Loss Minimization Yields Multicalibration for Large Neural Networks

Jarosław Błasiok
ETH Zürich, Switzerland

Parikshit Gopalan
Apple, Palo Alto, CA, USA

Lunjia Hu
Stanford University, CA, USA

Adam Tauman Kalai
Microsoft Research, Cambridge, MA, USA

Preetum Nakkiran
Apple, Palo Alto, CA, USA

Abstract

Multicalibration is a notion of fairness for predictors that requires them to provide calibrated predictions across a large set of protected groups. Multicalibration is known to be a distinct goal than loss minimization, even for simple predictors such as linear functions.

In this work, we consider the setting where the protected groups can be represented by neural networks of size k, and the predictors are neural networks of size $n > k$. We show that minimizing the squared loss over all neural nets of size n implies multicalibration for all but a bounded number of *unlucky* values of n. We also give evidence that our bound on the number of unlucky values is tight, given our proof technique. Previously, results of the flavor that loss minimization yields multicalibration were known only for predictors that were near the ground truth, hence were rather limited in applicability. Unlike these, our results rely on the expressivity of neural nets and utilize the representation of the predictor.

2012 ACM Subject Classification Theory of computation → Machine learning theory; Computing methodologies → Neural networks

Keywords and phrases Multi-group fairness, loss minimization, neural networks

Digital Object Identifier 10.4230/LIPIcs.ITCS.2024.17

Related Version *arXiv Version*: `https://arxiv.org/abs/2304.09424`

Funding *Jarosław Błasiok*: JB is supported by a Junior Fellowship from the Simons Society of Fellows.
Lunjia Hu: Part of this work was performed while LH was interning at Apple. LH is also supported by Moses Charikar's and Omer Reingold's Simons Investigators awards, Omer Reingold's NSF Award IIS-1908774, and the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.

Acknowledgements We thank Ryan O'Donnell and Rocco Servedio for suggesting the current proof of Lemma 14. We also thank Barry-John Theobald for comments on an early draft.

1 Introduction

In supervised binary prediction, we are given examples $(x, y) \in \mathcal{X} \times \{0, 1\}$ drawn independently from an unknown distribution $\mathcal{D}$, where the labels $y \in \{0, 1\}$ are binary. We wish to learn a predictor $f : \mathcal{X} \to [0, 1]$ which assigns to each point $x \in \mathcal{X}$ a prediction $f(x) \in [0, 1]$ as the estimated probability that the label is 1. The performance of a prediction model is commonly measured using a loss function; popular losses include the squared loss and the cross-entropy

15th Innovations in Theoretical Computer Science Conference (ITCS 2024).
Editor: Venkatesan Guruswami; Article No. 17; pp. 17:1–17:21
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

loss. Machine learning algorithms learn a predictor f by iteratively optimizing the expected loss (e.g. via stochastic gradient descent). This simple paradigm has proved remarkably powerful, and modern machine learning has a powerful arsenal of theoretical and empirical tools for loss minimization.

Yet despite its considerable success, loss minimization alone typically does not guarantee everything we might want from our prediction models. It is not obvious if important desiderata such as fairness, privacy and interpretability can be guaranteed just from loss minimization. There has been considerable research effort dedicated to understanding how to modify the loss minimization template to ensure these desiderata, and into possible tradeoffs between these goals and loss minimization.

A desirable fairness guarantee that has been studied intensively in recent years is *multicalibration*, introduced in the work of Hébert-Johnson et el. [23] (see also [31, 35]). Informally, it asks that the predictions be calibrated conditioned on each subgroup of the population $\mathcal{X}$ for a family of subgroups.

▶ **Definition 1** (Multicalibration [33, 23]). *Let $\gamma > 0$ and $\mathcal{C}$ be a class of auditor functions $c : \mathcal{X} \times [0,1] \to [-1,1]$. The predictor $f : \mathcal{X} \to [0,1]$ is $(\mathcal{C},\gamma)$-multicalibrated or $(\mathcal{C},\gamma)$-MC if for all $c \in \mathcal{C}$,*

$$\left| \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[c\big(x, f(x)\big) \cdot \big(y - f(x)\big)\right] \right| \leq \gamma. \tag{1}$$

When $\mathcal{C}$ is clear from context, we say that f is γ-MC for brevity. [1]

When c maps to $\{0,1\}$ values, it can be viewed as defining a *group* which is a subset of the domain $\mathcal{X}$ and MC means that $\mathbb{E}[y] \approx \mathbb{E}[f(x)]$ conditioned on membership in this group, where the degree of closeness depends on the size of the group. Good calibration is guaranteed for any group of sufficient likelihood that can be identified by the auditor family $\mathcal{C}$. The connection to calibration comes from observing that taking $c(x, f(x)) = \mathbf{1}[f(x) = v]$ yields a contribution of at most γ to the standard notion of expected calibration error (ECE) from the individuals x with $f(x) = v$, and taking $c(x, f(x)) = \eta(f(x))$ for all choices of $\eta : [0,1] \to [-1,1]$ yields having ECE $\leq \gamma$.

1.1 Loss Minimization and Multicalibration

Exploring the relation between multicalibration and loss minimization has led to a rich body of research work [23, 19, 17, 20, 27, 15]. Let us describe the main findings from this body of work and how it relates to the question that we address.

The original paper of [23] showed that multicalibration can be achieved via a post-processing procedure: we can take an arbitrary predictor f and *boost* it to a $\mathcal{C}$-multicalibrated predictor f' such that the squared loss (or any proper loss) only decreases. This suggests that multicalibration need not come at the cost of increased expected loss, provided we are willing to consider models of greater complexity (than f and $\mathcal{C}$). Indeed, during the boosting, the [23] algorithm repeatedly refines the predictor by augmenting it with functions from the base class $\mathcal{C}$, thus increasing its accuracy and its complexity. The connection to boosting is made even more explicit in [19, 16]. Given this, a natural question which motivates our work is:

[1] The MC definition above is version due to Appendix 1 in [33], which was also used by [9, 13]. The original definition [23, 21] considers only those c that can be factored as $c(x, f(x)) = g(x)w(f(x))$ where $g : \mathcal{X} \to [-1,1]$ and $w : [0,1] \to [-1,1]$. For the setting where $\mathcal{C}$ is a family of neural nets, the general definition is more natural, since it amounts to both x and $f(x)$ being inputs to an auditor neural net.

▶ **Question 1. Can we learn a $\mathcal{C}$-multicalibrated predictor by performing standard loss minimization over a family $\mathcal{C}'$ with greater complexity than $\mathcal{C}$?**

By standard loss minimization, we mean a procedure that aims to minimize a loss ℓ in expectation over a fixed class $\mathcal{C}'$ of predictors. The works of [23, 19, 16] point us to classes $\mathcal{C}'$ that do indeed contain $\mathcal{C}$-multicalibrated predictors, and they tell us how to find one such predictor using boosting iterations that also decrease the expected loss. However, they do not tell us that *directly* minimizing any particular loss will lead us to one such predictor: the loss minimizer might prefer other predictors within $\mathcal{C}'$ with lower expected loss, but which are not multicalibrated. Indeed we know examples where this tradeoff between loss minimization and multicalibration happens for some classes $\mathcal{C}'$ [22, 4]. Thus Question 1 asks if there is a more direct connection between loss minimization and multicalibration than what was previously known.

If we increase the power of $\mathcal{C}'$ to the extent that the family is expressive enough to contain the ground-truth predictor $f^*(x) := \mathbb{E}_\mathcal{D}[y|x]$ (the realizable case), for the squared loss and the cross-entropy loss, it is not hard to see that loss minimization over the family will bring our predictor f close to the ground truth f^*, and we will get multicalibration as a consequence (see Chapter 3 in [1] as well as [36]). However, this is not particularly insightful, since we do not expect this strong realizability assumption to hold in practice for predictor families used in common loss minimization algorithms.

For more reasonable choices of $\mathcal{C}'$ that do not guarantee realizability of the ground truth predictor, previous results reveal potential challenges to giving a positive answer to Question 1. For some simple choices of $\mathcal{C}'$, it is known that Question 1 has a negative answer since there is a tradeoff between loss minimization and multicalibration. Indeed, for the predictor family considered in logistic regression comprising sigmoids of linear functions, the predictor with minimum expected loss need not be even calibrated (see [4]). Even if $\mathcal{C}'$ contains predictor f_1 that is $\mathcal{C}$-multicalibrated, it might contain another predictor f_2 that is better at loss minimization.[2]

Another potential challenge comes from the work of [19], which showed that a multicalibrated predictor is an *omnipredictor*, i.e., it can be used to minimize *any* convex and Lipschitz loss function compared to a benchmark class of models defined based on $\mathcal{C}$. The omniprediction results seem to suggest that a positive answer to Question 1 without realizability is perhaps unlikely: a standard loss minimization procedure will find the best predictor $f \in \mathcal{C}'$ tailored to a particular loss ℓ, whereas in order to be $\mathcal{C}$-multicalibrated, f needs to be competitive with $\mathcal{C}$ for every Lipschitz, convex loss. It is tempting to believe that algorithms for achieving multicalibration might have to go beyond the usual framework of minimizing a single loss function over a family of predictors. Indeed, all previously known algorithms for multicalibration require boosting updates similar to the algorithm of [23].

To summarize, prior work tells us the following:

- There are examples of $\mathcal{C}'$ where loss minimization does not yield (multi)calibration. In these examples, $\mathcal{C}'$ is not too much more powerful than $\mathcal{C}$.
- Any positive answer to Question 1 requires $\mathcal{C}'$ to be sufficiently powerful relative to $\mathcal{C}$, so that optimizing over $\mathcal{C}'$ gives a $\mathcal{C}$-omnipredictor for all convex, Lipschitz losses.
- The answer to Question 1 is yes when $\mathcal{C}'$ is extremely (perhaps unreasonably) powerful, so that it contains the ground truth predictor.

This leaves open the possibility that a positive answer holds for $\mathcal{C}', \mathcal{C}$ pairs where $\mathcal{C}'$ is more expressive than $\mathcal{C}$, but loss minimization over $\mathcal{C}'$ is still tractable.

[2] Running the [23] boosting on f_2 will result in f_3 which is better at loss minimization than f_2 and is multicalibrated, but it may no longer belong to $\mathcal{C}'$.

1.2 Our Contribution: Multicalibration from Standard Loss Minimization

We show that multicalibration with respect to neural networks of size k can be achieved solely by minimizing the squared loss over the family of neural networks of size n, **for all but a few choices of** n. This result provides a positive answer to Question 1 without making any realizability assumptions. While we focus on the squared loss, our results extend to any proper loss such as the cross-entropy loss (see Appendix B).

Specifically, we take the auditor class $\mathcal{C}$ in Definition 1 to be NN_k^* which consists of all functions $c : \mathcal{X} \times [0,1] \to [-1,1]$ computable by some k-node neural network, where we assume the domain $\mathcal{X}$ is a subset of a Euclidean space $\mathbb{R}^d$. For concreteness, we use the ReLU activation function, which is a popular choice in practice (see Section 2 for formal definitions). Multicalibration w.r.t. $\mathcal{C} = \mathsf{NN}_k^*$ guarantees good calibration on *all* large enough groups identifiable by a size-k neural network. This is a strong guarantee when k is large enough to express interesting groups. We consider minimizing the squared loss over the family NN_n of all predictors $f : \mathcal{X} \to [0,1]$ computable by some n-node neural network. For a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, we say a predictor $f \in \mathsf{NN}_n$ is ε-loss-optimal if

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[(f(x)-y)^2] \le \inf_{f' \in \mathsf{NN}_n} \mathbb{E}_{(x,y)\sim\mathcal{D}}[(f'(x)-y)^2] + \varepsilon.$$

We prove the following theorem showing that loss optimality implies multicalibration for all but a few choices of n:

▶ **Theorem 2.** *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$. For every $k \in \mathbb{Z}_{>0}$ and every $\alpha > 0$, for all but at most $(k+2)/\alpha$ choices of $n \in \mathbb{Z}_{\ge 0}$, for any $\varepsilon \in (0,1)$, every ε-loss-optimal predictor $f \in \mathsf{NN}_n$ is $(\mathsf{NN}_k^*, \sqrt{\alpha+\varepsilon})$-MC.*

Note that the above theorem does not make any realizability assumptions. In particular, an ε-loss-optimal predictor $f \in \mathsf{NN}_n$ may have a significantly larger expected squared loss compared to f^*, and the theorem still guarantees that f is multicalibrated. Theorem 2 implies that if a neural network $f \in \mathsf{NN}_n$ violates multicalibration beyond a small threshold, its expected squared loss must be sub-optimal within the family NN_n, except for a few unlucky choices of n. Outside of these choices, a multicalibration violation indicates potential for further improvement of the expected loss *within the family* NN_n.

Our result can be viewed as a demonstration of the representation ability of neural networks, complementary to the universal representation theorems (e.g. [25]). While neural networks of a certain size n cannot express all functions (in particular, the ground truth f^* may be far from such neural networks), except for a few choices of n they can always express a multicalibrated predictor, and such a predictor can be found by minimizing the squared loss.

Our results are more about representational aspects of neural networks, and do not address algorithmic or sample complexity considerations. They currently do not apply to specific algorithms for optimizing neural networks, such as SGD, and they should not be interpreted as “fairness comes for free from optimizing neural networks”. The question of whether loss minimization over neural networks can be performed efficiently does not have a simple answer, SGD is found to do well in practice. The question of whether it results (for most n) in networks that are multicalibrated for smaller size neural networks is an interesting question from the theoretical and experimental viewpoint. See Section 5 for an extended discussion.

A Generalization

In our proof of Theorem 2, a key property we use about neural networks provides an explanation for their representation ability demonstrated by the theorem. The property is the simple fact that the composition of two neural networks can be implemented by another neural network with size roughly equal to the sum of the sizes of the two initial networks. Indeed, we generalize Theorem 2 to any sequence of families of predictors closed under composition:

▶ **Theorem 3.** *Let $\mathcal{C}$ be a class of auditing functions $c : \mathcal{X} \times [0,1] \to [-1,1]$. Let $\mathcal{F}_0, \mathcal{F}_1, \dots$ be families of predictors $f : \mathcal{X} \to [0,1]$ satisfying $\emptyset \neq \mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$. For some positive integer k, assume that for every nonnegative integer n, every $f \in \mathcal{F}_n$, every $c \in \mathcal{C}$, and every $\beta \in [-1,1]$, the composed predictor f' defined below satisfies $f' \in \mathcal{F}_{n+k}$:*

$$f'(x) = \mathrm{clip}(f(x) + \beta c(x, f(x))) \quad \text{for every } x \in \mathcal{X}, \tag{2}$$

where $\mathrm{clip}(z) = \max(0, \min(1, z)) \in [0,1]$ for every $z \in \mathbb{R}$. Then for every $\alpha > 0$, for all but at most k/α choices of $n \in \mathbb{Z}_{\geq 0}$, for any $\varepsilon > 0$, every ε-loss-optimal $f \in \mathcal{F}_n$ is $(\mathcal{C}, \sqrt{\alpha + \varepsilon})$-MC.

By definition, neural networks satisfy closeness under composition as required in Theorem 3, allowing us to prove Theorem 2 as a consequence of Theorem 3 (see Section 3 for details). In addition, Theorem 3 implies variants of Theorem 2 where we enforce various network architectures, though for simplicity we focus on a general feed-forward architecture in this paper.

Besides neural networks, we can apply Theorem 3 to other predictor families. For example, a well-studied family in computational learning theory is the family of *juntas*, which is also the family we consider in our lower bound results (see Section 1.3 below). Here, the domain $\mathcal{X}$ is the Boolean cube $\{-1,1\}^m$, and a function f over $\mathcal{X}$ is called a k-junta if $f(x)$ only depends on a fixed set of k coordinates of x for every $x \in \mathcal{X}$ (see Section 4 for formal definition). We use $\mathcal{J}_k$ to denote the class of all k-juntas $f : \mathcal{X} \to [0,1]$, and use $\mathcal{J}_k^*$ to denote the class of all k-juntas $f : \mathcal{X} \to [-1,1]$. We consider multiaccuracy (MA) (defined formally in Definition 12) which is a weaker notion than multicalibration where functions $c \in \mathcal{C}$ only takes x as input, instead of taking x and $f(x)$ as input as in Definition 1. Like neural networks, juntas also satisfy closeness under composition, so we get the following theorem as a corollary of Theorem 3 (see Section 4 for proof):

▶ **Theorem 4.** *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$, where $\mathcal{X} = \{-1,1\}^m$ for a positive integer m. Let $k \in \mathbb{Z}_{>0}$ and $\alpha \in (0,1]$ be parameters. Then for all but at most k/α nonnegative integers n, for any $\varepsilon > 0$, every ε-loss-optimal $f \in \mathcal{J}_n$ is $(\mathcal{J}_k^*, \sqrt{\alpha + \varepsilon})$-MA.*

A Smoothed Analysis Perspective

Since there are a few unlucky neural network sizes n for which Theorem 2 does not provide multicalibration guarantees, one may be worried that the size used in a task in practice might be among the unlucky ones. However, it is often reasonable to assume that there is sufficient randomness involved in a specific choice of n in practice, especially when n is large as in modern neural networks. If n is chosen, say, uniformly at random from a large range, the probability of n being among the few unlucky choices is small. This is the perspective taken in *smoothed analysis* [46]: by assuming that the instances arising in practice contain some intrinsic randomness, we can often show stronger guarantees for them than for the worst-case instance. We also show how to completely avoid unlucky sizes by adding a regularization term to the loss minimization problem (see Section 1.4).

Proof Sketch

Our proof of Theorem 3 is simple in hindsight, which we view as a plus. It combines the existing analysis of the boosting style updates of [23] with some new ideas, and exploits the structure of the updates.

For $\gamma = \sqrt{\alpha + \varepsilon}$, consider a predictor $f : \mathcal{X} \to [0,1]$ that is not γ-multicalibrated w.r.t. $\mathcal{C}$, i.e., there exists $c \in \mathcal{C}$ such that (1) is violated. A key step in previous boosting style algorithms is to use c to decrease the expected squared loss of f, by considering the predictor f' defined by (2). It can be shown that when $\beta \in [-1,1]$ is chosen properly, updating f to f' decreases the expected loss by more than $\gamma^2 = \alpha + \varepsilon$. In Theorem 3, if f belongs to some $\mathcal{F}_n$, then the updated predictor f' belongs to $\mathcal{F}_{n+k}$. Therefore, if $f \in \mathcal{F}_n$ is not γ-multicalibrated, f is not $(\alpha + \varepsilon)$-loss-optimal w.r.t. the *larger* class $\mathcal{F}_{n+k}$. To prove Theorem 3, we need to show that f is not ε-loss-optimal w.r.t. the *current* class $\mathcal{F}_n$.

Let OPT_n denote the minimum expected loss achievable by predictors from $\mathcal{F}_n$. If $\mathrm{OPT}_n \le \mathrm{OPT}_{n+k} + \alpha$, then $f \in \mathcal{F}_n$ being not $(\alpha + \varepsilon)$-loss-optimal w.r.t. $\mathcal{F}_{n+k}$ implies that it is not ε-loss-optimal w.r.t. $\mathcal{F}_n$, as desired. Thus we only need to worry about cases where OPT_n is larger than $\mathrm{OPT}_{n+k} + \alpha$. We observe that OPT_n is non-increasing in n and it is bounded in $[0,1]$, so there can only be a few choices of n for which OPT_n is larger than OPT_{n+k} by a significant amount α. Excluding these bad choices of n allows us to prove Theorem 3. A detailed proof is presented in Section 3.

1.3 Lower Bound

A natural question is whether the bound on the number of unlucky choices of n can be improved. While we are not able to show that the bound in Theorem 2 is optimal, we show tightness of the bound in Theorem 4 for juntas. We prove the following lower bound showing tightness up to constant: there are indeed $\Omega(k/\alpha)$ integers n for which the loss-optimal junta is not $\sqrt{\alpha}$-multiaccurate. The right quantitative bound uses the noise stability of the majority function (see [41]). Thus, a stronger result for neural networks than Theorem 2, if it exists, would use properties of neural networks beyond what our analysis currently uses.

▶ **Theorem 5** (Informal statement of Theorem 13). *For every k and small enough α, there exist $m \in \mathbb{Z}_{>0}$, a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\} = \{\pm 1\}^m \times \{0,1\}$, and at least $\Omega(k/\alpha)$ distinct non-negative integers n such that any model $f_n \in \mathcal{J}_n$ is* not *$(\mathcal{J}_k^*, \sqrt{\alpha})$-MA.*

1.4 Multicalibration from Structural Risk Minimization

There are a few unlucky choices of n for which Theorems 2 and 3 do not guarantee multicalibration from loss minimization. We show that by adding an appropriate regularization term to the loss minimization problem, we can avoid such unlucky choices of n. This regularized approach can be viewed as a form of structural risk minimization, and it can be motivated economically as follows.

Deploying large predictors is expensive. To deploy a finite predictor, one may consider a cost, say, a linear cost of for deploying a NN of n nodes. Once amortized over queries, this is equivalent to using regularization and choosing a predictor that minimizes loss plus a constant, say $1/N$, times size. Since the squared loss is bounded in $[0,1]$, the optimal solution is always an NN with at most N nodes. Our theorem below shows that large NNs optimized for loss, with sufficiently small regularization, achieve MC.

▶ **Theorem 6** (Informal version of Theorem 11)**.** *Let $k \in \mathbb{N}$ be an integer and $\alpha \in (0, 1]$. Then, for some $N_0 = O(k/\alpha)$ and any $N \geq N_0$, selecting a loss-optimal NN with size regularization coefficient $1/N$ will lead to a NN of size $\leq N$ that is $\sqrt{\alpha}$-MC with respect to NNs of size k.*

1.5 Related Work

The notion of multicalibration for multigroup fairness was introduced in [23], see also [31, 35]. This notion has proved to be unexpectedly rich, with connections to several other areas. The work of [12] relates it to indistinguishability, while [33] connects it to domain adaptation. A line of work on omniprediction [19, 17, 27, 34, 15] has shown that multicalibration for a class $\mathcal{C}$ implies strong loss minimization guarantees when compared against benchmarks from $\mathcal{C}$. This is the opposite direction to the one we study here: it shows settings where loss minimization results from multicalibration. A more recent work [20] in fact shows an equivalence between multicalibration and certain swap loss minimization problems. Without getting into details, the model of loss minimization is inspired by internal regret in online learning, and is different from the standard notion that we study here.

The relation between loss minimization and multicalibration has been investigated extensively in the literature over the last few years. In addition to what we have discussed earlier in the introduction, it is known that minimizing any proper loss for linear models over a base hypothesis class yields multiaccuracy w.r.t. the base class [17], but not multicalibration [22]. [32] showed that retraining the last layer of a DNN with cross-entropy guarantees multiaccuracy with respect to the features in the penultimate layer; this can be seen as a consequence of the result of [17] for linear models.

There is a long history studying calibration (not MC) which can be viewed as a special case of MC where the groups are defined by $f(x)$ only, i.e., $c(f(x))$, ignoring the features x itself. In particular, a set of practical studies that have found that large NNs are often calibrated "out of the box" despite being optimized solely for loss [37, 24, 30, 10, 8]. The work of [4] attempts to explain this phenomenon by proving that there is a tight connection between the *smooth calibration error* [29, 3] and the loss reduction that is obtainable by using Lipshcitz post-processing of the predicitons. The upshot is that models whose predictions cannot be improved by such post-processing have small calibration error. They speculate that this explains why some large NNs are calibrated out of the box.

The work of [32] showed that the performance of image classifiers on demographic subgroups can be improved by ensuring multiaccuracy. We are not aware of experimental work measuring the degree of multicalibration for large neural networks on massive training sets. There are numerous works on the representation ability of NNs (e.g. [26]) and about their accuracy and generalization (e.g. [48, 39]), too large to survey here.

1.6 Paper Organization

Section 2 defines the setting and mathematical preliminaries. Section 3 gives our main results, with the regularization analysis covered in Section 3.1. In Section 4 we show a matching lower bound for families of juntas. Finally, Section 5 discusses risks and limitations of the present work.

2 Preliminaries

We assume $\mathcal{X} \subseteq \mathbb{R}^d$ is a Euclidean space in some dimension d, and focus on binary outcomes (labels) in $\mathcal{Y} = \{0, 1\}$. We can also allow for $\mathcal{Y} = [0, 1]$, in which case the multicalibration guarantee we achieve is called mean multicalibration [28], or the multiclass setting with

$l > 2$ distinct labels. We focus on the binary outcome setting for simplicity. We assume an arbitrary, unknown joint distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$. For simplicity we focus on the expected squared loss of predictors $f : \mathcal{X} \to [0,1]$ defined as follows:

$$L(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\big(y - f(x)\big)^2\right].$$

Since only one distribution $\mathcal{D}$ is used throughout the analysis, we assume it is fixed and omit it from our notation, e.g., writing $L(f)$ rather than $L_{\mathcal{D}}(f)$ and omitting it from the expectation when clear from context.

We let NN_n denote the family of fully-connected Neural Networks with the standard ReLU activations and exactly n nodes, which map inputs from $\mathcal{X}$ to output values in $[0,1]$. The ReLU activation computes $\mathsf{ReLU}(z) := \max(0,z)$ on input z. Thus NN_n is defined by a Directed Acyclic Graph (DAG) with n nodes, where the activation of a node with d_{in} inputs is $a = \mathsf{ReLU}(w \cdot a_{\text{in}} + b)$ where $a_{\text{in}} \in \mathbb{R}^{d_{\text{in}}}$ is the vector of activations of its inputs, $w \in \mathbb{R}^{d_{\text{in}}}$ is a vector of equal dimension, and $b \in \mathbb{R}$ is a bias term. We also define $\mathsf{NN}_0 = \{0\}$, i.e., the constant 0 function is defined to be computed by a 0-node NN. The set NN_n is monotonically non-decreasing with n since we can trivially add identity nodes to compute the same function.

Enforcing an output in $[0,1]$ is added to the definition for convenience and can easily be achieved by wrapping the output z in two ReLUs:

$$\forall z \in \mathbb{R}, \quad \text{clip}(z) := \mathsf{ReLU}(z - \mathsf{ReLU}(z-1)) = \max(0, \min(1,z)) \in [0,1]. \tag{3}$$

Similarly to the definition of NN_n, we use NN^*_k to denote the family of fully-connected Neural Networks with the standard ReLU activations and exactly k nodes, which map inputs from $\mathcal{X} \times [0,1]$ to output values in $[-1,1]$. To get negative output values, we allow the output node to drop the ReLU transformation.

The results in this paper can be extended by enforcing various architectures, such as a NN with a given number of hidden layers, but we present the results for a general feed-forward architecture for simplicity.

3 Multicalibration from Loss Minimization

In this section, we prove Theorems 2 and 3 showing that for the family of neural networks of a given size as well as other families that satisfy closeness under composition, loss minimization over the family implies multicalibration.

We start with a key observation used in essentially all previous boosting-style algorithms for multicalibration:

▶ **Lemma 7** (Loss reduction from MC violation (see e.g. [23, 16, 4])). *Let $f : \mathcal{X} \to [0,1]$ be a predictor and $c : \mathcal{X} \times [0,1] \to [-1,1]$ be an auditor function. For a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, define $\beta \in [-1,1]$ by*

$$\beta := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[c\big(x, f(x)\big) \cdot \big(y - f(x)\big)\right].$$

Define a new predictor $h : \mathcal{X} \to [0,1]$ such that $h(x) := \text{clip}\big(f(x) + \beta c(x, f(x))\big)$ for every $x \in \mathcal{X}$. Then $L(h) \le L(f) - \beta^2$.

Proof. Define $g(x) := f(x) + \beta c(x, f(x))$, so $h(x) = \mathrm{clip}(g(x))$. Then,

$$\begin{aligned} L(g) &= \mathbb{E}\left[(y - f(x) - \beta c(x, f(x)))^2\right] \\ &= \mathbb{E}\left[(y - f(x))^2 - 2\beta c(x, f(x))(y - f(x)) + \beta^2 c^2(x, f(x))\right] \\ &= L(f) - 2\beta\, \mathbb{E}\left[c(x, f(x))(y - f(x))\right] + \beta^2\, \mathbb{E}\left[c^2(x, f(x))\right] \\ &= L(f) - 2\beta^2 + \beta^2\, \mathbb{E}\left[c^2(x, f(x))\right]. \end{aligned}$$

Since $c^2(x, f(x)) \le 1$, this implies that $L(g) \le L(f) - \beta^2$. Finally, note that for any (x, y), $(y - h(x))^2 \le (y - g(x))^2$ since $y \in \{0, 1\}$ and $h(x)$ is the projection of $g(x)$ onto the closest point in $[0, 1]$, thus $L(h) \le L(g)$. ◀

We prove Theorem 3 using Lemma 7:

Proof of Theorem 3. For each n, define OPT_n to be the infimum of $L(f)$ over $f \in \mathcal{F}_n$. We first show that for all but at most k/α choices of n,

$$\mathrm{OPT}_n \le \mathrm{OPT}_{n+k} + \alpha. \tag{4}$$

By our assumption, we have $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$, so OPT_n is non-increasing in n. It is also clear that $\mathrm{OPT}_n \in [0, 1]$. For $j \in \{1, \ldots, k\}$, suppose we can find m different nonnegative integers $n_1 < \cdots < n_m$ such that each n_i violates (4) (i.e., $\mathrm{OPT}_{n_i} > \mathrm{OPT}_{n_i+k} + \alpha$) and satisfies $n_i \equiv j \bmod k$. It is clear that $n_i + k \le n_{i+1}$, and thus $\mathrm{OPT}_{n_i+k} \ge \mathrm{OPT}_{n_{i+1}}$. Therefore,

$$1 \ge \mathrm{OPT}_{n_1} - \mathrm{OPT}_{n_m+k} \ge \sum_{i=1}^{m} (\mathrm{OPT}_{n_i} - \mathrm{OPT}_{n_i+k}) \ge m\alpha,$$

which implies $m \le 1/\alpha$. That is, there are at most $1/\alpha$ choices of n that violate (4) and satisfy $n \equiv j \bmod k$. Summing up over j, we know that there are at most k/α choices of n that violate (4).

It remains to show that for every n satisfying (4), for every $\varepsilon \in (0, 1)$, every ε-loss-optimal $f \in \mathcal{F}_n$ is $O(\sqrt{\alpha + \varepsilon})$-MC w.r.t. $\mathcal{C}$. For $c \in \mathcal{C}$, define $\beta := \big[c\big(x, f(x)\big) \cdot \big(y - f(x)\big)\big]$ and $h(x) := \mathrm{clip}\big(f(x) + \beta c(x, f(x))\big)$ as in Lemma 7. By our assumption, $h \in \mathcal{F}_{n+k}$. Therefore, by Lemma 7,

$$\beta^2 \le L(f) - L(h) \le L(f) - \mathrm{OPT}_{n+k} \le L(f) - \mathrm{OPT}_n + \alpha \le \varepsilon + \alpha.$$

This implies $|\beta| \le \sqrt{\alpha + \varepsilon}$. Since this holds for any $c \in \mathcal{C}$, the predictor f must be $(\mathcal{C}, \sqrt{\alpha + \varepsilon})$-MC. ◀

We prove Theorem 2 using Theorem 3 and the following basic property of neural networks:

▶ **Lemma 8** (Neural networks are closed under composition). *Let $f \in \mathsf{NN}_n$ and $c \in \mathsf{NN}^*_k$. Define predictor $h : \mathcal{X} \to [0, 1]$ such that $h(x) := \mathrm{clip}\big(f(x) + \beta c(x, f(x))\big)$ for every $x \in \mathcal{X}$. Then $h \in \mathsf{NN}_{n+k+2}$.*

Proof. Observe that $h(x) = \mathrm{clip}\big(f(x) + \beta c(x, f(x))\big)$ is computed by

$$h(x) = \mathsf{ReLU}\big(f(x) + \beta c(x, f(x)) - \mathsf{ReLU}(f(x) + \beta c(x, f(x)) - 1)\big). \tag{5}$$

as can be seen through Eq. (3). Then observe that Eq. (5) is indeed a representation of a $(n + k + 2)$-node NN: n nodes to compute $f(x)$, k nodes to compute $c(x, f(x))$ and then the 2 additional ReLU nodes as described (note that $f(x)$ and $c(x, f(x))$ are both re-used without recomputing them).[3] Thus $h \in \mathsf{NN}_{n+k+2}$ as claimed. ◀

Proof of Theorem 2. The theorem follows immediately by combining Theorem 3 and Lemma 8. ◀

[3] We allow the output node of the k-node network computing $c(x, f(x))$ to drop the ReLU transformation

3.1 Regularization

We show that the regularized approach discussed in Section 1.4 allows us to avoid the unlucky sizes n. Specifically, we prove the following theorem about general predictor families in the setting of Theorem 3 and then specialize it to neural networks in Theorem 11.

▶ **Theorem 9.** *In the setting of Theorem 3, fix $\alpha > 0$ and consider $n \in \mathbb{Z}_{\geq 0}$ and $f \in \mathcal{F}_n$ that minimize $L(f) + \alpha n/k$ up to error ε, where $\alpha n/k$ is a regularization term that depends on n. That is,*

$$L(f) + \alpha n/k \leq \inf_{n' \in \mathbb{Z}_{\geq 0}, f' \in \mathcal{F}_{n'}} L(f') + \alpha n'/k + \varepsilon. \tag{6}$$

Then f is $(\mathcal{C}, \sqrt{\alpha + \varepsilon})$-MC.

▶ Remark 10. We can always choose $n \leq k/\alpha$ in (6). If $n > k/\alpha$, we can replace n by $\tilde{n} = 0$ and replace f by an arbitrary $\tilde{f} \in \mathcal{F}_0$, and get an improvement: $L(\tilde{f}) + \alpha\tilde{n}/k \leq 1 < \alpha n/k \leq L(f) + \alpha n/k$.

Proof of Theorem 9. Fixing $n' = n + k$ in (6), we have

$$L(f) + \alpha n/k \leq \inf_{f' \in \mathcal{F}_{n+k}} L(f') + \alpha(n+k)/k + \varepsilon = \mathrm{OPT}_{n+k} + \alpha(n+k)/k + \varepsilon.$$

This implies that $L(f) \leq \mathrm{OPT}_{n+k} + \alpha + \varepsilon$. The rest of the proof is identical to the proof of Theorem 3. Specifically, for $c \in \mathcal{C}$, defining β and h as in Lemma 7, we get $\beta^2 \leq L(f) - L(h) \leq L(f) - \mathrm{OPT}_{n+k} \leq \alpha + \varepsilon$, which implies $|\beta| \leq \sqrt{\alpha + \varepsilon}$, as desired. ◀

Combining Theorem 9 and Lemma 8, we get the following theorem about neural networks:

▶ **Theorem 11** (Size-regularized NNs). *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$. For every $k \in \mathbb{Z}_{>0}$ and every $\alpha > 0$, consider $n \in \mathbb{Z}_{\geq 0}$ and $f \in \mathsf{NN}_n$ that minimize $L(f) + \alpha n/(k+2)$ up to error ε. That is,*

$$L(f) + \alpha n/(k+2) \leq \inf_{n' \in \mathbb{Z}_{\geq 0}, f' \in \mathsf{NN}_{n'}} L(f') + \alpha n'/(k+2) + \varepsilon.$$

Then f is $(\mathsf{NN}_k^*, \sqrt{\alpha + \varepsilon})$-MC.

By Remark 10, the theorem above gives a way of finding an $n \leq (k+2)/\alpha$-node NN that is $\sqrt{\alpha + \varepsilon}$-MC with respect to k-node NNs.

Finite training sets

If one has a training set consisting of m examples drawn i.i.d. from $\mathcal{D}$, a natural approach is to minimize training loss. If m is sufficiently large compared to the number of NN parameters, which is polynomial in n, generalization bounds guarantee every $f \in \mathsf{NN}_n$ will have training loss close to its true loss $L(f)$, and thus any $\hat{f}_n$ which minimizes training error (still ignoring

because $c(x, f(x)) \in [-1, 1]$ could be a negative value which cannot be the output of a ReLU transformation. Thus the $(n + k + 2)$-node network we construct to compute h may contain an internal node p that does not apply the ReLU transformation. This can be easily fixed by noting that node p computes its output a by $a = w \cdot a_{\text{in}} + b$ where a_{in} is a vector consisting of outputs of previous nodes. We can remove node p from the $(n + k + 2)$-node network without changing the final output because any node taking a as input can alternatively take a_{in} as input and reconstruct a using the affine transformation $w \cdot a_{\text{in}} + b$.

computation time) will have, say, $L(\hat{f}_n) \leq \min_{\mathsf{NN}_n} L(f) + \varepsilon$ for $\varepsilon = O(\alpha)$. This can, in turn, be used with the above theorems to show that, outside a set of $O(k/\alpha)$ NN sizes, all NNs that are optimal on the training set will be $O(\sqrt{\alpha})$-MC with respect to k-node NNs. While such an analysis would be straightforward, we find it unsatisfactory given the fact that current generalization bounds, though correct, seem to be too loose to capture the performance of many NN learning algorithms in practice (e.g. [48, 39]). However, our results above may be relevant in settings where generalization is guaranteed – for example, when models are trained using Stochastic Gradient Descent (SGD) for only one-epoch (one pass through the training data). In this case, the learning algorithm can be thought of as optimizing via SGD directly on the population loss, without considering a finite train set. If we heuristically believe that, when run for long enough, the optimization reaches close to a population loss minima within the architecture family, then our results imply most resulting minima will also be multicalibrated. Moreover, the one-epoch setting may not be far from reality, since most modern large language models are indeed trained for only one epoch [6, 2], and there is evidence that generalization in the multi-epoch setting can be understood via the one-epoch setting [40].

4 A Lower Bound

We would like to understand the tightness of the analysis we have presented. Unfortunately, this is challenging due to the complex structure of NNs. Instead, this section provides some evidence of the tightness of our analysis at least *using our current methods*. To do this, we consider another natural class of functions namely *k-juntas* and a weaker notion of multigroup fairness called multiaccuracy (MA), to which our analysis applies. In this setting, we show a sharp result, proving that our bounds are tight up to constant factors.

The class of *juntas*, functions that depend on a small subset of inputs, has been well studied in computational learning theory [38, 14]. In this section, we show that juntas satisfy a similar property to NNs in that, for most sizes, minimizing loss for juntas also implies MA with respect to smaller juntas. We also prove a *lower bound* for showing that our bounds are tight up to constant factors.

For our results in this section, we choose the domain $\mathcal{X}$ to be the Boolean cube $\{-1,1\}^m$ for a dimension $m \in \mathbb{Z}_{>0}$. That is, every $x \in \mathcal{X}$ can be written as $x = (x_1, \dots, x_m)$ where $x_i \in \{-1,1\}$ for every $i = 1, \dots, m$. For a positive integer k and a function $f : \mathcal{X} \to \mathbb{R}$, we say f is a *k-junta* if there exist $i_1, \dots, i_k \in \{1, \dots, m\}$ and $g : \{-1,1\}^k \to \mathbb{R}$ such that $f(x) = g(x_{i_1}, \dots, x_{i_k})$ for every $x \in \mathcal{X}$. We say a function $f : \mathcal{X} \to \mathbb{R}$ is a 0-junta if f is a constant function. We use $\mathcal{J}_k$ (resp. $\mathcal{J}_k^*$) to denote the family of k-juntas that map inputs in $\mathcal{X} = \{-1,1\}^m$ to output values in $[0,1]$ (resp. $[-1,1]$).

We consider multiaccuracy, which is a weaker notion than multicalibration, where the auditor functions c do not have access to the predictions $f(x)$:

▶ **Definition 12** (Multiaccuracy [23]). *Let $\gamma > 0$ and $\mathcal{C}$ be a class of auditor functions $c : \mathcal{X} \to [-1,1]$. The predictor $f : \mathcal{X} \to [0,1]$ is $(\mathcal{C}, \gamma)$-multiaccurate or $(\mathcal{C}, \gamma)$-MA if for all $c \in \mathcal{C}$,*

$$\left| \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[c(x) \cdot (y - f(x)) \right] \right| \leq \gamma.$$

We prove the following theorem for juntas which is similar to our result about neural networks in Theorem 2 (restating the theorem from the introduction):

▶ **Theorem 4.** *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$, where $\mathcal{X} = \{-1,1\}^m$ for a positive integer m. Let $k \in \mathbb{Z}_{>0}$ and $\alpha \in (0,1]$ be parameters. Then for all but at most k/α nonnegative integers n, for any $\varepsilon > 0$, every ε-loss-optimal $f \in \mathcal{J}_n$ is $(\mathcal{J}_k^*, \sqrt{\alpha+\varepsilon})$-MA.*

The proof works exactly like the proof of Theorem 2 using the following observation. For $f \in \mathcal{J}_n, c \in \mathcal{J}_k^*$, and $\beta \in \mathbb{R}$, define function $h : \mathcal{X} \to [0,1]$ as

$$h(x) = \operatorname{clip}(f(x) + \beta c(x)),$$

then $h \in \mathcal{J}_{n+k}$. The following theorem gives a lower bound that matches Theorem 4 up to a constant factor.

▶ **Theorem 13.** *For every $k \in \mathbb{Z}_{>0}$ and every $\alpha \in (0, 1/(4\pi^2)]$, there exist $m \in \mathbb{Z}_{>0}$, a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} = \{-1,1\}^m$ and $\mathcal{Y} = \{0,1\}$, and at least $k/(6\pi^2\alpha)$ distinct nonnegative integers n such that any model $f_n \in \mathcal{J}_n$ is* not *$(\mathcal{J}_k^*, \sqrt{\alpha})$-MA.*

Key to our proof of Theorem 13 is the *majority* function. For an odd positive integer k, we define the majority function $\mathsf{MAJ} : \{-1,1\}^k \to \{-1,1\}$ such that for every $(x_1, \dots, x_k) \in \{-1,1\}^k$, $\mathsf{MAJ}(x_1, \dots, x_k) = 1$ if $x_1 + \dots + x_k > 0$, and $\mathsf{MAJ}(x_1, \dots, x_k) = -1$ otherwise. The following lemma about majority functions follows from standard noise sensitivity bounds (see [41]). We provide a proof in Appendix A for completeness.

▶ **Lemma 14.** *For any odd positive integers m and k satisfying $k \le m$,*

$$\mathbb{E}[\mathsf{MAJ}(x_1, \dots, x_k)\mathsf{MAJ}(x_1, \dots, x_m)] > \frac{2}{\pi} \cdot \sqrt{\frac{k}{m}},$$

where the expectation is over $(x_1, \dots, x_m)$ drawn uniformly at random from $\{-1,1\}^m$.[4]

Using the majority function, we define a distribution $\mathcal{D}$ over $\{-1,1\}^m \times \{0,1\}$ for any odd positive integer m as follows. We first draw $x \in \{-1,1\}^m$ uniformly at random and then set

$$y = \frac{1}{2}(1 + \mathsf{MAJ}(x)) \in \{0,1\}.$$

We define $\mathcal{D}$ to be the distribution of (x,y). Lemma 14 allows us to prove the following lemma about the distribution $\mathcal{D}$, which we then use to prove Theorem 13.

▶ **Lemma 15.** *Let k, m be odd positive integers satisfying $k \le m$. Define $\mathcal{X} := \{-1,1\}^m$ and define distribution $\mathcal{D}$ as above. Let $\alpha > 0$ be a parameter satisfying*

$$\alpha \le \frac{k}{\pi^2 m}. \tag{7}$$

Then any function $f \in \mathcal{J}_{m-k}$ is not *$(\mathcal{C}, \sqrt{\alpha})$-MA with respect to $\mathcal{C} = \mathcal{J}_k^*$.*

Proof. It suffices to show that for any function $f \in \mathcal{J}_{m-k}$, there exist $i_1, \dots, i_k \in \{1, \dots, m\}$ such that

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[(y - f(x))\mathsf{MAJ}(x_{i_1}, \dots, x_{i_k})] > \frac{1}{\pi} \cdot \sqrt{\frac{k}{m}}. \tag{8}$$

[4] The constant $2/\pi$ cannot be improved because in the limit where $k, m \to \infty$, $\sqrt{k/m} \to \rho \in (0,1)$, using the central limit theorem one can show that $\mathbb{E}[\mathsf{MAJ}(x_1, \dots, x_k)\mathsf{MAJ}(x_1, \dots, x_m)] \to (2/\pi) \arcsin \rho$. Also, it is easy to show that $\mathbb{E}[f(x_1, \dots, x_k)\mathsf{MAJ}(x_1, \dots, x_m)]$ is maximized when $f = \mathsf{MAJ}$ among all $f : \{-1,1\}^k \to [-1,1]$ (see e.g. Lemma 13 in [18]).

By the definition of $f \in \mathcal{J}_{m-k}(\mathcal{X})$, there exist $j_1, \ldots, j_{m-k} \in \{1, \ldots, m\}$ and $g : \{-1, 1\}^{m-k}$ such that

$$f(x) = g(x_{j_1}, \ldots, g_{j_{m-k}}) \quad \text{for every } x \in \mathcal{X}.$$

Now we choose distinct $i_1, \ldots, i_k \in \{1, \ldots, m\} \setminus \{j_1, \ldots, j_{m-k}\}$. We have

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[f(x)\mathsf{MAJ}(x_{i_1}, \ldots, x_{i_k})] = \mathbb{E}[g(x_{j_1}, \ldots, g_{j_{m-k}})\mathsf{MAJ}(x_{i_1}, \ldots, x_{i_k})] = 0, \tag{9}$$

where the last equation holds because $(x_{i_1}, \ldots, x_{i_k})$ is independent from $(x_{j_1}, \ldots, x_{j_{m-k}})$. By Lemma 14 and our choice of distribution $\mathcal{D}$,

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[y\,\mathsf{MAJ}(x_{i_1}, \ldots, x_{i_k})] > \frac{1}{\pi} \cdot \sqrt{\frac{k}{m}}. \tag{10}$$

Combining (9) and (10) proves (8). ◀

We are now ready to prove our lower bound Theorem 13.

Proof of Theorem 13. For any positive integer k, define k_1 to be the largest odd integer that does not exceed k. It is easy to verify that $k_1 \geq k/2 > 0$. For any $\alpha \in (0, 1/(4\pi^2)]$, choose m to be the largest odd integer smaller than $k_1/(\pi^2\alpha)$. Our assumption that $\alpha \leq 1/(4\pi^2)$ ensures that $k_1/(\pi^2\alpha) \geq 4k_1$, and thus $m \geq 3k_1$ and $m \geq k_1/(2\pi^2\alpha)$. Moreover, our choice of m ensures that

$$\alpha \leq \frac{k_1}{\pi^2 m}.$$

By Lemma 15, for $\mathcal{X} = \{-1, 1\}^m$ and $\mathcal{Y} = \{0, 1\}$ there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ such that for every nonnegative integer $n = 0, \ldots, m - k_1$, any model $f_n \in \mathcal{J}_n$ is *not* $(\mathcal{C}, \sqrt{\alpha})$-MA with respect to $\mathcal{C} = \mathcal{J}^*_{k_1}$. Since $\mathcal{J}^*_{k_1} \subseteq \mathcal{J}^*_k$, any $f_n \in \mathcal{J}_n$ is also *not* $(\mathcal{C}, \sqrt{\alpha})$-MA with respect to $\mathcal{C} = \mathcal{J}^*_k$. The number of such integers n is

$$m - k_1 + 1 \geq 2m/3 \geq (2/3)(k_1/(2\pi^2\alpha)) = k_1/(3\pi^2\alpha) \geq k/(6\pi^2\alpha). \quad ◀$$

5 Discussion

We start by discussing some limitations of our work. We do not provide any guidance on selecting k, which is an important theoretical and practical issue. Second, there is a blowup in the NN size n required to achieve MC with respect to NNs of size k. Also, the model we study is highly unrealistic in that we assume that we have unbounded training data and computational resources at training time to achieve near-minimum expected loss over neural networks of a given size. Although current pre-training efforts do involve massive amounts of data and compute, it is not clear that they are or will ever approach what could be done in this limit.

Nonetheless, we feel that it still may be useful to understand the nature of different predictor representations and the tensions that they face. First of all, one conclusion of this work is that one can achieve MC with NNs–as a representation they are not limited in the same way as some other models. In particular, simpler models (which may enjoy other benefits such as interpretability) may face an inherent tension between accuracy and calibration overall and among different groups.

If one leads a group of practitioners training NNs and one is concerned about multicalibration, our work suggests that incentivizing them to simply minimize loss may, to some extent, be aligned with MC. This is formal in the sense of Lemma 7: if one can identify a group for which the current predictor is miscalibrated, then one can further reduce the loss, escaping a local minimum if the predictor was trained with a local optimization algorithm. But this may naturally happen if they were incentivized to minimize loss. If it is not happening naturally, such a "check" could be suggested and it would be entirely compatible with the group's loss incentives.

Future work

It would be interesting to empirically measure whether NNs used in practice are multicalibrated. If so, this might be viewed as one additional benefit of their use over simpler models. If not, it would be interesting to understand the reasons– the gaps between theory and practice.

Another direction would be to prove a similar result for other classes of hypotheses. The key property of neural nets that we use is that the post-processing required to update the predictor can be incorporated into a predictor of larger size. It is not hard to see that other classes such as decision trees and branching programs (parametrized by size) do have this property. But they are not known to have provable algorithms for squared loss minimization. In contrast linear models do admit efficient loss minimization. But they are not powerful enough to be closed under the kinds of post-processing that multicalibration requires. Moreover, they cannot express the kind of predicates that are interesting for multicalibration, such as $x \in S$ and $f(x) = 0.1$.

An intriguing question is whether there is a hypothesis class that is indeed powerful enough to give interesting multicalibration guarantees and is closed under post-processing, but which admits efficient loss minimization. It is also possible that there exists a no-free-lunch flavor of result which rules out such a possibility. We leave formalizing and proving such results to future work.

Risks

We conclude on a cautionary note. There is a risk that this and other similar work will be misinterpreted as "fairness/multicalibration comes for free if you just optimize loss in DNNs" but that is not a valid conclusion for multiple reasons. First of all, multicalibration is only one notion of fairness. It is particularly applicable in settings where and the groups of interest are identifiable from x. Multicalibration cannot protect groups of concern that are not identifiable from x, e.g., race is a complex feature that may not be explicitly coded or perfectly computable from existing features, and the protection gets weaker for smaller groups.

If a practitioner was given an *explicit* set of protected groups whose accuracy is to be prioritized, they ought to consider all steps in the machine learning pipeline and not just optimization: they might adjust their architectures, collect additional data, or expand the feature set so as to increase the family of protected subgroups and performance on those groups. For instance, explicitly investigating multicalibration may lead to discovering certain groups on which it is hard to achieve good calibration, which requires a different type of learning architecture, better datasets or more informative features.

References

1 Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019. URL: http://www.fairmlbook.org.

2 Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint*, 2023. arXiv:2304.01373.

3 Jarosław Błasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. A unifying theory of distance from calibration. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1727–1740, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585182.

4 Jarosław Błasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. When does optimizing a proper loss yield calibration? *arXiv preprint*, 2023. arXiv:2305.18764.

5 Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning with Fenchel-Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020. URL: http://jmlr.org/papers/v21/19-021.html.

6 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, ..., and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

7 Andreas Buja, Werner Stuetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications. *Manuscript*, 2005. URL: https://sites.stat.washington.edu/wxs/Learning-papers/paper-proper-scoring.pdf.

8 Annabelle Carrell, Neil Mallinar, James Lucas, and Preetum Nakkiran. The calibration generalization gap. *arXiv preprint*, 2022. arXiv:2210.01964.

9 Zhun Deng, Cynthia Dwork, and Linjun Zhang. Happymap : A generalized multicalibration method. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 41:1–41:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.41.

10 Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, 2020.

11 John Duchi, Khashayar Khosravi, and Feng Ruan. Multiclass classification, information, divergence and surrogate risk. *The Annals of Statistics*, 46(6B):3246–3275, 2018. doi:10.1214/17-AOS1657.

12 Cynthia Dwork, Michael P. Kim, Omer Reingold, Guy N. Rothblum, and Gal Yona. Outcome indistinguishability. In *ACM Symposium on Theory of Computing (STOC'21)*, 2021. arXiv:2011.13426.

13 Cynthia Dwork, Michael P. Kim, Omer Reingold, Guy N. Rothblum, and Gal Yona. Beyond bernoulli: Generating random outcomes that cannot be distinguished from nature. In *The 33rd International Conference on Algorithmic Learning Theory*, 2022.

14 Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753–787, 2004. Special Issue on FOCS 2002. URL: https://www.sciencedirect.com/science/article/pii/S0022000003001831.

15 Sumegha Garg, Christopher Jung, Omer Reingold, and Aaron Roth. Oracle efficient online multicalibration and omniprediction. *CoRR*, abs/2307.08999, 2023. doi:10.48550/arXiv.2307.08999.

16 Ira Globus-Harris, Declan Harrison, Michael Kearns, Aaron Roth, and Jessica Sorrell. Multicalibration as boosting for regression. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 11459–11492. PMLR, 23–29 July 2023. URL: `https://proceedings.mlr.press/v202/globus-harris23a.html`.

17 Parikshit Gopalan, Lunjia Hu, Michael P. Kim, Omer Reingold, and Udi Wieder. Loss Minimization Through the Lens Of Outcome Indistinguishability. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2023.60`.

18 Parikshit Gopalan, Adam Tauman Kalai, and Adam R. Klivans. Agnostically learning decision trees. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 527–536, 2008.

19 Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *Innovations in Theoretical Computer Science (ITCS'2022)*, 2022. `arXiv:2109.05389`.

20 Parikshit Gopalan, Michael Kim, and Omer Reingold. Characterizing notions of omniprediction via multicalibration. In *under submission*, 2023. `arXiv:2302.06726`.

21 Parikshit Gopalan, Michael P. Kim, Mihir Singhal, and Shengjia Zhao. Low-degree multicalibration. In *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 3193–3234. PMLR, 2022.

22 Parikshit Gopalan, Omer Reingold, Vatsal Sharan, and Udi Wieder. Multicalibrated partitions for importance weights. In *International Conference on Algorithmic Learning Theory, 29-1 April 2022, Paris, France*, volume 167 of *Proceedings of Machine Learning Research*, pages 408–435. PMLR, 2022.

23 Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018.

24 Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, 2020. URL: `https://openreview.net/forum?id=S1gmrxHFvB`.

25 Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. `doi:10.1016/0893-6080(89)90020-8`.

26 Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

27 Lunjia Hu, Inbal Rachel Livni Navon, Omer Reingold, and Chutong Yang. Omnipredictors for constrained optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 13497–13527. PMLR, 23–29 July 2023. URL: `https://proceedings.mlr.press/v202/hu23b.html`.

28 Christopher Jung, Changhwa Lee, Mallesh M Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation. *arXiv preprint*, 2020. `arXiv:2008.08037`.

29 Sham M. Kakade and Dean P. Foster. Deterministic calibration and Nash equilibrium. In John Shawe-Taylor and Yoram Singer, editors, *Learning Theory*, pages 33–48, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

30 Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael C Mozer, and Becca Roelofs. Soft calibration objectives for neural networks. *Advances in Neural Information Processing Systems*, 34:29768–29779, 2021.

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

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

33 Michael P Kim, Christoph Kern, Shafi Goldwasser, Frauke Kreuter, and Omer Reingold. Universal adaptability: Target-independent inference that competes with propensity scoring. *Proceedings of the National Academy of Sciences*, 119(4), 2022.

34 Michael P. Kim and Juan C. Perdomo. Making Decisions Under Outcome Performativity. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 79:1–79:15, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2023.79`.

35 Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference, ITCS*, 2017.

36 Lydia T Liu, Max Simchowitz, and Moritz Hardt. The implicit fairness criterion of unconstrained learning. In *International Conference on Machine Learning*, pages 4051–4060. PMLR, 2019.

37 Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694, 2021.

38 Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. Learning functions of k relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004. Special Issue on STOC 2003. `doi:10.1016/j.jcss.2004.04.002`.

39 Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

40 Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. In *International Conference on Learning Representations*, 2021. URL: `https://openreview.net/forum?id=guetrIHLFGI`.

41 Ryan O'Donnell. *Analysis of boolean functions.* Cambridge University Press, 2014.

42 Mark D. Reid and Robert C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11(83):2387–2422, 2010. URL: `http://jmlr.org/papers/v11/reid10a.html`.

43 Leonard J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971. `doi:10.1080/01621459.1971.10482346`.

44 Mark J. Schervish. A general method for comparing probability assessors. *The Annals of Statistics*, 17(4):1856–1879, 1989. `doi:10.1214/aos/1176347398`.

45 Emir H. Shuford, Arthur Albert, and H. Edward Massengill. Admissible probability measurement procedures. *Psychometrika*, 31(2):125–145, 1966. `doi:10.1007/BF02289503`.

46 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, May 2004. `doi:10.1145/990308.990310`.

47 Robert C Titsworth. *Correlation properties of cyclic sequences.* PhD thesis, California Institute of Technology, 1962.

48 Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

A Correlation between Majority Functions

We restate and prove Lemma 14.

▶ **Lemma 14.** *For any odd positive integers m and k satisfying $k \le m$,*

$$\mathbb{E}[\mathsf{MAJ}(x_1,\dots,x_k)\mathsf{MAJ}(x_1,\dots,x_m)] > \frac{2}{\pi}\cdot\sqrt{\frac{k}{m}},$$

where the expectation is over $(x_1,\dots,x_m)$ drawn uniformly at random from $\{-1,1\}^m$.[5]

Proof. For any $S \subseteq \{1,\dots,m\}$, we define

$$\widehat{\mathsf{MAJ}}_k(S) := \mathbb{E}[\mathsf{MAJ}(x_1,\dots,x_k)\prod_{i\in S} x_i].$$

By the Plancherel theorem,

$$\mathbb{E}[\mathsf{MAJ}(x_1,\dots,x_k)\mathsf{MAJ}(x_1,\dots,x_m)] = \sum_{S\subseteq\{1,\dots,m\}} \widehat{\mathsf{MAJ}}_k(S)\widehat{\mathsf{MAJ}}_m(S).$$

It is easy to verify that if S is not a subset of $\{1,\dots,k\}$, then $\widehat{\mathsf{MAJ}}_k(S) = 0$. Also, by calculating the Fourier coefficients of the majority function [47] (see Section 5.3 in [41]), we have $\widehat{\mathsf{MAJ}}_k(S)\widehat{\mathsf{MAJ}}_m(S) \ge 0$ for any $S \subseteq \{1,\dots,k\}$. Therefore,

$$\begin{aligned}\mathbb{E}[\mathsf{MAJ}(x_1,\dots,x_k)\mathsf{MAJ}(x_1,\dots,x_m)] &\ge \sum_{i=1}^{k}\widehat{\mathsf{MAJ}}_k(\{i\})\widehat{\mathsf{MAJ}}_m(\{i\})\\ &= k\left(\tbinom{k-1}{(k-1)/2}/2^{k-1}\right)\left(\tbinom{m-1}{(m-1)/2}/2^{m-1}\right)\\ &> k\sqrt{\frac{2}{\pi k}}\sqrt{\frac{2}{\pi m}}\\ &= \frac{2}{\pi}\sqrt{\frac{k}{m}}.\end{aligned}$$

Here we use the fact that for any odd positive integer k,

$$\tbinom{k-1}{(k-1)/2}/2^{k-1} = \frac{1\times 3\times\cdots\times(k-2)}{2\times 4\times\cdots\times(k-1)} = \sqrt{\frac{1}{k}\cdot\left(\frac{1}{2}\cdot\frac{3}{2}\cdot\frac{3}{4}\cdot\frac{5}{4}\cdot\cdots\cdot\frac{k-2}{k-1}\cdot\frac{k}{k-1}\right)} > \sqrt{\frac{1}{k}\cdot\frac{2}{\pi}},$$

where the last inequality follows from the Wallis product formula for π. ◀

B Generalization to Any Proper Loss

In our main results (Theorems 2, 3, 4, 9, 11), we show that multicalibration can be achieved by loss minimization over many predictor families. While we present these results using the squared loss, in this section we show that they can be generalized to any *proper* loss function satisfying basic regularity conditions, including the cross-entropy loss used widely

[5] The constant $2/\pi$ cannot be improved because in the limit where $k, m \to \infty$, $\sqrt{k/m} \to \rho \in (0,1)$, using the central limit theorem one can show that $\mathbb{E}[\mathsf{MAJ}(x_1,\dots,x_k)\mathsf{MAJ}(x_1,\dots,x_m)] \to (2/\pi)\arcsin\rho$. Also, it is easy to show that $\mathbb{E}[f(x_1,\dots,x_k)\mathsf{MAJ}(x_1,\dots,x_m)]$ is maximized when $f = \mathsf{MAJ}$ among all $f : \{-1,1\}^k \to [-1,1]$ (see e.g. Lemma 13 in [18]).

in neural network training. The main technical tool we use to establish this generalization is Theorem 19 below from the work of [4]. Theorem 19 generalizes the loss reduction lemma (Lemma 7) to general proper loss functions.

We first formally define proper loss functions. A loss function ℓ takes as input a binary outcome $y \in \{0,1\}$ and a prediction value $v \in [0,1]$, and outputs a real number $\ell(y,v) \in \mathbb{R}$. A proper loss function is defined as follows:

▶ **Definition 16** (Proper loss). *Let $V \subseteq [0,1]$ be a non-empty interval. We say a loss function $\ell : \{0,1\} \times V \to \mathbb{R}$ is* proper *if for every $v \in V$, it holds that $v \in \arg\min_{v' \in V} \mathbb{E}_{y \sim \mathsf{Ber}(v)}[\ell(y,v')]$.*

One can easily verify that the squared loss $\ell_{\mathsf{sq}}(y,v) = (y-v)^2$ is a proper loss function over $V = [0,1]$, and the cross-entropy loss $\ell_{\mathsf{xent}}(y,v) = -y \ln v - (1-y)\ln(1-v)$ is a proper loss function over $V = (0,1)$.

Given input $x \in \mathcal{X}$, a neural network trained to minimize the expected cross-entropy loss usually first computes a *logit* $t \in \mathbb{R}$, from which the final prediction $v \in (0,1)$ is obtained through the sigmoid transformation σ by $v = \sigma(t) := 1/(1+e^{-t})$. Given this relationship between v and t, the cross-entropy loss on v is the same as the *logistic loss* ℓ_{logistic} on t:

$$\ell_{\mathsf{xent}}(y,v) = \ell_{\mathsf{logistic}}(y,t) := \ln(1+e^t) - yt. \tag{11}$$

Therefore, instead of considering the cross-entropy loss of the final prediction v, we can equivalently consider the logistic loss of the logit t. This allows us to avoid complication caused by the sigmoid transformation and focus just on the ReLU network producing the logit.

The following lemma generalizes this correspondence between the cross-entropy loss and the logistic loss to any proper loss:

▶ **Lemma 17** (see [4]). *Let $V \subseteq [0,1]$ be a non-empty interval. Let $\ell : \{0,1\} \times V \to \mathbb{R}$ be a proper loss function. For every $v \in V$, define $\mathsf{dual}(v) := \ell(0,v) - \ell(1,v)$. Then there exists a convex function $\psi : \mathbb{R} \to \mathbb{R}$ such that*

$$\ell(y,v) = \psi(\mathsf{dual}(v)) - y\,\mathsf{dual}(v) \quad \textit{for every } y \in \{0,1\} \textit{ and } v \in V. \tag{12}$$

We can additionally ensure that $\frac{\psi(t_1)-\psi(t_2)}{t_1-t_2} \in [0,1]$ for any distinct $t_1, t_2 \in \mathbb{R}$. If ψ is differentiable, then $\nabla\psi(t) \in [0,1]$ for every $t \in \mathbb{R}$, and $\nabla\psi(\mathsf{dual}(v)) = v$ for every $v \in V$, where $\nabla\psi$ denotes the derivative of ψ.

A proof of Lemma 17 can be found in the work of [4]. It is based on a connection between proper loss functions and conjugate pairs of convex functions studied by [45, 43, 44, 7].

In Lemma 17, we say $\mathsf{dual}(v)$ is the *dual prediction* corresponding to a prediction value $v \in V$. When ℓ is the cross-entropy loss ℓ_{xent}, the dual prediction $\mathsf{dual}(v)$ is exactly the logit t corresponding to the prediction v. For any proper loss ℓ, Lemma 17 identifies a corresponding convex function $\psi : \mathbb{R} \to \mathbb{R}$. We define a *dual loss function* $\ell^{(\psi)} : \{0,1\} \times \mathbb{R} \to \mathbb{R}$ such that

$$\ell^{(\psi)}(y,t) = \psi(t) - yt \quad \text{for every } y \in \{0,1\} \text{ and } t \in \mathbb{R}. \tag{13}$$

This definition of a dual loss function is essentially the definition of the Fenchel-Young loss in the literature (see e.g. [11, 5]). Equation (12) implies the following for any $v \in V$ and the corresponding $t = \mathsf{dual}(v)$:

$$\ell(y,v) = \ell^{(\psi)}(y,t). \tag{14}$$

This generalizes (11) where ℓ_{xent} is a special case of the proper loss ℓ, and ℓ_{logistic} is exactly the dual loss $\ell^{(\psi)}$ for the function ψ obtained from Lemma 17. A loss function $\ell^{(\psi)}$ satisfying the relationship in (14) has been referred to as a *composite loss* (see e.g. [7, 42]).

Using Lemma 17 and (14), to study a general proper loss ℓ, it suffices to consider a convex function ψ obtained from Lemma 17 and the dual loss $\ell^{(\psi)}$ defined in (13). The dual loss $\ell^{(\psi)}$ depends on the dual prediction t, so instead of predictors $f : \mathcal{X} \to [0,1]$, it is more convenient to consider *dual predictors* $g : \mathcal{X} \to \mathbb{R}$ that output dual predictions $g(x) \in \mathbb{R}$. We consider the following definition of multicalibration for a dual predictor g. Note that by Lemma 17, we can recover a prediction $v \in V$ from its dual prediction $t = \mathsf{dual}(v)$ by $v = \nabla\psi(t)$, assuming that ψ is differentiable.

▶ **Definition 18** (Multicalibration for dual predictors). *Let $\psi : \mathbb{R} \to \mathbb{R}$ be a differentiable function satisfying $\nabla\psi(t) \in [0,1]$ for every $t \in \mathbb{R}$. Let $\gamma > 0$ and $\mathcal{C}$ be a class of auditor functions $c : \mathcal{X} \times \mathbb{R} \to [-1,1]$. For a dual predictor $g : \mathcal{X} \to \mathbb{R}$, define $f : \mathcal{X} \to [0,1]$ by $f(x) = \nabla\psi(g(x))$. We say g is $(\mathcal{C},\gamma)$-multicalibrated or $(\mathcal{C},\gamma)$-MC if for all $c \in \mathcal{C}$,*

$$\left| \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[c\big(x, g(x)\big) \cdot \big(y - f(x)\big) \right] \right| \le \gamma. \tag{15}$$

We can now state the theorem below which generalizes the loss reduction lemma (Lemma 7) to general proper loss functions. For $\lambda \ge 0$, we say a differentiable function $\psi : \mathbb{R} \to \mathbb{R}$ is λ-smooth if $|\nabla\psi(t_1) - \nabla\psi(t_2)| \le \lambda|t_1 - t_2|$ for every $t_1, t_2 \in \mathbb{R}$. For the cross-entropy loss, the corresponding function ψ is given by $\psi(t) = \ln(1 + e^t)$ and it is (1/4)-smooth.

▶ **Theorem 19** (Proper loss reduction from multicalibration violation [4]). *Let $\psi : \mathbb{R} \to \mathbb{R}$ be a differentiable function satisfying $\nabla\psi(t) \in [0,1]$ for every $t \in \mathbb{R}$. For $\lambda \ge 0$, assume that ψ is λ-smooth. For a dual predictor $g : \mathcal{X} \to \mathbb{R}$, define $f : \mathcal{X} \to [0,1]$ by $f(x) = \nabla\psi(g(x))$. Consider an auditor function $c : \mathcal{X} \times \mathbb{R} \to [-1,1]$. For a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, define $\beta \in [-1,1]$ by*

$$\beta := \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[c\big(x, g(x)\big) \cdot \big(y - f(x)\big) \right].$$

Define a new dual predictor $g' : \mathcal{X} \to \mathbb{R}$ such that $g'(x) = g(x) + (\beta/\lambda)c(x, g(x))$. Then,

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\, \ell^{(\psi)}(y, g'(x)) \le \mathbb{E}_{(x,y)\sim\mathcal{D}}\, \ell^{(\psi)}(y, g(x)) - \beta^2/(2\lambda).$$

We omit the proof of Theorem 19 as it can be found in the work of [4]. We now use Theorem 19 to generalize Theorem 3 to general proper loss functions. Our proof of Theorem 3 relies on the basic fact that the squared loss is bounded in $[0,1]$, but many proper loss functions including the cross-entropy loss do not have a similar boundedness property. Nevertheless, we can still generalize Theorem 3 by making a weaker assumption about the loss function. For the cross-entropy loss ℓ_{xent} and its corresponding dual loss $\ell^{(\psi)}$ (the logistic loss), the following holds if we choose $t_0 = 0$ and $B = \ln 2$:

$$\ell^{(\psi)}(y, t_0) \le \ell^{(\psi)}(y, t) + B \quad \text{for every } y \in \{0,1\} \text{ and } t \in \mathbb{R}, \tag{16}$$

because $\ell^{(\psi)}(y, t_0) = \ln 2$ when $t_0 = 0$, and $\ell^{(\psi)}(y, t) \ge 0$ for every $y \in \{0,1\}$ and $t \in \mathbb{R}$. For a general dual loss $\ell^{(\psi)}$, the assumption that (16) holds for some $t_0 \in \mathbb{R}$ and $B \in \mathbb{R}_{\ge 0}$ is a weaker assumption than boundedness.

Assuming (16), we can extend our main result Theorem 3 to general proper loss functions in Theorem 20 below. For a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ and a class $\mathcal{G}$ of dual predictors $g : \mathcal{X} \to \mathbb{R}$, we say $g \in \mathcal{G}$ is ε-loss-optimal w.r.t. a dual loss $\ell^{(\psi)}$ if

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell^{(\psi)}(y, g(x))] \le \inf_{g' \in \mathcal{G}} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell^{(\psi)}(y, g'(x))] + \varepsilon.$$

▶ **Theorem 20** (Generalization of Theorem 3 to any proper loss). *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$. Let $\psi : \mathbb{R} \to \mathbb{R}$ be a differentiable function satisfying $\nabla\psi(t) \in [0,1]$ for every $t \in \mathbb{R}$. For $\lambda \geq 0$, assume that ψ is λ-smooth. Assume that* (16) *holds for some $t_0 \in \mathbb{R}$ and $B \in \mathbb{R}_{\geq 0}$. Let $\mathcal{C}$ be a class of functions $c : \mathcal{X} \times \mathbb{R} \to [-1,1]$. Let $\mathcal{G}_0, \mathcal{G}_1, \ldots$ be families of dual predictors $g : \mathcal{X} \to \mathbb{R}$ satisfying $\mathcal{G}_0 \subseteq \mathcal{G}_1 \subseteq \cdots$ and that the constant function $g(x) = t_0$ belongs to $\mathcal{G}_0$. For some positive integer k, assume that for every nonnegative integer n, every $g \in \mathcal{G}_n$, every $c \in \mathcal{C}$, and every $\beta \in [-1/\lambda, 1/\lambda]$, the composed dual predictor g' defined below satisfies $g' \in \mathcal{G}_{n+k}$:*

$$g'(x) = g(x) + \beta c(x, g(x)) \quad \text{for every } x \in \mathcal{X}. \tag{17}$$

Then for every $\alpha > 0$, for all but at most Bk/α choices of $n \in \mathbb{Z}_{\geq 0}$, for any $\varepsilon > 0$, every ε-loss-optimal $g \in \mathcal{G}_n$ w.r.t. $\ell^{(\psi)}$ is $\big(\mathcal{C}, \sqrt{2\lambda(\alpha+\varepsilon)}\big)$-MC.

Theorem 20 can be proved similarly to Theorem 3. Specifically, by (16), we have $\mathbb{E}[\ell^{(\psi)}(y, t_0)] \leq \mathbb{E}[\ell^{(\psi)}(y, g(x))] + B$ for any dual predictor $g : \mathcal{X} \to \mathbb{R}$. If we define OPT_n to be the infimum of $\mathbb{E}[\ell^{(\psi)}(y, g(x))]$ over $g \in \mathcal{G}_n$, we have

$$\mathrm{OPT}_0 \leq \mathbb{E}[\ell^{(\psi)}(y, t_0)] \leq \mathrm{OPT}_n + B \tag{18}$$

by our assumption that the constant predictor $g(x) = t_0$ belongs to $\mathcal{G}_0$. Therefore we have $B \geq \mathrm{OPT}_0 - \mathrm{OPT}_n$, bounding the total decrease of OPT_n as a non-increasing function of n. We can combine this bound with Theorem 19 to prove Theorem 20. We omit the details.

Using Theorem 20, we can prove a result for neural networks generalizing Theorem 2 to any proper loss. We use $\widetilde{\mathsf{NN}}_n$ to denote the family of all functions $f : \mathcal{X} \to \mathbb{R}$ computable by an n-node feed-forward network with ReLU activations. We use $\widetilde{\mathsf{NN}}^*_k$ to denote the family of all functions $c : \mathcal{X} \times \mathbb{R} \to [-1,1]$ computable by a k-node feed-forward network with ReLU activations. To make a negative output possible, we allow the output node of a network in $\widetilde{\mathsf{NN}}_n$ and $\widetilde{\mathsf{NN}}^*_k$ to drop the ReLU transformation.

▶ **Theorem 21** (Generalization of Theorem 2 to any proper loss). *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$. Let $\psi : \mathbb{R} \to \mathbb{R}$ be a differentiable function satisfying $\nabla\psi(t) \in [0,1]$ for every $t \in \mathbb{R}$. For $\lambda \geq 0$, assume that ψ is λ-smooth. Assume that* (16) *holds for some $t_0 \in \mathbb{R}$ and $B \in \mathbb{R}_{\geq 0}$. Then for every $k \in \mathbb{Z}_{>0}$ and every $\alpha > 0$, for all but at most $B(k+1)/\alpha$ choices of $n \in \mathbb{Z}_{>0}$, for any $\varepsilon \in (0,1)$, every ε-loss-optimal $g \in \widetilde{\mathsf{NN}}_n$ w.r.t. $\ell^{(\psi)}$ is $\Big(\widetilde{\mathsf{NN}}^*_k, \sqrt{2\lambda(\alpha+\varepsilon)}\Big)$-MC.*

The proof of Theorem 21 is analogous to Theorem 2. We can similarly generalize Theorems 4, 9, and 11 to any proper loss. We omit the details.