Reducing Adversarially Robust Learning to Non-Robust PAC Learning

Omar Montasser
omar@ttic.edu

Steve Hanneke
steve.hanneke@gmail.com

Nathan Srebro
nati@ttic.edu

Toyota Technological Institute at Chicago

Abstract

We study the problem of reducing adversarially robust learning to standard PAC learning, i.e. the complexity of learning adversarially robust predictors using access to only a black-box non-robust learner. We give a reduction that can robustly learn any hypothesis class $C$ using any non-robust learner $A$ for $C$. The number of calls to $A$ depends logarithmically on the number of allowed adversarial perturbations per example, and we give a lower bound showing this is unavoidable.

1 Introduction

We consider the problem of learning predictors that are robust to adversarial examples at test time. That is, we would like to be robust against an adversary $U : \mathcal{X} \rightarrow 2^\mathcal{X}$ that can perturb examples at test-time, where $U(x) \subseteq \mathcal{X}$ is the set of allowed corruptions the adversary might replace $x$ with, as measured by the robust risk:

$$R_U(h; D) \triangleq \mathbb{E}_{(x, y) \sim D} \left[ \sup_{z \in U(x)} 1[h(z) \neq y] \right].$$

(1)

For example, $U$ could be perturbations of bounded $\ell_p$-norms [Goodfellow et al., 2015].

We ask whether we can adversarially robustly learn a given target hypothesis class $C \subseteq \mathcal{Y}^\mathcal{X}$ (e.g. neural networks)—that is, whether, if there exists a predictor in $C$ with zero robust risk w.r.t. some unknown distribution $D$ over $\mathcal{X} \times \mathcal{Y}$, can we find a predictor with (arbitrarily small) robust risk using $m$ i.i.d. (uncorrupted) samples $S = \{(x_i, y_i)\}_{i=1}^m$ from $D$. Recently, Montasser et al. [2019] showed that if $C$ is PAC-learnable non-robustly, then $C$ is also adversarially robustly learnable. However, their result is not constructive and the robust learning algorithm given is inefficient, complex, and does not actually directly use a non-robust learner. In this paper, we ask a more constructive version of this question:

*Can we learn adversarially robust predictors given only black-box access to a non-robust learner?*

That is, we are asking whether it is possible to reduce adversarially robust learning to standard non-robust learning. Since we have a plethora of algorithms devised for standard non-robust learning, it would be useful if we could design efficient *reduction* algorithms that leverage such non-robust learning algorithms in a black-box manner to learn robustly. That is, design generic wrapper methods that take as input a learning algorithm $A$ and a specification of the adversary $U$, and robustly learn by calling $A$. Many systems in practice perform standard learning but with no robustness guarantees, and therefore, it would be beneficial to provide wrapper procedures that can guarantee adversarial robustness in a black-box manner without needing to modify current learning systems internally.

Related Work Recent work [Mansour et al., 2015, Feige et al., 2015, 2018, Attias et al., 2019] can be interpreted as giving reduction algorithms for adversarially robust learning. Specifically, Feige et al. [2015] gave a reduction algorithm that can robustly learn a finite hypothesis class $C$ using black-box access to an ERM for $C$. Later, Attias et al. [2019] improved this to handle infinite hypothesis classes $C$. But their complexity and the number of calls to ERM depend super-linearly on the number of possible perturbations $|U| = \sup_x |U(x)|$, which is undesirable for most types of perturbations—we completely avoid a sample complexity dependence on $|U|$, and reduce the oracle complexity to at most a poly-logarithmic dependence. Furthermore, their work assumes access specifically to an ERM procedure, which is a very specific type of learner, while we only require access to any method that PAC-learns $C$ and whose image has bounded VC-dimension.

A related goal was explored by Salman et al. [2020]: They proposed a method to robustify pre-trained predictors. Their method takes as input a black-box predictor (not a learning algorithm) and a point $x$, and outputs a label prediction $y$ for $x$ and a radius $r$ such that the label $y$ is robust to $\ell_q$ perturbations of radius $r$. But this doesn’t guarantee that the predictions $y$ are correct, nor that the radius $r$ would be what we desire, and even if the predictor was returned by a learning algorithm and has a very small non-robust error, we do not end up with any guarantee on the robust risk of the robustified predictor. In this paper, we require black-box access to a learning algorithm (not just to a single predictor), but we output a predictor that is guaranteed to have small robust risk (if one exists in the class, see Definition 2.2). We also provide a general treatment for arbitrary adversaries $U$, not just $\ell_p$ perturbations.

Finally, we note that the approach of Montasser et al. [2019] can be interpreted as using black-box access to an oracle RERM$_C$ minimizing the robust empirical risk:

$$\hat{h} \in \text{RERM}_C(S) \triangleq \arg\min_{h \in C} \frac{1}{m} \sum_{i=1}^m \sup_{z \in U(x)} \mathbb{I}[h(z) \neq y].$$

But this goes well beyond just a non-robust learning algorithm, or even ERM.

Efficient Reductions From a computational perspective, the relationship between standard non-robust learning and adversarially robust learning is not well-understood. It is natural to wonder whether there is a general efficient reduction for adversarially robust learning, using only non-robust learners. Recent work has provided strong evidence that this is not the case in general. Specifically, Bubeck et al. [2019] showed that there exists a learning problem that can be learned efficiently non-robustly, but is computationally intractable to learn robustly (under plausible complexity-theoretic assumptions). In this paper, we aim to understand when such efficient reductions are possible.

Main Results When studying reductions of adversarially robust learning to non-robust learning, an important aspect emerges regarding the form of access that the reduction algorithm has to the adversary $\mathcal{U}$. How should we model access to the sets of adversarial perturbations represented by $\mathcal{U}$?

In Section 3, we study the setting where the reduction algorithm has explicit access/knowledge of the possible adversarial perturbations induced by the adversary $\mathcal{U}$ on the training examples. We first show that there is an algorithm that can learn adversarially robust predictors with black-box oracle access to a non-robust algorithm:

**Theorem 3.1** (Informal). For any adversary $\mathcal{U}$, Algorithm 1 robustly learns any target class $C$ using any black-box non-robust PAC learner $A$ for $C$, with $O(\log^2 |\mathcal{U}|)$ oracle calls to $A$ and sample complexity independent of $|\mathcal{U}|$.

The oracle complexity dependence on $|\mathcal{U}|$, even if only logarithmic, might be disappointing, but we show it is unavoidable:

**Theorem 3.2** (Informal). There exists an adversary $\mathcal{U}$ such that for any reduction algorithm $B$, there exists a target class $C$ and a PAC learner $A$ for $C$ such that $\Omega(\log |\mathcal{U}|)$ oracle queries to $A$ are necessary to robustly learn $C$.

This tells us that only requiring a non-robust PAC learner $A$ is not enough to avoid the $\log |\mathcal{U}|$ dependence, even with explicit knowledge of $\mathcal{U}$. In Section 4, we show that having an online learner $A$ for $C$, allows us to robustly learn $C$ with access to a mistake oracle for $\mathcal{U}$ (see Definition 4.1) where no explicit knowledge of $\mathcal{U}$ is assumed and no dependence on $|\mathcal{U}|$ is incurred:

**Theorem 4.2** (Informal). There exists an algorithm $B$ that can robustly learn any target class $C$ w.r.t. any adversary $\mathcal{U}$ when given access to a mistake oracle $Q_\mathcal{U}$ and a black-box online learner $A$ for $C$. The sample complexity, number of calls to $A$, and number of calls to $Q_\mathcal{U}$ are independent of $|\mathcal{U}|$. 

2
2 Preliminaries

Let $X$ denote the instance space and $Y = \{\pm 1\}$ denote the label space. Let $U : X \rightarrow 2^X$ denote an arbitrary adversary. For any adversary $U$, denote by $|U| \triangleq \sup_x |U(x)|$ the number of allowed adversarial perturbations. We start with formalizing the notions of non-robust (standard) PAC learning and robust PAC learning:

**Definition 2.1 (PAC Learnability).** A target hypothesis class $C \subseteq Y^X$ is said to be PAC learnable if there exists a learning algorithm $A : (X \times Y)^* \rightarrow Y^X$ with sample complexity $m(\varepsilon, \delta) : (0, 1)^2 \rightarrow \mathbb{N}$ such that: for any $\varepsilon, \delta \in (0, 1)$, for any distribution $D$ over $X \times Y$, and any target concept $c \in C$ with zero risk, $\text{err}_D(c) = 0$, with probability at least $1 - \delta$ over $S \sim D^{m(\varepsilon, \delta)}$,

$$\text{err}_D(A(S)) \leq \Pr_{(x,y)\sim D} [A(S)(x) \neq y] \leq \varepsilon.$$

**Definition 2.2 (Robust PAC Learnability).** A target hypothesis class $C \subseteq Y^X$ is said to be robustly PAC learnable with respect to adversary $U$ if there exists a learning algorithm $B : (X \times Y)^* \rightarrow Y^X$ with sample complexity $m(\varepsilon, \delta) : (0, 1)^2 \rightarrow \mathbb{N}$ such that: for any $\varepsilon, \delta \in (0, 1)$, for any distribution $D$ over $X \times Y$, and any target concept $c \in C$ with zero robust risk, $R_U(c; D) = 0$, with probability at least $1 - \delta$ over $S \sim D^{m(\varepsilon, \delta)}$,

$$R_U(B(S); D) \leq \varepsilon.$$

We recall the Vapnik-Chervonenkis dimension (VC dimension) is defined as follows:

**Definition 2.3 (VC dimension).** We say that a sequence $\{x_1, \ldots, x_k\} \in X$ is shattered by $C$ if $\forall y_1, \ldots, y_k \in Y, \exists h \in C$ such that $\forall i \in [k], h(x_i) = y_i$. The VC dimension of $C$ (denoted $vc(C)$) is then defined as the largest integer $k$ for which there exists $\{x_1, \ldots, x_k\} \in X$ that is shattered by $C$. If no such $k$ exists, then $vc(C)$ is said to be infinite.

Another important complexity measure that is utilized in the study of robust PAC learning is the notion of dual VC dimension, which we define below:

**Definition 2.4 (Dual VC dimension).** Consider a dual space $G$: a set of functions $g_z : C \rightarrow Y$ defined as $g_z(h) = h(x)$, for each $h \in C$ and each $x \in X$. Then, the dual VC dimension of $C$ (denoted $vc^*(C)$) is defined as the VC dimension of $G$. In other words, $vc^*(C) = vc(G)$ and it represents the largest set $\{h_1, \ldots, h_k\}$ that is shattered by points in $X$.

If the VC dimension is finite, then so is the dual VC dimensio, and it can be bounded as $vc^*(C) \leq 2^{vc(C) + 1}$ [Assouad, 1983]. Although this exponential dependence is tight for some classes, for many natural classes, such as linear predictors and some neural networks (see, e.g. Lemma 3.2), the primal and dual VC dimensions are equal, or at least polynomially related.

We also formally define what we mean by a reduction algorithm:

**Definition 2.5 (Reduction Algorithm).** For an adversary $U$, a reduction algorithm $B_U$ takes as input a black-box learning algorithm $A$ and a training set $S \subseteq X \times Y$, and can use $A$ by calling it $T$ times on inputs $B_U$ constructs each of size $m_0 \in \mathbb{N}$, and outputs a predictor $f \in Y^X$.

We emphasize that $B_U$ is allowed to be adaptive in its calls to $A$. That is, it can call $A$ on one constructed data set, then construct another data set depending on the returned predictor, and call $A$ on this new data set. Such adaptive use of the base learner $A$ is central to boosting-type constructions.

We know that a hypothesis class $C$ is PAC learnable if and only if its VC dimension is finite [Vapnik and Chervonenkis, 1971, 1974, Blumer et al., 1989, Ehrenfeucht et al., 1989]. And in this case, $C$ is properly PAC learnable with ERM$_C$. Montasser et al. [2019, Theorem 4] showed that if $C$ is PAC learnable, then $C$ is adversarially robustly PAC learnable with an improper learning rule that required a RERM$_C$ oracle (see Equation 2) and sample complexity of $\hat{O} \left( \frac{vc(C)vc^*(C)}{\varepsilon} \right)$. In this paper, we study whether it is possible to adversarially robustly PAC learn $C$ using only a black-box non-robust PAC learner $A$ for $C$. We will not require $A$ is “proper” (i.e. returns a predictor in $C$), but we will rely on it returning a predictor in some, possibly much larger, class which still has finite VC-dimension. To this end, we denote by $vc(A) = vc(\text{im}(A))$ and $vc^*(A) = vc^*(\text{im}(A))$ the primal and dual VC dimension of the image of $A$, i.e. the class $\text{im}(A) = \{A(S) | S \in (X \times Y)^*\}$ of the possible hypothesis $A$ might return. For ERM, or any other proper learner, $\text{im}(A) \subseteq C$ and so $vc(A) \leq vc(C)$ and $vc^*(A) \leq vc^*(C)$.
3 Learning with Explicitly Specified Adversarial Perturbations

When studying reductions of adversarially robust PAC learning to non-robust PAC learning, an important aspect emerges regarding the form of access that the reduction algorithm has to the adversary $\mathcal{U}$. How should we model access to the sets of adversarial perturbations represented by $\mathcal{U}$?

In this section, we explore the setting where the reduction algorithm has explicit knowledge of the adversary $\mathcal{U}$. That is, the reduction algorithm knows the set of possible adversarial perturbations for each example in the training set. This is in accordance with what is typically considered in practice, where the adversary $\mathcal{U}$ (e.g. $\ell_\infty$ perturbations) is known to the algorithm, and this knowledge is used in adversarial training (see e.g. Madry et al. [2018]). Formally, we consider the following question:

For any adversary $\mathcal{U}$, does there exist an algorithm that can learn a target class $C$ robustly w.r.t $\mathcal{U}$ given only a black-box non-robust PAC learner $A$ for $C$?

We give a positive answer to this question. In Theorem 3.1, we present an algorithm (see Algorithm 1)—based on the $\alpha$-Boost algorithm [Schapire and Freund, 2012, Section 6.4.2] and recent work of Montasser et al. [2019, Theorem 4]—that can adversarially robustly PAC learn a target class $C$ with only black-box oracle access to a PAC learner $A$ for $C$.

**Algorithm 1: Robustify The Non-Robust**

**Input:** Training dataset $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, black-box non-robust learner $A$

1. Inflate dataset $S$ to $S_\mathcal{U} = \bigcup_{x \in S} \{(z, y) : z \in \mathcal{U}(x)\}$. // $S_\mathcal{U}$ contains all possible perturbations of $S$.
2. Set $m_0 = O(vc(A)vc(\mathcal{A})^* \log vc(\mathcal{A})^*)$, and $T = O(\log |S_\mathcal{U}|)$.
3. for $1 \leq t \leq T$ do
   4. Set distribution $D_t$ on $S_\mathcal{U}$ as in the $\alpha$-Boost algorithm.
   5. Sample $S' \sim D_{t,m_0}$, and project $S'$ to dataset $L \subseteq S$ by replacing each perturbation $z$ with its corresponding example $x$.
   6. Call $\text{ZeroRobustLoss}$ on $L$, and denote by $f_t$ its output predictor.
   7. Sample $N_{co} = O(vc^*(A) \log vc^*(A))$ i.i.d. indices $i_1, \ldots, i_{N_{co}} \sim \text{Uniform}\{1, \ldots, T\}$.
   8. (repeat previous step until $f = \text{MAJ}(f_{i_1}, \ldots, f_{i_{N_{co}}})$ has $R_{\mathcal{U}}(f; S) = 0$)
   9. Output: A majority-vote $\text{MAJ}(f_{i_1}, \ldots, f_{i_{N_{co}}})$ predictor.

**ZeroRobustLoss(Dataset $L$, Learner $A$):**

10. Inflate dataset $L$ to $L_\mathcal{U} = \bigcup_{(x, y) \in L} \{(z, y) : z \in \mathcal{U}(x)\}$, and set $T_L = O(\log |L_\mathcal{U}|)$
11. Run $\alpha$-Boost with black-box access to $A$ on $L_\mathcal{U}$ for $T_L$ rounds.
12. Let $h_1, \ldots, h_{T_L}$ denote the hypotheses produced by $\alpha$-Boost with $T_L$ oracle queries to $A$.
13. Sample $N = O(vc^*(A))$ i.i.d. indices $i_1, \ldots, i_N \sim \text{Uniform}\{1, \ldots, T_L\}$.
14. (repeat previous step until $f = \text{MAJ}(h_{i_1}, \ldots, h_{i_N})$ has $R_{\mathcal{U}}(f; L) = 0$)
15. return $f = \text{MAJ}(h_{i_1}, \ldots, h_{i_N})$.

**$\alpha$-Boost(Dataset $L$, Learner $A$):**

16. Initialize $D_1$ to be uniform over $L$, and set $T_L = O(\log |L|)$.
17. for $1 \leq t \leq T_L$ do
18. Run $A$ on $S' \sim D_{t,m_0}$, and denote by $h_t$ its output. (repeat until $\text{err}_{D_t}(h_t) \leq 1/3$)
19. Compute a new distribution $D_{t+1}$ by applying the following update for each $(x, y) \in L$:
20. 
   $$D_{t+1}(x) = \frac{D_t(x)}{Z_t} \times \begin{cases} e^{-2\alpha} & \text{if } h_t(x) = y \\ 1 & \text{otherwise} \end{cases}$$
21. where $Z_t$ is a normalization factor and $\alpha$ is set as in Lemma 3.3
22. return $h_1, \ldots, h_{T_L}$.

**Theorem 3.1.** For any adversary $\mathcal{U}$, Algorithm 1 can robustly PAC learn any target class $C$ using black-box oracle calls to any PAC learner $A$ for $C$ with:

1. Sample Complexity $m = O \left( \frac{dd^* \log^2 d^*}{\epsilon} \log \left( \frac{dd^* \log^2 d^*}{\epsilon} + \frac{\log(1/\delta)}{\epsilon} \right) \right)$.
2. Oracle Complexity $T = O \left( \left( \log m + \log |U| \right)^2 + \log(1/\delta) \right)$.
where \( d = \text{vc}(A) \) and \( d^* = \text{vc}^*(A) \) are the primal and dual VC dimension of \( A \).

Importantly, the sample complexity of Algorithm 1 is independent of the number of allowed perturbations \(|U|\), in contrast to work by Attias et al. [2019], that can be interpreted as giving a reduction with sample complexity \( m \propto |U| \log |U| \) and oracle complexity \( T \propto |U| \log^2 |U| \).

Before proceeding with the proof of Theorem 3.1, we briefly describe our strategy and its main ingredients. Given a dataset \( S \) that is robustly realizable by some target concept \( c \in \mathcal{C} \), we show that we can use the non-robust learner \( A \) to implement a RERM oracle that guarantees zero empirical robust loss on \( S \) using \( \text{ZeroRobustLoss} \) in Algorithm 1. But what about the population robust loss? Our main goal is to adversarially robustly learn \( c \) and not just minimize the empirical robust loss. Fortunately, we show that the arguments on robust generalization based on sample compression in [Montasser et al., 2019, Theorem 4] will still go through when we replace the RERM\(_C\) oracle they used with our \( \text{ZeroRobustLoss} \) procedure in Algorithm 1. This is achieved by showing that the image of \( \text{ZeroRobustLoss} \) has bounded VC dimension and dual VC dimension. The following lemma, whose proof is provided in Appendix A, bounds the dual VC dimension of the convex-hull of a class \( \mathcal{H} \). This result might be of independent interest.

**Lemma 3.2.** Let \( \text{co}^k(\mathcal{H}) = \{x \mapsto \text{MAJ}(h_1, \ldots, h_k)(x) : h_1, \ldots, h_k \in \mathcal{H} \} \). Then, the dual VC dimension of \( \text{co}^k(\mathcal{H}) \) satisfies \( \text{vc}^*(\text{co}^k(\mathcal{H})) \leq O(d^* \log k) \).

In addition, we state two extra key lemmas that will be useful for us in the proof. First, Lemma 3.3 states that running \( \alpha\)-Boost on a dataset for enough rounds produces a sequence of predictors that achieve zero loss on the dataset (with a margin).

**Lemma 3.3** (see, e.g., Corollary 6.4 and Section 6.4.3 in Schapire and Freund [2012]). Let \( S = \{(x_i, c(x_i))\}_{i=1}^m \) be a dataset where \( c \in \mathcal{C} \) is some target concept, and \( A \) an arbitrary PAC learner for \( \mathcal{C} \) (for \( \varepsilon = 1/3, \delta = 1/3 \)). Then, running \( \alpha\)-Boost (see description in Algorithm 1) on \( S \) with black-box oracle access to \( A \) with \( \alpha = \frac{1}{2} \log \left(1 + \frac{\sqrt{2\log m}}{T}\right) \) for \( T = \lceil 112 \log(m) \rceil = O(\log m) \) rounds suffices to produce a sequence of hypotheses \( h_1, \ldots, h_T \in \text{im}(A) \) such that

\[
\forall (x, y) \in S, \quad \frac{1}{T} \sum_{i=1}^T \mathbb{I}[h_i(x) = y] \geq \frac{5}{9},
\]

In particular, this implies that the majority-vote MAJ\((h_1, \ldots, h_T)\) achieves zero error on \( S \).

Second, Lemma 3.4 describes a sparsification technique due to Moran and Yehudayoff [2016] which allows us to control the complexity of the majority-vote predictors that we use in Algorithm 1.

**Lemma 3.4** (Sparsification of Majority Votes, Moran and Yehudayoff [2016]). Let \( \mathcal{H} \) be a hypothesis class with finite primal and dual VC dimension, and \( h_1, \ldots, h_T \) be predictors in \( \mathcal{H} \). Then, for any \((\varepsilon, \delta) \in (0, 1)\), with probability at least \( 1 - \delta \) over \( N = O\left(\frac{\text{vc}(\mathcal{H}) + \log(1/\delta)}{\varepsilon^2}\right) \) independent random indices \( i_1, \ldots, i_N \sim \text{Uniform}\{1, \ldots, T\} \), we have:

\[
\forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \quad \left| \frac{1}{N} \sum_{j=1}^N \mathbb{I}[h_{i_j}(x) = y] - \frac{1}{T} \sum_{i=1}^T \mathbb{I}[h_i(x) = y] \right| < \varepsilon.
\]

We are now ready to proceed with the proof of Theorem 3.1.

**Proof of Theorem 3.1.** Let \( \mathcal{U} \) be an arbitrary adversary. Let \( \mathcal{C} \) be a target class that is PAC learnable with some PAC learner \( A \). Let \( \mathcal{H} \) denote the base class of hypotheses of learner \( A \). Let \( d \) denote the VC dimension of \( \mathcal{H} \), and \( d^* \) denote the dual VC dimension of \( \mathcal{H} \). Our proof is divided into two parts.

**Zero Empirical Robust Loss.** Let \( L = \{(x_1, y_1), \ldots, (x_m, y_m)\} \) be a dataset that is robustly realizable with some target concept \( c \in \mathcal{C} \); in other words, for each \((x, y) \in L \) and each \( z \in \mathcal{U}(x) \), \( c(z) = y \). We will show that we can use the non-robust learner \( A \) to guarantee zero empirical robust loss on \( L \). This procedure is described in \( \text{ZeroRobustLoss} \) in Algorithm 1. We inflate dataset \( L \) to include all possible perturbations under the adversary \( \mathcal{U} \). Let \( L_\mathcal{U} = \bigcup_{i \leq m} \{(z, y_i) : z \in \mathcal{U}(x_i)\} \) denote the inflated dataset. Observe that \( |L_\mathcal{U}| \leq m|\mathcal{U}| \), since each point \( x \in \mathcal{X} \) has at most \(|\mathcal{U}| \)
possible perturbations. We run the $\alpha$-Boost algorithm on the inflated dataset $L_d$ with black-box access to PAC learner $A$, where in each round of boosting $m_0$ samples are fed to $A$ (where $m_0$ is chosen Step 2). By Lemma 3.3, running $\alpha$-Boost with $T = O(\log(|L_d|))$ oracle calls to $A$ suffices to produce a sequence of hypotheses $h_1, \ldots, h_T \in \mathcal{H}$ such that

$$\forall (z, y) \in L_d, \frac{1}{T} \sum_{i=1}^{T} I[h_i(z) = y] \geq \frac{5}{9}.$$ 

Specifically, the above implies that a majority-vote over hypotheses $h_1, \ldots, h_T$ achieves zero robust loss on dataset $L$, $R_d(MAJ(h_1, \ldots, h_T); L) = 0$. By Step 13 in ZeroRobustLoss in Algorithm 1 and Lemma 3.4 (with $\varepsilon = 1/18, \delta = 1/3$), we have that for $N = O(d^*)$, the sampled predictors $h_{i_1}, \ldots, h_{i_N}$ satisfy

$$\forall (z, y) \in L_d, \frac{1}{N} \sum_{j=1}^{N} I[h_{i_j}(z) = y] > \frac{1}{T} \sum_{i=1}^{T} I[h_i(z) = y] - \frac{1}{18} > \frac{5}{9} - \frac{1}{18} = \frac{1}{2}.$$ 

Therefore, the majority-vote over the sampled hypotheses $\text{MAJ}(h_{i_1}, \ldots, h_{i_N})$ achieves zero robust loss on $L, R_d(MAJ(h_{i_1}, \ldots, h_{i_N}); S) = 0$. Thus, we can implement a RERM oracle (see Equation 2) using the procedure ZeroRobustLoss in Algorithm 1. The sparsification step (Step 12) controls the complexity of the image of ZeroRobustLoss, i.e., the hypothesis class that is being implicitly used. Specifically, observe that the sparsified predictor $f = \text{MAJ}(h_{i_1}, \ldots, h_{i_N})$ lives in $co^{O(d^{'})}(\mathcal{H})$, which is the convex-hull of $\mathcal{H}$ that combines at most $O(d^{'})$ predictors. To guarantee robust generalization in the next part, it suffices to bound the VC dimension and dual VC dimension of $co^{O(d^{'})}(\mathcal{H})$. By [Blumer et al., 1989], the VC dimension of $co^{O(d^{'})}(\mathcal{H})$ is at most $O(d^* \log d^*)$, and by Lemma 3.2, the dual VC dimension of $co^{O(d^{'})}(\mathcal{H})$ is at most $O(d^* \log d^*)$.

**Robust Generalization through Sample Compression.** This part builds on the approach of Montasser et al. [2019, Theorem 4]. Specifically, we observe that their proof works even if we replace the RERM oracle they used, with our ZeroRobustLoss procedure in Algorithm 1 that is described above. We provide a self-contained analysis below.

Let $D$ be an arbitrary distribution over $X \times Y$ that is robustly realizable with some concept $c \in \mathcal{C}$, i.e., $R_d(c; D) = 0$. Fix $\varepsilon, \delta \in (0, 1)$ and a sample size $m$ that will be determined later. Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be an i.i.d. sample from $D$. We run the $\alpha$-Boost algorithm (see Algorithm 1) on the inflated dataset $S_d$, this time with ZeroRobustLoss as the subprocedure. Specifically, on each round of boosting, $\alpha$-Boost computes an empirical distribution $D_t$ over $S_d$ (according to Step 18). We draw $m_0 = O(dd^* \log d^*)$ samples $S'$ from $D_t$, and project $S'$ to a dataset $L_t \subset S$ by replacing each perturbation $(z, y) \in S'$ with its corresponding original point $(x, y) \in S$, and then we run ZeroRobustLoss on dataset $L_t$. The projection step is crucial for the proof to work, since we use a sample compression argument to argue about robust generalization, and the sample compression must be done on the original points that appeared in $S$ rather than the perturbations in $S_d$.

By classic PAC learning guarantees [Vapnik and Chervonenkis, 1974, Blumer et al., 1989], with $m_0 = O(\text{vc}(co^{O(d^{'})}(\mathcal{H}))) = O(dd^* \log d^*)$, we are guaranteed uniform convergence of 0-1 risk over predictors in $co^{O(d^{'})}(\mathcal{H})$ (the effective hypothesis class used by ZeroRobustLoss). So, for any distribution $D$ over $X \times Y$ with inf$_{c \in C} \text{err}(c; D) = 0$, with nonzero probability over $S' \sim D^{m_0}$, every $f \in co^{O(d^{'})}(\mathcal{H})$ satisfying $\text{err}_{S'}(f) = 0$, also has $\text{err}_{D}(f) < 1/3$. By the guarantee of ZeroRobustLoss (established above), we know that $f_t = \text{ZeroRobustLoss}(L_t, A)$ achieves zero robust loss on $L_t$, $R_d(f_t; L_t) = 0$, which by definition of the projection means that $\text{err}_{S'}(f_t) = 0$, and thus $\text{err}_{D}(f_t) < 1/3$. This allows us to use ZeroRobustLoss with $\alpha$-Boost to establish a robust generalization guarantee. Specifically, Lemma 3.3 implies that running the $\alpha$-Boost algorithm with $S_d$ as its dataset for $T = O(\log(|S_d|))$ rounds, using ZeroRobustLoss to produce the hypotheses $f_t \in co^{O(d^{'})}(\mathcal{H})$ for the distributions $D_t$ produced in each round of the algorithm, will produce a sequence of hypotheses $f_1, \ldots, f_T \in co^{O(d^{'})}(\mathcal{H})$ such that:

$$\forall (z, y) \in S_d, \frac{1}{T} \sum_{i=1}^{T} I[f_i(z) = y] \geq \frac{5}{9}.$$
we will sparsify the majority-vote as in Step 7 in Algorithm 1. Lemma 3.4 (with $\varepsilon$) for an appropriately large numerical constant $\varepsilon$.

The oracle complexity of Algorithm 1 depends on $\text{vc}(C)$ for some $L_t \subseteq S$ with $|L_t| = m_0$. Thus, the classifier $\text{MAJ}(f_1, \ldots, f_T)$ is representable as the value of an (order-dependent) reconstruction function $\phi$ with a compression set size

$$m_0 T = O(\text{vc}(C) \log(|S_t|)) = O(d^* \log d^* (\log m + \log |U|)),$$

This is not enough, however, to obtain a sample complexity bound that is independent of $|U|$. For that, we will sparsify the majority-vote as in Step 7 in Algorithm 1. Lemma 3.4 (with $\varepsilon = 1/18$, $\delta = 1/3$) guarantees that for $N_{co} = O(d^* \log d^*)$, the sampled predictors $f_1, \ldots, f_{N_{co}}$ satisfy:

$$\forall (z, y) \in S_U, \quad \frac{1}{N_{co}} \sum_{j=1}^{N_{co}} \mathbb{I}[f_j(z) = y] > \frac{1}{T} \sum_{i=1}^{T} \mathbb{I}[f_i(z) = y] - \frac{1}{18} > \frac{5}{9} - \frac{1}{18} = \frac{1}{2},$$

so that the majority-vote achieves zero robust loss on $S$, $R_{\text{U}}(\text{MAJ}(f_1, \ldots, f_{N_{co}}); S) = 0$. Since again, each $f_i$ is the result of $\text{ZeroRobustLoss}(L, A)$ for some $L_t \subseteq S$ with $|L_t| = m_0$, we have that the classifier $\text{MAJ}(f_1, \ldots, f_{N_{co}})$ can be represented as the value of an (order-dependent) reconstruction function $\phi$ with a compression set size $m_0 N_{co} = O(d^* \log d^* \cdot d^* \log d^*) = O(d^2 \log^2 d^*)$. Lemma A.1 (Montasser et al. [2019]) which extends to the robust loss the classic compression-based generalization guarantees from the 0-1 loss, implies that for $m \geq c d d^2 \log^2 d^*$ (an appropriately large numerical constant $c$), with probability at least $1 - \delta$ over $S \sim \mathcal{D}^m$,

$$R_{\text{U}}(\text{MAJ}(f_1, \ldots, f_{N_{co}}); \mathcal{D}) \leq O\left(\frac{dd^2 \log^2 d^*}{m} \log(m) + \frac{1}{m} \log(1/\delta)\right).$$

Setting this less than $\varepsilon$ and solving for a sufficient size of $m$ to achieve this yields the stated sample complexity bound.

Our oracle complexity $T$ (number of calls to $A$) is at most $O((\log |S_U|)^2 + \log(1/\delta)) \leq O\left( (\log m + \log |U|)^2 + \log(1/\delta) \right)$, since $\text{ZeroRobustLoss}$ in Algorithm 1 terminates in at most $O(\log |S_U|)$ rounds each time it is invoked, and it is invoked at most $O(\log |S_U|)$ times by the outer $\alpha$-Boost algorithm in Algorithm 1. Therefore, we have at most $O\left( (\log m + \log |U|)^2 \right)$ geometric random variables that represent that number of times Step 19 is invoked, which is the step where learner $A$ is called. The success probability of Step 19 is a constant (say 2/3), therefore the mean of the sum of the geometric random variables is $O\left( (\log m + \log |U|)^2 \right)$. Since sums of geometric random variables concentrate around the mean [Brown], we get that with probability at least $1 - \delta$, the total number of times Step 19 is executed is at most $O\left( (\log m + \log |U|)^2 + \log(1/\delta) \right)$. This concludes the proof.

### 3.1 A Lowerbound on the Oracle Complexity

The oracle complexity of Algorithm 1 depends on $\log |U|$. Can this dependence be reduced or avoided? Unfortunately, we show in Theorem 3.5 that the dependence on $\log |U|$ is unavoidable and no reduction algorithm can do better.

It is relatively easy to show, by relying on lower bounds from the boosting literature [Schapire and Freund, 2012, Section 13.2.2], that for any reduction algorithm, there exists a target class $C$ with $\text{vc}(C) \leq 1$ and a PAC learner $A$ for $C$ such that $\Omega(\log |U|)$ oracle calls to this PAC learner are necessary to achieve zero empirical robust loss. But this is done by constructing a “crazy” improper learner with $\text{vc}(A) \propto |U| \gg \text{vc}(C)$.

Perhaps we can ensure better oracle complexity by requiring a more constrained PAC learner $A$, e.g. with low VC dimension (as in our upper bound), or perhaps even a proper learner, or an ERM, or maybe where $\text{im}(A)$ and so also $C$ finite. We next present a lower bound to show that none of these help improve the oracle complexity. Specifically, we will present a construction showing that for any reduction algorithm $B$ there is a randomized target class $C$ and a PAC learner $A$ for $C$ with
With such an oracle, Algorithm 1 can be implemented without the need to do explicit inflation $U$ as well as access via a sampling oracle. A more realistic form of access is having a mistake oracle for

**Theorem 3.5.** For any sufficiently large integer $u$, if $|X| \geq 2^{10u}$, there exists an adversary $U$ with $|U| = u$ such that for any reduction algorithm $B$ and for any $\varepsilon > 0$, there exists a target class $C$ and a PAC learner $A$ for $C$ with $vc(A) = 1$ such that, if the training sample has size at most $(1/8)|U|^9$, then $B$ needs to make $T \geq \frac{\log |U|}{\log(2/\varepsilon)}$ oracle calls to $A$ in order to robustly learn $C$.

**Computational Efficiency** Although the sample complexity of Algorithm 1 is independent of $|U|$, we showed that the $\log |U|$ dependence in oracle complexity is unavoidable. This implies that the runtime of Algorithm 1 will be at best weakly polynomial and have at least a $\log |U|$ dependence. But maybe this is not so bad, because it is equivalent to the number of bits required to represent the adversarial perturbations. This weak poly-time dependence is common in almost all optimization algorithms (gradient descent, interior point methods, etc). What is more concerning is the linear runtime and memory dependence on $|U|$ that emerges from the explicit representation of the adversarial perturbations during training. In practice, many of the adversarial perturbations $U$ are infinite, but specified implicitly, and not by enumerating over all possible perturbations (e.g. $\ell_p$ perturbations). This motivates the following next steps: What operations do we need to be able to implement efficiently on $U$ in order to robustly learn? What access (oracle calls, or “interface”) do we need to $U$?

**Sampling Oracle Over Perturbations** A reasonable form of access to $U$ that is sufficient for implementing Algorithm 1 is a sampling oracle that takes as input a point $x$ and an energy function $E : X \to \mathbb{R}$, and does the following:

(a) Samples a perturbation $z$ from a distribution given by $p_x(z) \propto \exp(E(z)) \cdot \mathbb{1}[z \in U(x)]$. That is, the oracle samples from the set $U(x)$ based on the weighting encoded in $E$.

(b) Calculates $\Pr[z \in U(x)]$ for the distribution given by $p(z) \propto \exp(E(z))$.

With such an oracle, Algorithm 1 can be implemented without the need to do explicit inflation of $S$ to $S_{U}$, and can avoid the linear dependence on $|U|$. This is because Algorithm 1 and its subprocedure ZeroRobustLoss just need to sample from distributions over the inflated set $S_{U}$ that are constructed by $\alpha$-Boost (as required in Steps 5 and 17 in Algorithm 1). This can be simulated via a two-stage process where we maintain a conditional distribution over $S$ (the original points), and then draw perturbations using the sampling oracle. Specifically, to sample from a distribution $D_{t}$ that is constructed by $\alpha$-Boost, we use two energy functions $E^+_t(z) = -2\alpha \sum_{i \leq t} \mathbb{1}[g_i(z) = +1]$ and $E^-_t(z) = -2\alpha \sum_{i \leq t} \mathbb{1}[g_i(z) = -1]$, where $g_1, \ldots, g_t$ represent the sequence of predictors produced during the first $t$ rounds of boosting (either $h_i$’s produced by non-robust learner $A$ or $f_i$’s produced by ZeroRobustLoss). Using the sampling oracle, we can sample from $D_{t}$ by first sampling $(x, y)$ from $S$ based on the marginal estimates computed by the oracle (operation (b) described above) using energy function $E^+_t$, and then sampling $z$’s from their $U(x)$ (operation (a) described above) using energy function $E^-_t$.

### 4 Learning with a Mistake Oracle for Adversarial Perturbations

In Section 3, we considered an explicit form of access to the set of adversarial perturbations $U$, as well as access via a sampling oracle. A more realistic form of access is having a mistake oracle for $U$:

**Definition 4.1 (Mistake Oracle).** Denote by $O_{U}$ a mistake oracle for $U$. $O_{U}$ takes as input a predictor $f : X \to Y$ and an example $(x, y)$ and either: (a) asserts that $f$ is robust on $(x, y)$ (i.e. $\forall z \in U(x), f(z) = y$), or (b) returns an adversarial perturbation $z \in U(x)$ such that $f(z) \neq y$.

Having only a mistake oracle for $U$, rather than an explicit representation, is a more realistic form of access. In this case, the reduction algorithm has no explicit knowledge of the set of adversarial perturbations $U$ and is forced to interact with the mistake oracle $O_{U}$ in order to learn an adversarially robust predictor. Furthermore, what is typically referred to as adversarial training in practice fits exactly into this framework [Madry et al., 2018].
Does access to a mistake oracle $O_U$ suffice to robustly learn a target class $C$ using a black-box non-robust learner $A$ for $C$? First, can we achieve a similar upper bound as in Theorem 3.1 but with a mistake oracle $O_U$ rather than explicit access to $U$? Unfortunately, even with a black-box ERM for $C$, one can show that $|U|$ oracle calls to $O_U$ are unavoidable (proof provided in Appendix A):

**Claim 4.2.** For any reduction algorithm $B$, there exists an adversary $U$, target class $C$, and an ERM for $C$ with VC dimension 1, such that $B$ needs to make $T \geq |U|$ oracle calls to $O_U$.

Thus, a non-robust PAC learner $A$ for $C$ is not enough to learn $C$ robustly with a mistake oracle $O_U$ without a linear dependence on $|U|$. This suggests that a stronger assumption about $A$ is required.

We next show that an online learner $A$ for $C$ suffices to robustly learn $C$ with a mistake oracle $O_U$ and without any dependence on $|U|$. Before proceeding, we include a brief reminder of what it means to learn in an online setting with a finite mistake bound:

**Definition 4.3.** (Mistake Bound Model) We say an online learner $A$ learns a hypothesis class $C$ with mistake bound $M_A$ if learner $A$ makes at most $M_A$ mistakes on any sequence of examples that are labeled with some concept $c \in C$.

We are now ready to state our main result for this section.

**Theorem 4.4.** Algorithm 3 robustly PAC learns any target class $C$ w.r.t. an adversary $U$ with black-box access to a mistake oracle $O_U$ and an online learner $A$ for $C$ with sample complexity, number of calls to $A$, and number of calls to $O_U$ that is at most $2 \frac{M_A}{\varepsilon} \log \left( \frac{3M_A}{\delta} \right)$, where $M_A$ is the mistake-bound of online learner $A$.

**Proof sketch.** Run the online learner $A$ on the sequence of input examples using the mistake oracle $O_U$ to find mistakes. Details and algorithm are provided in Appendix A.

**Example 4.5.** Let $C$ be the class of OR functions over the boolean hypercube $\{0, 1\}^n$. There is an online learner $A$ that learns $C$ with a mistake bound $M_A = n$. Theorem 4.4 implies that we can robustly learn $C$ using $A$ with sample complexity, number of calls to $A$, and number of calls to $O_U$ that is at most $2 \frac{n}{\varepsilon} \log \left( \frac{2}{\delta} \right)$.

5 Discussion

The main contribution of this paper is in formulating the question of reducing adversarially robust learning to standard non-robust learning and providing answers in some settings. We outline a few directions for future work below.

**Mistake Oracle for $U$** This is a more challenging setting (but perhaps more realistic) where the reduction algorithm has no knowledge of $U$ and can only interact with a mistake oracle for $U$. Theorem 4.4 shows that online learnability is sufficient for robust learning in this model. Beyond this, are there weaker conditions that would enable robust learning under this model? Or is having an online learner essential? What if we consider specific target classes? Montasser et al. [2020] recently gave an algorithm that robustly learns halfspaces in this model. A natural next step is to ask which other classes can be robustly learned in this model, or more ambitiously characterize a necessary and sufficient condition for learning in this model.

**Agnostic Setting** We focused only on robust PAC learning in the realizable setting, where we assume there is a $c \in C$ with zero robust error. It would be desirable to extend our results also to the agnostic setting, where we want to compete with the best $c \in C$. We remark that an agnostic-to-realizable reduction described in Montasser et al. [2019, Theorem 6] can be used in our setting, however, it has runtime that is exponential in $\text{vc}(A)$. Another attempt through the agnostic boosting frameworks [e.g. Kalai and Kanade, 2009] requires a non-robust PAC learner $A$ with error $\varepsilon$ that scales with $|U|^2$, which results in a sample complexity that depends on $|U|$, and this is something we would like to avoid.

**Boosting and Robustness** Boosting has led to many exciting developments in theory and practice of machine learning. It started with asking: Can we boost the accuracy of weak predictors to attain a predictor with high accuracy? Freund and Schapire [1997] showed that boosting the accuracy is possible and can be done efficiently. What we consider in this paper can be viewed as a question of boosting robustness: Can we boost non-robust predictors to attain a robust predictor? and can we do this efficiently? Another natural question to consider which we did not study in this paper is: Can we boost weakly robust predictors to attain a robust predictor?
Broader Impact

Learning predictors that are robust to adversarial perturbations is an important challenge in contemporary machine learning. Current machine learning systems have been shown to be brittle against different notions of robustness such as adversarial perturbations [Szegedy et al., 2013, Biggio et al., 2013, Goodfellow et al., 2014], and there is an ongoing effort to devise methods for learning predictors that are adversarially robust. As machine learning systems become increasingly integrated into our everyday lives, it becomes crucial to provide guarantees about their performance, even when they are used outside their intended conditions.

We already have many tools developed for standard learning, and having a universal wrapper that can take any standard learning method and turn it into a robust learning method could greatly simplify the development and deployment of learning that is robust to test-time adversarial perturbations. The results that we present in this paper are still mostly theoretical, and limited to the realizable setting, but we expect and hope they will lead to further theoretical study as well as practical methodological development with direct impact on applications.

In this work we do not deal with training-time adversarial attacks, which is a major, though very different, concern in many cases.

As with any technology, having a more robust technology can have positive and negative societal consequences, and this depends mainly on how such technology is utilized. Our intent from this research is to help with the design of robust machine learning systems for application domains such as healthcare and transportation where its critical to ensure performance guarantees even outside intended conditions. In situations where there is a tradeoff between robustness and accuracy, this work might be harmful in that it would prioritize robustness over accuracy and this may not be ideal in some application domains.

Acknowledgments and Disclosure of Funding

We would like to thank Shay Moran for the insightful discussions that led to the formalization of the question we study in this paper. We also thank the anonymous reviewers for their thoughtful and helpful feedback. This work is partially supported by DARPA\footnote{This paper does not reflect the position or the policy of the Government, and no endorsement should be inferred.} cooperative agreement HR00112020003.

References


Daniel G Brown. How I wasted too long finding a concentration inequality for sums of geometric variables.


