mathcal{C}|/\xi)}{\alpha^2\gamma}\right)$ samples.*

Note that the γ dependence in the sample complexity is only $1/\gamma$. Naively, applying the guarantees of the statistical query oracle, we would obtain a $1/\gamma^2$ dependence. To achieve this bound, we note that because calibration requires relative error, we can be more judicious with our use of samples. We will exploit this observation subsequently to obtain improvements to the sample complexity for learning α-multicalibrated predictors.

Proof. To obtain the claimed sample complexity bound, we observe that in Algorithm 3.1, we only use the statistical query oracle to guarantee bounds on the relative error of each query – not absolute error. In particular, for $S \subseteq \mathcal{X}$ with $|S| \geq \gamma N$, let $\bar{p}_S = \frac{1}{|S|}\sum_{i \in S} p_i^*$. To run Algorithm 3.1, we need only implement an oracle $\hat{p}(S)$ satisfying

$$\bar{p}_S - \tau \leq \hat{p}(S) \leq \bar{p}_S + \tau.$$

By Chernoff bounds, for a fixed set S of cardinality at least γN, if we take $O(t/\gamma\alpha^2)$ independent samples, the probability that the estimate of $\bar{p}_S$ differs by more than $\alpha|S|$ is at most $e^{-\Omega(t)}$. Taking $t = c(\log|\mathcal{C}| + \log(1/\xi))$ for an appropriate constant c, a union bound implies the probability that the estimate of every set $S \in \mathcal{C}$ is α-accurate is at least $1 - \xi$. □

3.2 α-Multicalibrated predictors

Next, we present the full algorithm for learning α-multicalibrated predictors on $\mathcal{C}$. The algorithm is based on Algorithm 3.1 but differs in a few key ways. First, instead of updating the predictions

on entire sets $S \in \mathcal{C}$ whose overall expectation is wrong, we update the predictions on uncalibrated categories $S_v = S \cap \{i : x_i = v\}$. This is a simple change to the algorithm in the statistical query model; however, when we wish to implement this statistical query oracle from a finite sample, we need to be more careful.

In particular, the categories of the predictor we learn are not fixed *a priori*, so our queries will be selected *adaptively* based on the results of earlier statistical queries. Stated another way, we cannot simply union bound against the collection of sets on which we wish to be calibrated. The most naive approach to bounding the sampling complexity would be as follows: at each iteration take a fresh sample large enough to guarantee ωN absolute error, for $\omega = \alpha\gamma$. Following our analysis below for this naive strategy, it's easy to see that this approach would result in $\Omega(1/\alpha^4\gamma^4)$ iterations and sample complexity $n \geq \Omega(1/\alpha^6\gamma^6)$. We will improve on this approach by polynomial factors achieving a bound of at most $O(1/\alpha^4\gamma)$ iterations with $O(1/\alpha^{5/2}\gamma^{3/2})$ samples.

To achieve these improvements, we combine two ideas. As before, we will leverage the observation that calibration only requires relative error (as in Corollary 3.4), and thus, in principle should require fewer samples. Additionally, to avoid naively resampling but still guarantee good generalization from a small sample, we interact with the sample through a mechanism which we call a *guess-and-check* statistical query (similar in spirit to mechanisms proposed in [HR10, BH15, GRU12]). We show how to implement this mechanism in a manner that guarantees generalization on the unseen data even after asking many adaptively chosen statistical queries. We defer our discussion of privacy to Section 3.3.

Details of the algorithm Next, we give an iterative procedure to learn a (α, λ)-multicalibrated predictor on $\mathcal{C}$ as described in Algorithm 3.2. The procedure is similar to Algorithm 3.1, but deliberately interacts with its statistical queries through a so-called guess-and-check oracle. In particular, each time the algorithm needs to know the value of a statistical query on a set S, rather than asking the query directly, we require that the algorithm submit its current guess $x_S = \frac{1}{|S|}\sum_{i \in S} x_i$ to the oracle, as well as an acceptable "window" $\omega \in [0, 1]$. Intuitively, if the algorithm's guess is far from the window centered around the true expectation, then the oracle will respond with the answer to a statistical query with tolerance ω. If, however, the guess is sufficiently close to the true value, then the oracle responds with ✓ to indicate that the current guess is close to the expectation, without revealing another answer.

Definition 3.5 (Guess-and-check oracle)**.** *Let $\tilde{q} : 2^{\mathcal{X}} \times [0, 1] \times [0, 1] \to [0, 1] \cup \{\checkmark\}$. $\tilde{q}$ is a* guess-and-check oracle *with window ω_0 if for $S \subseteq \mathcal{X}$ with $p_S = \sum_{i \in S} p_i^*$, $v \in [0, 1]$, and any $\omega \geq \omega_0$, the response to $\tilde{q}(S, v, \omega)$ satisfies the following conditions:*

- *if $\left|p_S - |S|\, v\right| < 2\omega N$, then $\tilde{q}(S, v, \omega) = \checkmark$*
- *if $\left|p_S - |S|\, v\right| > 4\omega N$, then $\tilde{q}(S, v, \omega) \in [0, 1]$*
- *if $\tilde{q}(S, v) \neq \checkmark$, then*

$$p_S - \omega N \leq \tilde{q}(S, v, \omega)\,|S| \leq p_S + \omega N.$$

Note that if the guess is such that $\left|p_S - |S|\, v\right| \in [2\omega N, 4\omega N]$, the the oracle may respond with some ω-accurate $r \in [0, 1]$ or with ✓. Of course, if we have a lower bound ω_0 on the window over

a sequence of guess-and-check queries, we can implement the queries given access to a statistical query oracle with tolerance $\tau \leq \omega_0$; it is also clear that a statistical oracle with tolerance τ can be implemented with access to a guess-and-check oracle with window $\tau/4$. The advantage of using this guess-and-check framework is that it can be implemented in a differentially private manner. This will in turn allow us to give an algorithm for learning α-multicalibrated predictors from a small number of samples that generalizes well.

Algorithm 3.2 – Learning a (α, λ)-calibrated predictor on $\mathcal{C}$

Let $\alpha, \lambda > 0$ and let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be such that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$.
For $S \subseteq \mathcal{X}$ and $v \in [0,1]$, let $\tilde{q}(\cdot,\cdot,\cdot)$ be a guess-and-check oracle.

- Initialize:
  - Let $x = (1/2, \ldots, 1/2) \in [0,1]^N$
- Repeat:
  - For each $S \in \mathcal{C}$, $v \in \Lambda[0,1]$, for each $S_v = S \cap \{i : x_i = \lambda(v)\}$ such that $|S_v| = \beta N \geq \alpha\lambda |S|$
    - Let $\bar{v} = \frac{1}{|S_v|} \sum_{i \in S_v} x_i$
    - Let $r = \tilde{q}(S_v, \bar{v}, \alpha\beta/4)$
    - If $r \neq \checkmark$:
      update $x_i \leftarrow x_i + (r - \bar{v})$ for all $i \in S_v$ (projecting x_i onto $[0,1]$ if necessary)
  - If no S_v updated, exit
- For $v \in \Lambda[0,1]$:
  - Let $\bar{v} = \sum_{i \in \lambda(v)} x_i$
  - For $i \in \lambda(v)$: $x_i \leftarrow \bar{v}$
- Output x

Algorithm 3.2 runs through each possible category S_v and if S_v is large enough, queries the oracle. The algorithm continues searching for uncalibrated categoires until x's guesses on all sufficiently large categories receive $\checkmark$. By the definition of the guess-and-check oracle, if for some category S_v where $|S_v| = \beta N$ the query returns $\checkmark$, then $\bar{v}$ is at most $4 \cdot (\alpha\beta N/4) = \alpha |S_v|$ far from the true value $\frac{1}{S_v} \sum_{i \in S_v} p_i^*$. Thus, by the stopping condition of the loop, the predictor where all $i \in \lambda(v)$ receive $x_i = \bar{v}$ will be α-calibrated on every large category. Finally, the algorithm updates x to be λ-discretized, so by Claim 2.10, x will be $(\alpha + \lambda)$-calibrated. Further, the number of updates necessary to terminate is bounded.

Lemma 3.6. *Suppose $\alpha, \lambda > 0$ and $\mathcal{C} \subseteq 2^{\mathcal{X}}$ where for all $S \in \mathcal{C}$, $|S| \geq \gamma N$. Algorithm 3.2 returns x after receiving at most $O(1/\alpha^3\lambda\gamma)$ guess-and-check responses where $r \in [0,1]$ and at most $O(|C|/\alpha^4\lambda\gamma)$ responses $r = \checkmark$.*

Proof. For some non-$\checkmark$ response on $S_v = \{i : x_i \in \lambda(v)\} \cap S$, by the properties of the guess-and-check oracle, we can lower bound the update step size. Recall, we only query on sets wherer

$|S_v| = \beta N \geq \alpha |S|$ with a window of $\omega = \alpha\beta/4$.

$$\begin{aligned}\left|\sum_{i \in S_v} p_i^* - x_i\right| &= \left|\sum_{i \in S_v} p_i^* - \bar{v}\right| \\ &\geq 2\left(\alpha\beta/4\right) N \\ &= \alpha \left|S_v\right| /2.\end{aligned}$$

Letting $\delta_v = r - \bar{v}$. We can measure progress in the same way as in Lemma 3.2.

$$\begin{aligned}\left\|p^* - x\right\|^2 - \left\|p^* - x'\right\|^2 &= \sum_{i \in S_v} (p_i^* - x_i)^2 - \sum_{i \in S_v} (p_i^* - \pi(x_i + \delta_v))^2 \\ &\geq \sum_{i \in S_v} ((p_i^* - x_i)^2 - (p_i^* - (x_i + \delta_v))^2) \\ &= \sum_{i \in S_v} (2(p_i^* - x_i)\delta_v - \delta_v^2) \\ &= \left(2\delta_v \sum_{i \in S_v} (p_i^* - x_i)\right) - \delta_v^2 \left|S_v\right|\end{aligned}$$

Let $\nu = \frac{1}{|S_v|} \sum_{i \in S_v} (p_i^* - x_i)$. By the properties of the guess-and-check oracle, we can rewrite δ_v as $\nu - \eta$ for some $\eta \in [-\omega/\beta, \omega/\beta]$. This gives us a lower bound on the progress as follows.

$$\left(2(\nu - \eta)\nu - (\nu - \eta)^2\right) |S_v| = \left(\nu^2 + \nu\eta - (\eta)^2\right) |S_v|$$

This concave function in η is minimized at an extreme value for η (depending on the sign of ν). Noting that $|\nu| \geq \alpha/2$ and $|\eta| \leq \omega/\beta = \alpha/4$, we can lower bound our progress by $(\alpha/4)^2 |S_v| = \alpha^2\beta N/16 = \alpha^3\lambda\gamma N/16$. As $\|p^*\|^2 \leq N$, we make at most $O(1/\alpha^3\lambda\gamma)$ updates upper bounding the number of non-✓ responses. By working with a λ-discretization, there are at most $|C|/\lambda$ categories to consider in every phase, so we receive at most $O(|C|/\alpha^3\lambda^2\gamma)$ ✓ responses. □

Thus, we conclude the following theorem.

Theorem 3.7. *For $\alpha, \lambda > 0$ and $\mathcal{C} \subseteq 2^{\mathcal{X}}$ where for all $S \in \mathcal{C}$, $|S| \geq \gamma N$, there is a statistical query algorithm that learns a (α, λ)-multicalibrated predictor with respect to $\mathcal{C}$ in $O(|\mathcal{C}|/\alpha^3\lambda^2\gamma)$ queries.*

Again, note that our output is, in fact, $(\alpha + \lambda)$-multicalibrated, so taking $\lambda = \alpha$, we obtain a (2α)-multicalibrated predictor in $O(|\mathcal{C}|/\alpha^5\gamma)$ queries.

3.3 Answering guess-and-check queries from a random sample

Next, we argue that we can implement a guess-and-check oracle from a set of random samples in a manner that guarantees good generalization. This, in turn, allows us to translate our statistical query algorithm for learning an (α, λ)-multicalibrated predictor with respect to $\mathcal{C}$ into an algorithm that learns from samples. As mentioned in the beginning of Section 3, naively, we could resample for every update the algorithm makes to the predictor. Suppose that $\mathcal{C}$ is such that for all $S \in$

$\mathcal{C}$, $|S| \geq \gamma N$; let $\beta = \alpha\lambda\gamma$. Using our tighter analysis of Algorithm 3.2, we could take $n = \tilde{O}(\log(|\mathcal{C}|)/\alpha^2\beta)$ samples per update to guarantee generalization, resulting in an overall sample complexity of $\tilde{O}(\log(|\mathcal{C}|)/\alpha^4\beta^2)$. We show how to improve upon this approach further. In particular, we argue that there is a differentially private algorithm that can answer the entire sequence of guess-and-check queries accurately. Appealing to known connections between differential privacy and adaptive data analysis [DFH+15c, DFH+15a, BNS+16, DFH+15b], this will guarantee that our calibration algorithm generalizes given a set of $\tilde{O}(\log(|\mathcal{C}|)/\alpha^{5/2}\beta^{3/2})$ random samples.

Algorithm 3.2 only interacts with the sample through the guess-and-check oracle. Thus, to give a differentially private implementation of the algorithm, it suffices to give a differentially private implementation of the guess-and-check oracle [DR14].

Consider the sequence of queries that Algorithm 3.2 makes to the guess-and-check oracle. We say the sequence $\langle (S_1, v_1, \omega_1), \ldots, (S_k, v_k, \omega_k) \rangle$ is a (k, m)-sequence of guess-and-check queries if, over the course of the k queries, the response to at most m of the queries is some $r \in [0, 1]$, and the responses to the remaining queries are all ✓. We will assume that we know a lower bound on the minimum window $\omega = \min_{j \in [k]} \omega_j$ over all of the queries. We say that some algorithm $\mathcal{A}$ responds to a guess-and-check query (S, v, ω) according to a random sample X if its response satisfies the guess-and-check properties with $\sum_{i \in S} p_i^*$ replaced by its empirical estimate on X,

$$\hat{p}_S(X) = \frac{|S|}{|S \cap X|} \sum_{i \in S \cap X} o_i.$$

Responding to such a sequence in a differentially private manner can be achieved using techniques from the private multiplicative weights mechanism.

Theorem 3.8 ([HR10]). *Suppose $\varepsilon, \delta, \omega, \xi > 0$ and suppose $X \sim (\mathcal{X} \times \{0, 1\})^n$ is a set of n random samples. Then there exists an (ε, δ)-differentially private algorithm $\mathcal{A}$ that responds to any (k, m)-sequence of guess-and-check queries with minimum window ω according to X provided*

$$n = \Omega\left(\sqrt{\frac{\log(k/\xi) \cdot m \cdot \log(1/\delta)}{\varepsilon \cdot \omega^2}}\right)$$

with probability at least $1 - \xi$ over the randomness of $\mathcal{A}$.

Using this differentially private algorithm, we can apply generalization bounds based on privacy developed in [DFH+15c, BNS+16, DFH+15a, DFH+15b] to show that, with a modest increase in sample complexity, we can respond to all k guess-and-check queries.

Theorem 3.9. *Let $s_k = \langle (S_1, v_1, \hat{\omega}_1), \ldots, (S_k, v_k, \hat{\omega}_k) \rangle$ be a (k, m)-sequence of guess-and-check queries such that for all $j \in [k]$, $|S_j| = \beta_j N \geq \beta N$ and $\hat{\omega}_j = \Omega(\alpha\beta_j)$. Then there is an algorithm $\mathcal{A}$ that, given n random samples $X \sim (\mathcal{X} \times \{0, 1\})^n$, responds to s_k such that for all $j \in [k]$, the response $\mathcal{A}(S_j, v_j, \hat{\omega}_j; X)$ satisfies the guess-and-check properties with window $\omega_j = \alpha\beta_j$ provided*

$$n = \Omega\left(\frac{\log(|\mathcal{C}|/\alpha\beta\xi)}{\alpha^{5/2} \cdot \beta^{3/2}}\right)$$

with probability at least $1 - \xi$ over the randomness of $\mathcal{A}$ and the draw of X.

This theorem implies that, asymptotically, we can answer the k adaptively chosen guess-and-check queries with only a $\sqrt{1/\alpha\beta}$ factor increase in the sample complexity compared to if we knew the queries in advance. Theorem 3.9 follows from tailoring the proof of the main "transfer" theorem of [BNS+16] (Theorem 3.4) specifically to the requirements of our guess-and-check oracle and applying the differentially private mechanism described in Theorem 3.8. Combining these theorems and Algorithm 3.2 and the fact that $\beta = \alpha\lambda\gamma$, we obtain an algorithm for learning α-multicalibrated predictors from random samples.

Theorem 3.10. *Suppose $\alpha, \lambda, \gamma, \xi > 0$, and $\mathcal{C} \subseteq 2^{\mathcal{X}}$ where for all $S \in \mathcal{C}$, $|S| \geq \gamma N$. Then there is an algorithm that learns an (α, λ)-multicalibrated predictor with respect to $\mathcal{C}$ with probability at least $1 - \xi$ from $n = O\left(\frac{\log(|\mathcal{C}|/\alpha\lambda\gamma\xi)}{\alpha^4 \cdot \lambda^{3/2} \cdot \gamma^{3/2}}\right)$ samples.*

3.4 Runtime analysis of Algorithm 3.2

Here, we present a high-level runtime analysis of Algorithm 3.2 for learning an (α, λ)-calibrated predictor on $\mathcal{C}$. In Lemma 3.6, we claim an upper bound of $O(|\mathcal{C}|/\alpha^3\lambda^2\gamma)$ on the number of guess-and-check queries needed before Algorithm 3.2 converges. Here, we formally argue that each of these queries can be implemented in the random sample model without much overhead, which upper-bounds the running time of the algorithm overall. This upper bound is not immediate from our earlier analysis, as the sets and our predictor are represented implicitly as circuits.

Claim 3.11. *Algorithm 3.2 runs in time $O(|\mathcal{C}| \cdot t \cdot \text{poly}(1/\alpha, 1/\lambda, 1/\gamma))$, where t is an upper bound on the time it takes to evaluate set membership for $S \in \mathcal{C}$.*

Proof. As before, let $\beta = \alpha\lambda\gamma$. First, for each $S \in \mathcal{C}$, we need to evaluate $|S_v|$ for $S_v = \{i : x_i \in \lambda(v)\} \cap S$ for each of the $O(1/\lambda)$ values $v \in \Lambda[0, 1]$. We do this by sampling $i \sim \mathcal{X}$ and evaluating whether $i \in S$, and if so, checking the current value of x_i. Each of the membership queries takes at most t time and each evaluation of x_i takes at most $O(t/\alpha^2\beta)$ time by the same argument as our upper bound on the circuit size from Theorem 3.12. After $\tilde{O}(1/\lambda\beta^2)$ samples, we will be able to detect with constant probability which of the S_v have cardinality $|S_v| \geq \beta N$. Further, if $|S_v|$ is large, we can estimate $\bar{v}$ by evaluating the current predictor on samples from S_v, by rejection sampling. Similarly, to answer the guess-and-check queries, we will estimate the true empirical estimate of the query based on samples from S_v and respond based on a noisy comparison between the $\bar{v}$ and the estimate of $\sum_{i \in S_v} o_i$. These estimates can all be computed in $\text{poly}(1/\alpha, 1/\beta)$. Then, as discussed in the proof of Theorem 3.12, each update to the predictor can be implemented in time proportional to the bit complexity of the arithmetic computations, which is upper bounded by t. Repeating this process for each $S \in \mathcal{C}$ gives the upper bound of $O(|\mathcal{C}| \cdot t \cdot \text{poly}(1/\alpha, 1/\lambda, 1/\gamma))$. Finally, applying the upper bound on the number of guess-and-check queries from Lemma 3.6, the claim follows. □

3.5 The circuit complexity of multicalibrated predictors

As discussed in Section 1.2, an interesting corollary of our algorithm is a theorem about the complexity of representing a multicalibrated predictor. Indeed, from the definition of multicalibration

alone, it is not immediately clear that there should be succinct descriptions of multicalibrated predictors; after all, $\mathcal{C}$ could contain many sets. We argue that the cardinality of $\mathcal{C}$ is not the operative parameter in determining the circuit complexity of a predictor x that is multicalibrated on $\mathcal{C}$; instead it is the circuit complexity necessary to describe sets $S \in \mathcal{C}$, as well as the cardinality of the subsets in $\mathcal{C}$, and the degree of approximation.

Leveraging Lemma 3.6, we can see that Algorithm 3.2 actually gives us a way to build up a circuit that computes the mapping from individuals to the probabilities of our learned multicalibrated predictor x. Suppose that for all sets $S \in \mathcal{C}$, set membership can be determined by a circuit family of bounded complexity; that is, for all $S \in \mathcal{C}$, there is some c_S with size at most s, such that $c_S(i) = 1$ if and only if $i \in S$. Then we can use this family of circuits to build a circuit that implements x. We assume that we maintain real-valued numbers up to $b \geq \log(1/\alpha)$ bits of precision.

Theorem 3.12. *Suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is a collection of sets where each $S \in \mathcal{C}$ can be implemented by a boolean circuit c_S and for all $S \in \mathcal{C}$, the size of c_S is $O(s)$. Then there is a predictor that is α-multicalibrated on $\mathcal{C}$ implemented by a circuit of size $O((s+b)/\alpha^4\gamma)$. Further, Algorithm 3.2 can be used to learn such a circuit.*

Proof. We describe how to construct a circuit f_x that, on input i, will output the prediction x_i according to the predictor learned by our algorithm. We initialize f_x to be the constant function $f_x(i) = 1/2$ for all $i \in \mathcal{X}$. Throughout, we will update f_x *based on the current outputs of* f_x.

Consider an iteration of Algorithm 3.2 where for some S described by $c_S \in \mathcal{C}$, we update x based on a category $S_v = S \cap \{i : x_i \in \lambda(v)\}$. This occurs when the guess-and-check query returns some $r = \tilde{q}(S_v, \bar{v}, \omega) \in [0, 1]$. Our goal is to implement the update to x (i.e. update f_x), such that for all $i \in S_v$, the new value $x_i = r$ and all other values are unchanged.

We achieve this update by testing membership $i \in S$ and separately testing if the current value $f_x(i) = v$; if both tests pass, then we update the value output by $f_x(i)$ to be r. Specifically, we include a copy of c_S and hard-code v and $\delta_v = r - \bar{v}$ into the circuit; if $c_S(i) = 1$ and the current value of $f_x(i)$ is in $\lambda(v)$, then we update $f_x(i)$ to add the hardcoded δ_v to its current estimate of x_i; if either test fails, then $f_x(i)$ remains unchanged. This logic can be implemented with addition and subtraction circuits to a precision of λ with boolean circuits of size $O(b)$. We string these update circuits together, one for each iteration. Learning an $(\alpha/2, \alpha/2)$-multicalibrated predictor with Algoirthm 3.2 only requires $O(\alpha^4\gamma)$ updates. By this upper bound, we obtain an $O(\alpha^4\gamma)$ upper bound on the resulting circuit size. □

4 Multicalibration and weak agnostic learning

Note that in the algorithm and analysis in Section 3, we've assumed nothing about the structure of the underlying p^* or $\mathcal{C}$; the true probabilities could be adversarially chosen and yet, our algorithm guarantees α-multicalibration on $\mathcal{C}$. That said, the running time of the algorithm depends linearly on $|\mathcal{C}|$. As we imagine $\mathcal{C}$ to be a large, rich class of subsets of $\mathcal{X}$, in many cases linear depedence on $|\mathcal{C}|$ will be expensive. Thus, we turn our attention to when we can exploit structure within the collection of subsets $\mathcal{C}$ to speed up the learning process.

The main running time bottleneck in the algorithms arises from searching for some $S \in \mathcal{C}$ where calibration is violated. Without any assumptions about $\mathcal{C}$, we need to loop over the collection; however, if we can find such a set without looping over the entire collection of sets, then we would improve the running time of the algorithm. At a high level, we will show a connection between the agnostic learnability of $\mathcal{C}$ and the ability to speed up learning a multicalibrated predictor on $\mathcal{C}$. Imagining the sets $S \in \mathcal{C}$ as boolean concepts, we show that if it is possible to perform weak agnostic learning over a class $\mathcal{C}$ efficiently (in the sense of [KMV08, Fel10]), then there is an efficient search algorithm to find an update to the current predictor that will make progress towards multicalibration.

While there are some classes for which we have weak agnostic learners, in general, agnostic learning is considered a notoriously challenging problem. A natural question to ask is whether there is a way to speed up learning a multicalibrated predictor that does not involve agnostic learning. We answer this question in the negative. Roughly, we show that for a concept class $\mathcal{C}$, any predictor that is α-multicalibrated on the large sets of $\mathcal{C}$ can be used as the response to a query for distribution-specific weak agnostic learning on $\mathcal{C}$. In this sense, the reduction to weak agnostic learning is inherent; any efficient algorithm for multicalibration gives rise to an algorithm for weak agnostic learning.

In all, these results show that weak agnostic learning on a class $\mathcal{C}$ is equivalent to learning an α-multicalibrated predictor with respect to $\mathcal{C} = \{S \in \mathcal{C} : |S| \geq \gamma N\}$, the large sets defined by $\mathcal{C}$, up to polynomial factors in $1/\alpha, 1/\gamma$ where ρ and τ will be a function of α and γ.

4.1 Weak agnostic learning

For this discussion, we think of boolean concepts $c \in \mathcal{C}$ as $c : \mathcal{X} \to \{-1, 1\}$. We will overload the notions of a concept class $\mathcal{C}$ of boolean functions $c : \mathcal{X} \to \{-1, 1\}$ and our collection of subsets $\mathcal{C} \subseteq 2^{\mathcal{X}}$; in particular, there is a natural bijection between concepts and sets: a concept $c : \mathcal{X} \to \{-1, 1\}$ defines a set $S \subseteq 2^{\mathcal{X}}$ where $i \in S$ if $c(i) = 1$ and $i \notin S$ if $c(i) = -1$. We will connect the problem of finding a set $S \in \mathcal{C}$ on which a predictor x violates calibration to the problem of learning over the concept class $\mathcal{C}$ on a distribution $\mathcal{D}$.

For some distribution $\mathcal{D}$ supported on $\mathcal{X}$ and $x, y \in [-1, 1]^N$, let $\langle x, y \rangle_{\mathcal{D}} = \sum_{i \in \mathcal{X}} \mathcal{D}_i x_i y_i$. This inner product measures the correlation between x and y in $[-1, 1]^N$ over the discrete distribution $\mathcal{D}$. Throughout our discussion, we will focus on learning over the uniform distribution on $\mathcal{X}$ and drop explicit reference to $\mathcal{D}$. As per Remark 2, this may be a rich distribution over the features of individuals.

In our results, we will work with the *distribution-specific* weak agnostic learners of [Fel10][4].

Definition 4.1 (Weak agnostic learner)**.** *Let $\rho \geq \tau > 0$. Let $\mathcal{D}$ be a distribution supported on $\mathcal{X}$. A (ρ, τ)-*weak agnostic learner *$\mathcal{L}$ for $\mathcal{D}$ solves the following promise problem: given a collection of labeled samples $\{(i, y_i)\}$ where $i \sim \mathcal{D}$ and $y_i \in [-1, 1]$, if there is some $c \in \mathcal{C}$ such that $\langle c, y \rangle_{\mathcal{D}} > \rho$, then $\mathcal{L}$ returns some $h : \mathcal{X} \to [-1, 1]$ such that $\langle h, y \rangle_{\mathcal{D}} > \tau$.*

Intuitively, if there is a concept $c \in \mathcal{C}$ that correlates nontrivially with the observed labels, then

[4]Often, such learners are defined in terms of their error rates rather than correlations; the definitions are equivalent up to factors of 2 in ρ and τ. Also, we will always work with a hypothesis class $\mathcal{H} = [-1, 1]^{\mathcal{X}}$ the set of functions from $\mathcal{X}$ to $[-1, 1]$, so we fix this class in the definition.

the weak agnostic learner returns a hypothesis h (not necessarily from $\mathcal{C}$), that is also nontrivially correlated with the observed labels. In particular, ρ and τ are typically taken to be $\rho = 1/p(d)$ and $\tau = 1/q(d)$ for polynomials $p(d) \leq q(d)$, where $d = \log(|\mathcal{X}|)$.

4.2 Multicalibration from weak agnostic learning

In this section, we show how we can use a weak agnostic learner to solve the search problem that arises at each iteration of Algorithm 3.2: namely, to find an update that will make progress towards mulitcalibration. Formally, we show the following theorem.

Theorem 4.2 (Formal statement of Theorem 3)**.** *Let $\rho, \tau > 0$ and $\mathcal{C} \subseteq 2^{\mathcal{X}}$ be some concept class. If $\mathcal{C}$ admits a (ρ, τ)-weak agnostic learner that runs in time $T(|\mathcal{C}|, \rho, \tau)$, then there is an algorithm that learns a predictor that is (α, λ)-multicalibrated on $\mathcal{C}' = \{S \in \mathcal{C} : |S| \geq \gamma N\}$ in time $O(T(|\mathcal{C}|, \rho, \tau) \cdot \text{poly}(1/\alpha, 1/\lambda, 1/\gamma))$ as long as $\rho \leq \alpha^2\lambda\gamma/2$ and $\tau \geq \text{poly}(1/\alpha, 1/\lambda, 1/\gamma)$.*

That is, if there is an algorithm for learning the concept class $\mathcal{C}$ over the hypothesis class of real-valued functions $\mathcal{H} = \{h : \mathcal{X} \to [-1, 1]\}$ on the distribution of individuals in polynomial time in $\log(|\mathcal{C}|), 1/\rho$, and $1/\tau$, then there is an algorithm for learning an α-multicalibrated predictor on the large sets in $\mathcal{C}$ that runs in time polynomial in $\log(|\mathcal{C}|), 1/\alpha, 1/\lambda, 1/\gamma$. For clarity of presentation in the reduction, we make no attempts to optimize the sample complexity or running time. Indeed, the exact sample complexity and running time will largely depend on how strong the weak learning guarantee is for the specific class $\mathcal{C}$.

We prove Theorem 4.2 by using the weak learner for $\mathcal{C}$ to learn a (α, λ)-calibrated predictor. Recall Algorithm 3.2: we maintain a predictor x and iteratively look for a set $S \in \mathcal{C}$ where x violates the calibration constraint on $S_v = \{i : x_i \in \lambda(v)\} \cap S$ for some value v. In fact, the proof of Lemma 3.6 reveals that we are not restricted to updates on S_v for $S \in \mathcal{C}$. As long as there is some uncalibrated category S_v, we can find an update that makes nontrivial progress in ℓ_2^2 distance from p^* – even if this update is not on any $S \in \mathcal{C}$ – then we can bound the number of iterations it will take before there are no more uncalibrated categories. We show that a weak agnostic learner allows us to find such an update.

Proof. Throughout the proof, let $\beta = \alpha\lambda\gamma$, $\rho = \alpha\beta/2$, and $\tau = \rho^d$ for some constant $d \geq 1$. Let x be a predictor initialized to be the constant function $x_i = 1/2$ for all $i \in \mathcal{X}$.

Consider the search problem that arises during Algorithm 3.2 immediately after updating the predictor x. Let $\mathcal{X}_v = \{i : x_i \in \lambda(v)\}$ be the set of individuals in the λ-interval surrounding v. Our goal is to determine if there is some $v \in \Lambda[0, 1]$ and $S \in \mathcal{C}$ such that $|S_v| \geq \beta N$, where

$$\left| \sum_{i \in S_v} x_i - p_i^* \right| \geq \alpha |S_v| . \tag{1}$$

We reduce this search problem to the problem of weak agnostic learning over $\mathcal{C}$ on the distribution $\mathcal{D}_X$. For any $v \in \Lambda[0, 1]$, if $|X_v| < \beta N$, then clearly there is no uncalibrated category S_v with $|S_v| \geq \beta N$; for each $v \in \Lambda[0, 1]$, we will test if $\mathcal{X}_v$ is large enough by taking $O(\log(1/\beta\xi)/\beta)$ random draws from $\mathcal{X}$.

Supposing $\mathcal{X}_v$ is large enough, we take a fresh sample of size $n \geq \tilde{O}(\log(|\mathcal{C}/\xi|)/\beta^2\tau^2)$. We take n large enough that over all categories S_v of $|S_v| \geq \beta N$, the observable statistics deviate from their expectation by at most $\tau/4$:

$$\left| \frac{1}{n} \sum_{j \in [n]} o_j - \frac{1}{|S_v|} \sum_{i \in S_v} p_i^* \right| \leq \tau/4 \tag{2}$$

Additionally, assume that x is overall observably $\tau/4$-calibrated with respect to $\mathcal{X}$ (recall, this means calibrated on the set of observations). Note that observable calibration on $\mathcal{X}$ implies that for each $v \in \Lambda[0,1]$,

$$\left| \frac{1}{n} \sum_{j \in [n]} (o_j - x_j) \right| \leq \tau/4. \tag{3}$$

(If x is not $\tau/4$-calibrated then for the $\mathcal{X}_v$ that violates calibration, offset all the values of x_i from their current values such that $\left|\sum_{i \in \mathcal{X}} x_i - o_i\right| \leq \tau N/4$ and resample; as in Algorithm 3.1, this will make at least $\Omega(\tau^2)$ progress towards p^*).

For each $v \in \Lambda[0,1]$, we consider the following learning problem. For $i \in \mathcal{X}_v$, let $\Delta_i = \frac{x_i - o_i}{2}$. For $i \in \mathcal{X} \setminus \mathcal{X}_v$, let $\Delta_i = 0$. We claim that if there is some S_v satisfying (1), then for $i \sim \mathcal{D}_{\mathcal{X}}$, the labeled samples of either (i, Δ_i) or $(i, -\Delta_i)$ satisfy the weak learning promise for $\rho = \alpha\beta/2$.

Claim 4.3. *Let $c_S : \mathcal{X} \to \{-1, 1\}$ be the boolean function associated with some $S \in \mathcal{C}$. For $v \in \Lambda[0,1]$, if $S_v = \{i : x_i \in \lambda(v)\} \cap S$ satisfies $\sum_{i \in S_v}(x_i - p_i^*) \geq \alpha\,|S_v|$, then*

$$\langle c_S, \Delta \rangle_{\mathcal{D}_{\mathcal{X}}} \geq \rho.$$

Note that the supposition of the claim is satisfied when (1) holds without the absolute value. In the case where (1) holds in the other direction, the claim will hold for $-\Delta$. The argument will be identical.

$$\begin{aligned}
\langle c_S, \Delta \rangle_{\mathcal{D}_{\mathcal{X}}} &= \frac{1}{N} \sum_{i \in \mathcal{X}} \left(\frac{x_i - o_i}{2} \right) \cdot c_S(i) \\
&= \frac{1}{N} \sum_{i \in \mathcal{X}_v} \left(\frac{x_i - o_i}{2} \right) \cdot c_S(i) + \sum_{i \in \mathcal{X} \setminus \mathcal{X}_v} 0 \cdot c_S(i) \\
&= \frac{1}{N} \left(\sum_{i \in S_v} \left(\frac{x_i - o_i}{2} \right) - \sum_{i \in \mathcal{X}_v \setminus S_v} \left(\frac{x_i - o_i}{2} \right) \right) \\
&= \frac{1}{N} \left(\sum_{i \in S_v} \left(\frac{x_i - o_i}{2} \right) - \left(\sum_{i \in \mathcal{X}_v} \left(\frac{x_i - o_i}{2} \right) - \sum_{i \in S_v} \left(\frac{x_i - o_i}{2} \right) \right) \right) \\
&\geq \frac{2}{N} \sum_{i \in S_v} \left(\frac{x_i - o_i}{2} - \tau\,|\mathcal{X}_v|\,/4 \right) && (4) \\
&\geq \frac{2}{N} (\alpha\beta N - \tau N/4) && (5) \\
&\geq 2\rho - \tau/2 && (6)
\end{aligned}$$

where the inequality (4) follows from (3), (5) follows from the assumption that $|S_v| \geq \beta N$ and our assumption on $\sum_{i \in S_v}(x_i - p_i^*)$, and (6) follows from our assumption that the sample size is large enough to guarantee at most $\tau/8$ error. Noting that $\tau/2 \leq \rho$ gives the claim.

Thus, because the (ρ, τ)-weak agnostic learning promise is satisfied, the learner will return to us some $h : \mathcal{X} \to [-1, 1]$ satisfying the following inequality.

$$\begin{aligned}\tau &\leq \langle \Delta_i, h_i \rangle_{\mathcal{D}_v} \\ &= \frac{1}{2\left|\mathcal{X}\right|} \sum_{i \in \mathcal{X}_v} (x_i - o_i) \cdot h_i \\ &\leq \frac{1}{2\left|\mathcal{X}\right|} \sum_{i \in \mathcal{X}_v} (v - p_i^*) \cdot h_i + \tau/8.\end{aligned}$$

where the final inequality follows by the assumed statistical accuracy. This shows that the h returned to us by the weak agnostic learner is nontrivally correlated with $x - p^*$ on $\mathcal{X}_v$. In particular, if we use this h as a gradient step, updating $x_i \to v - \eta h_i$ (projecting onto $[0, 1]$ if necessary) for $\eta = \Omega(\tau/\beta)$, then we can guarantee that each such update will achieve $\tau^2 N$ progress in $\|x - p^*\|^2$. The analysis follows in the same way as the analysis of Algorithm 3.2. □

4.3 Weak agnostic learning from multicalibration

In this section, we show the converse reduction. In particular, we will show that for a concept class $\mathcal{C}$, an efficient algorithm for obtaining an α-multicalibrated predictor with respect to $\mathcal{C}' = \{S \in \mathcal{C} : |S| \geq \gamma N\}$, gives an efficient algorithm for responding to weak agnostic learning queries on $\mathcal{C}$. In fact, we will show that we can obtain a weak agnostic learner even by learning a multi-AE predictor.

Theorem 4.4. *Let $\alpha, \gamma > 0$ and suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is a concept class. If there is an algorithm for learning an α-multicalibrated predictor on $\mathcal{C}' = \{S \in \mathcal{C} : |S| \geq \gamma N\}$ in time $T(|C|, \alpha, \gamma)$ then we can implement a (ρ, τ)-weak agnostic learner for $\mathcal{C}$ in time $O(T(|C|, \alpha, \gamma) \cdot \mathrm{poly}(1/\tau))$ for any $\rho, \tau > 0$ such that $\tau \leq \min\{\rho - 2\gamma, \rho - 6\alpha\}$.*

Proof. Suppose we want to weak agnostically learn over $\mathcal{C}$ on sampled observations from $y \in [-1, 1]^N$. We assume there is some $c_S \in \mathcal{C}$ such that $\langle c_S, y \rangle > 1/2 + \rho$. There are two cases to handle. First, suppose the support of c_S is small; that is, for the corresponding $S \in \mathcal{C}$, $|S| < \gamma$.

Then, consider the correlation between y and the the constant hypothesis $h(i) = -1$ for all $i \in \mathcal{X}$.

$$\begin{aligned}
\langle y, -1\rangle &= -\sum_{i\in\mathcal{X}} y_i \\
&= -\sum_{i\in S} y_i - \sum_{i\in\mathcal{X}\setminus S} y_i \\
&\geq -\gamma - \sum_{i\in\mathcal{X}\setminus S} y_i \\
&= -\gamma + (\langle c_S, y\rangle - \sum_{i\in S} y_i) \\
&\geq -2\gamma + \langle c_S, y\rangle \\
&\geq \rho - 2\gamma
\end{aligned}$$

Thus, for $\tau < \rho - 2\gamma$, in the case when the support of c_S is small, then we can return the hypothesis -1. We can test if the constant hypothesis is sufficiently correlated with y in $\text{poly}(1/\tau)$ time by random sampling.

Turning to the case when c_S defines a set where $|S| \geq \gamma N$, we show that responding with a multicalibrated predictor will satisfy the weak agnostic learning guarantee. To do this, we prove the following lemma.

Lemma 4.5. *Let $\alpha, \rho > 0$ and suppose $\mathcal{C}' \subseteq 2^{\mathcal{X}}$. Suppose x is an α-multi-AE predictor with respect to $\mathcal{C}' \cup \{\mathcal{X}\}$. Then, if there is some $S \in \mathcal{C}$ such that the associated circuit c_S satisfies $\langle c_S, 2p^* - 1\rangle \geq \rho$, then $\langle c_S, 2x - 1\rangle \geq \tau$ for $\tau \leq \rho - 6\alpha$.*

This lemma implies that for sufficiently large ρ, the ability to learn an α-multi-AE predictor with respect to $\mathcal{C}' = \{S \in \mathcal{C} : |S| \geq \gamma N\}$ gives a way to answer weak agnostic learning queries. In particular, given labeled samples for agnostic learning, we can let the labels define p^*, learn a multicalibrated predictor, and use these predictions as the weak agnostic learning hypothesis.

Proof. Consider $N \cdot \langle c_S, p^* - x\rangle$.

$$\begin{aligned}
\sum_{i\in\mathcal{X}} c_S(i) \cdot (p_i^* - x_i) &= \sum_{i\in S}(p_i^* - x_i) - \sum_{i\in\mathcal{X}\setminus S}(p_i^* - x_i) \\
&= 2\sum_{i\in\mathcal{S}}(p_i^* - x_i) - \sum_{i\in\mathcal{X}}(p_i^* - x_i) \\
&\leq 3\alpha N
\end{aligned}$$

where the final inequality follows by the assumption that $S \in \mathcal{C}$ and x is α-multi-AE on $\mathcal{C} \cup \{\mathcal{X}\}$. The lemma follows by linearity and rearranging. □

Thus, for $\tau \leq \min\{\rho - 2\gamma, \rho - 6\alpha\}$, if we can learn an α-multicalibrated predictor on sets $\mathcal{C}' = \{S \in \mathcal{C} : |S| \geq \gamma N\}$, then we can implement a (ρ, τ)-weak agnostic learner on $\mathcal{C}$. □

5 Multicalibration achieves "best-in-class" prediction

While our notion of multicalibration provides a protection against discrimination for groups, we argue that this protection comes at virtually no cost in the utility of the predictor. In fact, we argue that Algorithm 3.2 can be used as an effective post-processing step to turn any predictor, or family of predictors, into a multicalibrated predictor that achieves comparable (or improved) prediction error.

Suppose we are given a collection $\mathcal{C}$ of sets of individuals on which we wish to be multicalibrated. Additionally, suppose we have a collection of candidate predictors $\mathcal{H}$, which achieves low prediction error but might violate calibration arbitrarily. From these collections, we would like to produce a predictor x that is α-multicalibrated on $\mathcal{C}$ but achieves prediction error commensurate with the best predictor in $\mathcal{H}$; in particular, $\|x - p^*\|^2$ should be not much larger than $\|h^* - p^*\|^2$ (and ideally would be smaller). In this sense, the calibrated x would not only be fair, but would also achieve (approximately) best-in-class prediction error over $\mathcal{H}$.

Consider some $h \in \mathcal{H}$ and consider the partition of $\mathcal{X}$ into sets according to the predictions of h – in particular, we will first apply a λ-discretization to the range of each h to partition $\mathcal{X}$ into categories. That is, let $S_v(h) = \{i : h_i \in \lambda(v)\}$, and note that $S_v(h)$ is disjoint from $S_{v'}(h)$ for $v \neq v'$, and $\bigcup_{v \in \Lambda[0,1]} S_v(h) = \mathcal{X}$. In addition to calibrating with respect to $S \in \mathcal{C}$, we can also ask for calibration on $S_v(h)$ for all $h \in \mathcal{H}$ and $v \in \Lambda[0,1]$. Specifically, let $\mathcal{S}(\mathcal{H}) = \{S_v(h)\}_{h \in \mathcal{H}, v \in \Lambda[0,1]}$; we consider imposing calibration on $\mathcal{C} \cup \mathcal{S}(\mathcal{H})$. Calibrating in this manner protects the groups defined by $\mathcal{C}$ but additionally gives a strong utility guarantee.

Theorem 5.1 (Best-in-class prediction)**.** *Suppose $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is a collection of subsets of $\mathcal{X}$ and $\mathcal{H}$ is a set of predictors. Then there is a predictor x that is α-multicalibrated on $\mathcal{C}$ such that*

$$\|x - p^*\|^2 - \|h^* - p^*\|^2 < 6\alpha N,$$

where $h^* = \operatorname{argmin}_{h \in \mathcal{H}} \|h - p^*\|^2$. Further, suppose that for all $S \in \mathcal{C}$, $|S| \geq \gamma N$, and suppose that set membership for $S \in \mathcal{C}$ and $h \in \mathcal{H}$ are computable by circuits of size at most s; then x is computable by a circuit of size at most $O(s/\alpha^4\gamma)$.

The proof of Theorem 5.1 actually reveals something stronger: if x is calibrated on the set $\mathcal{S}(\mathcal{H})$, then for every category $S_v(h) \in \mathcal{S}(\mathcal{H})$, if x is significantly different from h on this category – that is, if $\sum_{i \in S_v(h)} (h_i - x_i)^2$ is large – then x actually achieves significantly improved prediction error on this category compared to h. This is stated formally in Lemma 5.2.

Lemma 5.2. *Suppose y is an arbitrary predictor and let $\mathcal{S}(y) = \{S_v(y)\}_{v \in \Lambda[0,1]}$. Suppose x is an arbitrary α-multicalibrated predictor on $\mathcal{S}(y)$. Then for all $v \in \Lambda[0,1]$,*

$$\sum_{i \in S_v(y)} \left((y_i - p_i^*)^2 - (x_i - p_i^*)^2\right) \geq \sum_{i \in S_v(y)} (v - x_i)^2 - (4\alpha + \lambda) \left|S_v(y)\right|.$$

Consequently,

$$\|y - p^*\|^2 - \|x - p^*\|^2 \geq \|x - y\|^2 - (4\alpha + \lambda)N.$$

This lemma shows that calibrating on the categories of a predictor not only prevents the squared prediction error from degrading beyond a small additive approximation, but it also guarantees that

if calibrating changes the predictor significantly on any category, this change represents significant progress towards the true underlying probabilities on this category. Assuming Lemma 5.2, Theorem 5.1 follows.

Proof of Theorem 5.1. Note that if x is α-multicalibrated on $\mathcal{C}$, then x is α-multicalibrated on any $\mathcal{C}' \subseteq \mathcal{C}$. Consider enforcing calibration on the collection $\mathcal{C} \cup \mathcal{S}(\mathcal{H})$ as defined above. If x is α-calibrated on $\mathcal{C} \cup \mathcal{S}(\mathcal{H})$ then it is α-multicalibrated on $\{S_v(h)\}_{v \in \Lambda[0,1]}$ for all $h \in \mathcal{H}$ and specifically for h^*. By Lemma 5.2, and the fact that the squared difference is nonnegative, we obtain the following inequality:

$$\|h^* - p^*\|^2 - \|x - p^*\|^2 \geq \|x - h^*\|^2 - (4\alpha + \lambda)N \geq -(4\alpha + \lambda)N.$$

This inequality suffices to prove the accuracy guarantee; however, to also guarantee the predictor x can be implemented by a small circuit, we have to be a bit more careful. In particular, when calibrating, we will ignore any $S_v(h)$ such that $|S_v(h)| < \lambda\alpha N$. Note that because we have λ-discretized, there are at most $1/\lambda$ categories; thus, excluding the sets $S_v(h)$ where $|S_v(h)| < \alpha\lambda N$ introduces at most an additional αN error. Taking $\lambda = \alpha$, in turn, this implies that the difference in squared prediction error can be bounded as $\|x - p^*\|^2 - \|h^* - p^*\|^2 \leq 6\alpha N$. Finally, because the sets we want to calibrate on are at least $\alpha^2\gamma N$ in cardinality, the circuit complexity bound follows by applying Algorithm 3.2 and Theorem 3.12. □

Thus, given any method for learning an accurate predictor h, we can turn it into a method for learning a fair and accurate predictor h' by running Algorithm 3.2 on the set of categories of h. Combined with Theorem 3.12, this theorem shows that for any such class of predictors $\mathcal{H}$ of bounded complexity, there exists a calibrated predictor with similar circuit complexity that performs nearly as well as the best $h \in \mathcal{H}$ in terms of accuracy. Further, by Lemma 5.2, this (nearly) best-in-class property will hold not just over the entire domain $\mathcal{X}$, but on every sufficiently large category $S_v(h)$ identified by some $h \in \mathcal{H}$. That is, if x is calibrated on $\mathcal{S}(\mathcal{H})$, then for every category $S_v(h)$, the average squared prediction error $\mathbb{E}_{i \in S_v(h)}\left[(x_i - p_i^*)^2\right]$ will be at most 6α worse than prediction given by h on this set. If we view $\mathcal{H}$ as defining a set $\mathcal{S}(\mathcal{H})$ of "computationally-identifiable" categories, then we can view any predictor that is calibrated on $\mathcal{S}(\mathcal{H})$ as at least as fair and at least as accurate on this set of computationally-identifiable categories as the predictor that identified the group (up to some small additive approximation).

We turn to proving Lemma 5.2. The lemma follows by expanding the difference in squared prediction errors and invoking the definition of α-calibration.

Proof of Lemma 5.2. Let S_{vu} represent the set of individuals i where $y \in \lambda(v)$ and x assigns value u. By the assumption that x is α-calibrated on $\mathcal{S}(y)$, we know for every $S_v(y) \in \mathcal{S}(y)$, there is some subset $S'_v(y) \subseteq S_v(y)$ such that $|S'_v(y)| \geq (1 - \alpha)|S_v(y)|$ for which x's predictions are approximately correct. In particular, let $S'_{vu} = S'_v(y) \cap S_u(x)$; if x is α-calibrated with respect to $S_v(y)$, this guarantees that for all values $u \in [0, 1]$, we have

$$\left| \sum_{i \in S'_{vu}} p_i^* - u \right| \leq \alpha \left| S'_{vu} \right|. \tag{7}$$

Using this fact, and the fact that the remaining α-fraction of $S_v(y)$ can contribute at most $\alpha|S_v(y)|$

to the squared error, we can express the difference in the squared errors of y and x on $S_v(y)$:

$$\begin{aligned}\sum_{i\in S_v(y)}(y_i-p_i^*)^2-\sum_{i\in S_v(y)}(x_i-p_i^*)^2 &= \sum_{i\in S_v(y)}(v-p_i^*+(y_i-v))^2-\sum_{i\in S_v(y)}(x_i-p_i^*)^2\\ &= \sum_{i\in S_v(y)}(v-p_i^*)^2-\sum_{i\in S_v(y)}(x_i-p_i^*)^2+2\sum_{i\in S_v(y)}(v-p_i^*)(y_i-v)\\ &\geq \sum_{i\in S_v(y)}(2(p_i^*-v)(x_i-v)-(x_i-v)^2)-\lambda\left|S_v(y)\right|. \qquad (8)\end{aligned}$$

where (8) follows by the observation that if $y_i \in \lambda(v)$, then $|y_i - v| \leq \lambda/2$ and $|v - p_i^*|$ is trivially bounded by 1. We bound the sum over $i \in \mathcal{X}$ of the first term:

$$\begin{aligned}\sum_{i\in S_v(y)}(p_i^*-v)(x_i-v) &= \sum_{u\in[0,1]}\sum_{i\in S_{vu}}(p_i^*-v)(u-v)\\ &= \sum_{u\in[0,1]}(u-v)\sum_{i\in S_{vu}}(p_i^*-v)\\ &= \sum_{u\in[0,1]}(u-v)\sum_{i\in S_{vu}}(u-v+p_i^*-u)\\ &= \sum_{u\in[0,1]}\left(\left|S_{vu}\right|(u-v)^2+(u-v)\sum_{i\in S_{vu}}(p_i^*-u)\right).\end{aligned}$$

At this point, we note that $|u - v| \leq 1$. Thus, we can bound the contribution of the sum over S_{vu} by its negative absolute value:

$$\begin{aligned}&\geq \sum_{u\in[0,1]}\left(\left|S_{vu}\right|(u-v)^2-|u-v|\left|\sum_{i\in S_{vu}}(p_i^*-u)\right|\right)\\ &\geq \sum_{u\in[0,1]}\left(\left|S_{vu}\right|(u-v)^2-\left|\sum_{i\in S'_{vu}}(p_i^*-u)+\sum_{i\in S_{vu}\setminus S'_{vu}}(p_i^*-u)\right|\right)\\ &\geq \sum_{u\in[0,1]}\left(\left|S_{vu}\right|(u-v)^2-\left|\sum_{i\in S'_{vu}}(p_i^*-u)\right|-\alpha\left|S_v(y)\right|\right)\\ &\geq \sum_{u\in[0,1]}\left|S_{vu}\right|(u-v)^2-2\alpha\left|S_v(y)\right|\\ &= \sum_{i\in S_v(y)}(v-x_i)^2-2\alpha\left|S_v(y)\right|,\end{aligned}$$

where we bound the sums over S_{vu} by invoking α-calibration and applying (7). Plugging this bound into (8), we see that

$$\begin{aligned}\sum_{i\in S_v(y)}\left((y_i-p_i^*)^2-(x_i-p_i^*)^2\right) &\geq 2\left(\sum_{i\in S_v(y)}(v-x_i^*)^2-2\alpha\left|S_v(y)\right|\right)-\lambda\left|S_v(y)\right|-\sum_{i\in S_v(y)}(v-x_i)^2\\ &= \sum_{i\in S_v(y)}(v-x_i^*)^2-(4\alpha-\lambda)\left|S_v(y)\right|.\end{aligned}$$

Summing over $v \in [0,1]$, we can conclude

$$\|y - p^*\|^2 - \|x - p^*\|^2 \geq \|x - y\|^2 - (4\alpha - \lambda)N$$

showing the lemma. □

Acknowlegments The authors would like to thank Cynthia Dwork, Roy Frostig, Parikshit Gopalan, Moritz Hardt, Aditi Raghunathan, Jacob Steinhardt, and Greg Valiant for helpful discussions related to this work.

References

[BDLM01] Shai Ben-David, Philip Long, and Yishay Mansour. Agnostic boosting. In *Computational Learning Theory*, pages 507–516. Springer, 2001.

[BH15] Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *International Conference on Machine Learning*, pages 1006–1014, 2015.

[BHJ+17] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

[BNS+16] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1046–1059. ACM, 2016.

[BR17] Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In *Tutorials on the Foundations of Cryptography*, pages 79–158. Springer, 2017.

[CDPF+17] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. *arXiv preprint arXiv:1701.08230*, 2017.

[CG17] Alexandra Chouldechova and Max G'Sell. Fairer and more accurate, but for whom? *arXiv preprint arXiv:1707.00046*, 2017.

[Cho17] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *arXiv preprint arXiv:1703.00056*, 2017.

[CSV17] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*, 2017.

[DFH+15a] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, pages 2350–2358, 2015.

[DFH+15b] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

[DFH+15c] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126. ACM, 2015.

[DHP+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science (ITCS)*, pages 214–226, 2012.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[Fel10] Vitaly Feldman. Distribution-specific agnostic boosting. In *Proceedings of the First Symposium on Innovations in Computer Science10*, 2010.

[FKL16] Benjamin Fish, Jeremy Kun, and Ádám D. Lelkes. A confidence-based approach for balancing fairness and accuracy. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 144–152. SIAM, 2016.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. In *Foundations of Computer Science, 1984. 25th Annual Symposium on*, pages 464–479. IEEE, 1984.

[GRU12] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. *Theory of Cryptography*, pages 339–356, 2012.

[Hau92] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[HPS16] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016.

[HR10] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 61–70. IEEE, 2010.

[HRRU14] Justin Hsu, Aaron Roth, Tim Roughgarden, and Jonathan Ullman. Privately solving linear programs. In *International Colloquium on Automata, Languages, and Programming*, pages 612–624. Springer, 2014.

[JKM+17] Matthew Joseph, Michael Kearns, Jamie H. Morgenstern, Seth Neel, and Aaron Roth. Fair algorithms for infinite and contextual bandits. *arXiv:1610.09559*, 2017.

[JKMR16] Matthew Joseph, Michael Kearns, Jamie H. Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016.

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[KK09] Varun Kanade and Adam Kalai. Potential-based agnostic boosting. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 880–888. Curran Associates, Inc., 2009.

[KMR16] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

[KMV08] Adam Tauman Kalai, Yishay Mansour, and Elad Verbin. On agnostic boosting and parity learning. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 629–638. ACM, 2008.

[KNRW17] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *arXiv preprint arXiv:1711.05144*, 2017.

[KRCP+17] Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. *arXiv preprint arXiv:1706.02744*, 2017.

[KSS94] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[KV94] Michael J. Kearns and Umesh Virkumar Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[LRD+17] Yang Liu, Goran Radanovic, Christos Dimitrakakis, Debmalya Mandal, and David C. Parkes. Calibrated fairness in bandits. *arXiv preprint arXiv:1707.01875*, 2017.

[PRW+17] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. On fairness and calibration. *arXiv preprint arXiv:1709.02012*, 2017.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[SS12] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[Ull15] Jonathan Ullman. Private multiplicative weights beyond linear queries. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 303–312. ACM, 2015.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[WS17] Blake Woodworth and Nathan Srebro. Lower bound for randomized first order convex optimization. *arXiv preprint arXiv:1709.03594*, 2017.