Adversarially Robust Learning with Unknown Perturbation Sets

Omar Montasser OMAR@TTIC.EDU
Steve Hanneke STEVE.HANNEKE@GMAIL.COM
Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago, Chicago IL, USA

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

We study the problem of learning predictors that are robust to adversarial examples with respect to an unknown perturbation set, relying instead on interaction with an adversarial attacker or access to attack oracles, examining different models for such interactions. We obtain upper bounds on the sample complexity and upper and lower bounds on the number of required interactions, or number of successful attacks, in different interaction models, in terms of the VC and Littlestone dimensions of the hypothesis class of predictors, and without any assumptions on the perturbation set.

Keywords: adversarially robust PAC learning, unknown adversaries, sample and oracle complexity.

1. Introduction

We consider the problem of learning predictors that are *robust* to adversarial corruptions at test time. Given an instance space $\mathcal{X}$ and label space $\mathcal{Y} = \{\pm 1\}$, we would like to be robust against some *perturbation set* $\mathcal{U} : \mathcal{X} \to 2^{\mathcal{X}}$, where $\mathcal{U}(x) \subseteq \mathcal{X}$ represents the set of possible corruptions of x.

Almost all prior work on adversarial robustness starts with specifying a perturbation set $\mathcal{U}$ we would like to be robust against. The type of perturbation sets we are truly interested in are often sets $\mathcal{U}$ that capture "natural" or "imperecptible" perturbations. But partially because of the need to specify $\mathcal{U}$ explicitly during training, simpler sets are often used, such as ℓ_p-norm balls (Goodfellow et al., 2015), or orbits w.r.t. translations and rotations (Engstrom et al., 2019). Furthermore, training procedures are often specific to the perturbation set $\mathcal{U}$, or have the perturbation set "hard coded" inside them. Some methods rely on predictor implementations that need to "know" the specific perturbation set $\mathcal{U}$ at test-time (e.g., randomized smoothing Lécuyer et al., 2019; Cohen et al., 2019; Salman et al., 2019), and some methods use "explicit" knowledge of $\mathcal{U}$ only during training-time (e.g., Wong and Kolter, 2018; Raghunathan et al., 2018a,b; Montasser et al., 2019).

Main Question:

Can we design robust learning algorithms that do not require explicit knowledge of the adversarial perturbations $\mathcal{U}$?
What reasonable models of access to, or interactions with, $\mathcal{U}$ could we rely on instead?

In this paper, we ask whether it is possible to develop generic learning algorithms with robustness guarantees, without knowing the perturbation set $\mathcal{U}$ a-priori. That is, we want to design general robust algorithmic frameworks that work for any perturbation set $\mathcal{U}$, given a *reasonable* form of access to $\mathcal{U}$, and avoid algorithms tailored to a specific $\mathcal{U}$ such as ℓ_∞ or ℓ_2 perturbations. This is important if we want to be able to easily adapt our training procedures to different perturbation sets, or would like to build ML systems that are robust to fairly abstract perturbation sets $\mathcal{U}$ such as "images that are

	Sample Complexity	Oracle Complexity	
Realizable	$\tilde{O}(\text{lit}(\mathcal{H}))$	$\tilde{O}(\text{lit}(\mathcal{H}))$	Montasser et al. (2020b).
	$\tilde{O}(\text{vc}(\mathcal{H})\text{vc}^{*2}(\mathcal{H}))$	$2^{\tilde{O}(\text{vc}^2(\mathcal{H})\text{vc}^{*2}(\mathcal{H}))}\text{lit}(\mathcal{H})$	New result in this paper (Theorem 2).
		$\Omega(\text{lit}(\mathcal{H}))$	New result in this paper (Theorem 5).
Agnostic	$\tilde{O}(\text{lit}(\mathcal{H}))$	$\tilde{O}(\text{lit}^2(\mathcal{H}))$	New result in this paper (Theorem 3).
	$\tilde{O}(\text{vc}(\mathcal{H})\text{vc}^{*2}(\mathcal{H}))$	$2^{\tilde{O}(\text{vc}^2(\mathcal{H})\text{vc}^{*2}(\mathcal{H}))}\text{lit}(\mathcal{H})$	New result in this paper (Theorem 4).

TABLE 1: We show that a hypothesis class $\mathcal{H}$ is robustly learnable in the Perfect Attack Oracle model if and only if $\mathcal{H}$ is *online* learnable. We give upperbounds (corresponding to algorithms) in the realizable setting (Section 3.1) and the agnostic setting (Section 3.2), and lower bounds on the oracle complexity in the realizable setting (Section 3.3). Futhermore, our results show that sophisticated algorithms that leverage online learners can be favorable to more traditional online-to-batch conversion schemes in terms of their robust generalization guarantees. The $\tilde{O}$ notation hides logarithmic factors and dependence on error ε and failure probability δ, $\text{vc}(\mathcal{H})$ and $\text{vc}^*(\mathcal{H})$ denote the primal and dual VC dimension of $\mathcal{H}$, and $\text{lit}(\mathcal{H})$ denotes the Littlestone dimension of $\mathcal{H}$.

indistinguishable to the human eye" (see e.g., Laidlaw, Singla, and Feizi, 2020). In our frameworks, instead of redesigning or reprogramming the training algorithm, one would only need to implement or provide specific "attack procedures" for $\mathcal{U}$.

In this paper, we consider robustly learning a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ (e.g., neural networks). The learning algorithm receives as input m iid samples $S = \{(x_i, y_i)\}_{i=1}^m$ drawn from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. A predictor $h : \mathcal{X} \to \mathcal{Y}$ is robustly correct on an example (x, y) w.r.t. $\mathcal{U}$ if $\sup_{z \in \mathcal{U}(x)} \mathbb{1}[h(z) \neq y] = 0$. The learning algorithm has no *explicit* knowledge of $\mathcal{U}$, but instead is allowed the following forms of access:

Access to a (perfect) adversarial attack oracle In this model, the learning algorithm has access to a "mistake oracle", which we can also think of as a perfect attack oracle for $\mathcal{U}$. A perfect attack oracle for $\mathcal{U}$ receives as input a predictor $g : \mathcal{X} \to \mathcal{Y}$ and a labeled example (x, y), and is asked to either assert that g is robustly correct on (x, y), or return a perturbation $z \in \mathcal{U}(x)$ that is miss-classified (see Definition 4). The learning algorithm can query the perfect attack oracle for $\mathcal{U}$ by calling it T times with queries of the form: $(g_t, (x'_t, y'_t))$, where g_t is a predictor and (x'_t, y'_t) is a labeled example (not necessarily from the training set S). The goal of the learning algorithm is to output a predictor $\hat{h}$ with small *robust* risk

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

In Section 3, we present algorithms, guarantees on the required sample complexity and number of oracle accesses, and lower bounds on the required number of accesses, for robustly learning $\mathcal{H}$ in the Perfect Attack Oracle model. These results are summarized in Table 1.

To program such a perfect attack oracle, $\mathcal{U}$ still has to be specified inside it. And even for simple $\mathcal{U}$, a perfect attack oracle is generally intractable. Furthermore, practical attack engines used in training (e.g., PGD Madry et al., 2018) are not perfect, and are not always guaranteed to find miss-classified adversarial perturbations even when they do exist. Can we still provide meaningful robustness guarantees if we only have access to an *imperfect* attack oracle?

Access to an (imperfect) adversarial attack oracle In this model, the learning algorithm has access to a possibly imperfect attacking algorithm $\mathbb{A}$ for $\mathcal{U}$. The learning algorithm can query $\mathbb{A}$ by calling it T times with queries of the form: $(g_t, (x'_t, y'_t))$, where $g_t : \mathcal{X} \to \mathcal{Y}$ is a predictor and (x'_t, y'_t) is a labeled example. The goal of the learning algorithm is to output a predictor $\hat{h}$ with small error w.r.t. future attacks from $\mathbb{A}$,

$$\mathrm{err}_{\mathbb{A}}(\hat{h}; \mathcal{D}) \triangleq \Pr_{\substack{(x,y)\sim\mathcal{D} \\ \text{randomness of } \mathbb{A}}} \left[\hat{h}(\mathbb{A}(\hat{h}, (x, y))) \neq y\right]. \tag{2}$$

In Section 5, we give an algorithm with sample and oracle complexity of $\tilde{O}(\mathrm{lit}(\mathcal{H}))$ that guarantees small $\mathrm{err}_{\mathbb{A}}$ when then attacker $\mathbb{A}$ is "stationary", i.e., $\mathbb{A}$ doesn't learn or adapt over time.

But what happens if the adversary $\mathbb{A}$ changes over time? In the above model, the predictor $\hat{h}$ is fixed after training, and thus if the adversary $\mathbb{A}$ changes, e.g., by adapting to the returned predictor, or perhaps if we encounter an altogether different adversary than the one we accessed during training, this might result in a much higher error rate. Is it possible to continually adapt to changing adversaries in a meaningful way, ensuring strong robustness guarantees?

Interaction with an actual attacker In this online model, the learning algorithm $\mathcal{B}$ can monitor the behaviour of an actual attacker $\mathbb{A}$ and adapt accordingly. The attacker knows the current predictor h used, as well as the perturbation set $\mathcal{U}$, and attempts attacks on an iid stream of samples (x_t, y_t). Whenever the attacker succeeds in finding a perturbation $z_t \in \mathcal{U}(x_t)$ s.t. $h(x_t) \neq y_t$, it scores a "successful attack", but the perturbation z_t is revealed to learner $\mathcal{B}$, who can also obtain the true label y_t, and learner $\mathcal{B}$ can update its predictor. The goal of the learner is to bound the total number of successful attacks.

Monitoring and adapting to an attacker might sometimes be possible and appropriate, e.g., when attacks to predictors can be detected in hindsight and when the predictor is running on the cloud or when predictor updates can be pushed to devices, which is becoming increasingly common. But beyond such scenarios, this online model is also useful as an analysis tool of the imperfect attack oracle model above, and our methods for the imperfect attack oracle model are based on this online model.

In Section 4, we show upper bounds and lower bounds on the the number of successful attacks in terms of the Littlestone dimension $\mathrm{lit}(\mathcal{H})$, although our results leave open a possible exponential gap in the bound on the number of successful attacks (in a setting where the learner has access to infinitely many uncorrupted samples, i.e. knows the uncorrupted source distribution).

Practical Relevance Our goal is to understand how, from a theoretical perspective, it is possible to depart from assuming full and explicit knowledge of the perturbation set, and what types of other accesses and interactions could still enable adversarially robust learning. We obviously need some dependence on the perturbation set or possible attacks during training, and we are making the first steps in establishing what forms of access (beyond explicit exact knowledge) could be sufficient, and how they could be used, and what are the limits (lower bounds) on what might be possible using different access models. Some of the models above already capture approaches used in practice. But perhaps more importantly, we hope this study will lead to interest in defining "better" access models, and finding the "right" framework for adversarially robust learning, perhaps, by way of analogy, similar to how early studies of privacy in data analysis struggled with finding the "right" attack models and definitions.

Related Work Most prior work on adversarial robustness has focused on methods that are tailored to specific perturbation sets $\mathcal{U}$. For example, in randomized smoothing (Lécuyer et al., 2019; Cohen et al., 2019; Salman et al., 2019), computing a prediction on a test-point x requires sampling perturbations z from a distribution P over $\mathcal{U}(x)$, and returning the most likely prediction given by a learned predictor $f : \mathcal{X} \to \mathcal{Y}$. Distribution P is chosen based on $\mathcal{U}$, for example, if $\mathcal{U}$ is ℓ_2 perturbations, then P is an isotropic Gaussian distribution $\mathcal{N}(x, \sigma^2 I)$. In addition, there are algorithms (certified defenses) that minimize some surrogate loss $\ell_{\mathcal{U}}$ where the construction of $\ell_{\mathcal{U}}$ depends on $\mathcal{U}$ (e.g., Wong and Kolter, 2018; Raghunathan et al., 2018a,b).

The adversarial training framework (Goodfellow et al., 2015; Madry et al., 2018) does not use explicit knowledge of $\mathcal{U}$, but only uses an attacking algorithm (e.g., FGSM or PGD) implemented for the perturbation set $\mathcal{U}$. However, no formal guarantees are known about adversarial training in terms of robust generalization. Specifically, it is not known whether adversarial training will yield predictors that generalize to future adversarial perturbations from $\mathcal{U}$, or even generalize to specific perturbations chosen by PGD or FGSM. It has been observed that common forms of adversarial training on deep neural nets do not generalize to future attacks from PGD (Schmidt et al., 2018). Our work can be seen as a theoretical study of such generic approaches, which leads to different, and considerably more sophisticated methods (yet at this stage, perhaps not easily implementable).

Towards our quest in this paper for finding the right form of access to $\mathcal{U}$, we build on algorithms by Montasser et al. (2019) and Montasser et al. (2020b) by re-interpreting them in light of our questions, but also extending them significantly in the following ways: we avoid using a *robust* empirical risk minimization $\mathrm{RERM}_{\mathcal{U}}$ oracle that requires explicit knowledge of $\mathcal{U}$ as used in Montasser et al. (2019) and use an online learning algorithm instead, and we carry-out a technical inflation procedure of the training sample to include perturbations by utilizing a perfect attack oracle for $\mathcal{U}$ without explicit knowledge of $\mathcal{U}$ as was done in Montasser et al. (2019, 2020b). Furthermore, Montasser et al. (2020b) considered robustly PAC learning $\mathcal{H}$ using only black-box access to a non-robust PAC learner for $\mathcal{H}$ but allowed explicit knowledge of $\mathcal{U}$, their reduction makes oracle calls that depend on the bit complexity $\log |\mathcal{U}|$, and they show this is unavoidable. In this work, our algorithms can be viewed as black-box reductions that use an *online* learner for $\mathcal{H}$ (instead of just a PAC learner), furthermore, they do not require a explicit knowledge of $\mathcal{U}$ but only an attack oracle for $\mathcal{U}$. Our algorithms achieve the same sample complexity bound, but with number of calls to the online learner that is independent of $\log |\mathcal{U}|$ and only depends on the VC dimension $\mathrm{vc}(\mathcal{H})$.

Ashtiani et al. (2020) considered a weaker form of attacking algorithms – those that receive as input a black-box predictor – in a $\mathcal{U}$-specific learning framework and showed an upper bound of $\tilde{O}(\mathrm{vc}(\mathcal{H}))$ on the sample complexity of robust PAC learning when the pair $(\mathcal{H}, \mathcal{U})$ admits a query efficient attacking algorithm. Their learning algorithm relies on a *robust* empirical risk minimization $\mathrm{RERM}_{\mathcal{U}}$ oracle that requires explicit knowledge of $\mathcal{U}$. In this work, we focus on modularity and avoid using a RERM oracle for $\mathcal{H}$, and use a black-box online learner $\mathcal{A}$ for $\mathcal{H}$.

Goldwasser et al. (2020) considered classifying *arbitrary* test examples in a transductive selective classification setting. They gave an algorithm that takes as input: (a) training examples from a distribution P over $\mathcal{X}$ labeled with some unknown function h^* in a class $\mathcal{H}$ with finite VC dimension, and (b) a batch of arbitrary unlabeled test examples (possibly chosen by an unknown adversary), and outputs a selective predictor $\hat{f}$ – which abstains from predicting on some examples – that has a low rejection rate w.r.t. P, and low error rate on the test examples. Selective predictor $\hat{f}$, however, can potentially abstain from classifying most test examples if they are adversarial. In this paper, we consider classifying test examples $x \sim P$ or adversarial perturbations $z \in \mathcal{U}(x)$ where

the perturbation set $\mathcal{U}$ is unknown, and output predictors that do not abstain but always provide a classification with low error rate. We do not require unlabeled test examples, but require black-box access to an attack oracle for $\mathcal{U}$.

2. Preliminaries

Let $\mathcal{X}$ denote the instance space, $\mathcal{Y} = \{\pm 1\}$ denote the label space, and $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ denote a hypothesis class. We denote by $\mathrm{vc}(\mathcal{H})$ the VC dimension of $\mathcal{H}$. For a *dual space* $\mathcal{G}$: a set of functions $g_x : \mathcal{H} \to \mathcal{Y}$ defined as $g_x(h) = h(x)$, for each $h \in \mathcal{H}$ and each $x \in \mathcal{X}$, the dual VC dimesion of $\mathcal{H}$, denoted $\mathrm{vc}^*(\mathcal{H})$, is defined as the VC dimension of $\mathcal{G}$. The dual VC dimesion is known to satisfy: $\mathrm{vc}^*(\mathcal{H}) < 2^{\mathrm{vc}(\mathcal{H})+1}$ (Assouad, 1983). While this exponential dependence is tight for some classes, for many natural classes, such as linear predictors and some neural networks, the primal and dual VC dimensions are equal, or at least polynomially related. The dual VC dimension is utilized in the study of adversarially robust learning (Montasser et al., 2019), which we formally define next:

Definition 1 (Robust PAC Learnability) *Learner $\mathcal{B}$ (ε, δ)-robustly PAC learns $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with sample complexity $m(\varepsilon, \delta) : (0,1)^2 \to \mathbb{N}$ if for any perturbation set $\mathcal{U}$, any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, with probability at least $1 - \delta$ over $S \sim \mathcal{D}^{m(\varepsilon,\delta)}$: $\mathrm{R}_{\mathcal{U}}(\mathcal{B}(S); \mathcal{D}) \leq \inf_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; \mathcal{D}) + \epsilon$.*

Online Learnability and Littlestone Dimension An online learning algorithm $\mathcal{A}$ is a (measurable) map $(\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$. For a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, the mistake bound of $\mathcal{A}$ is the maximum possible number of mistakes algorithm $\mathcal{A}$ makes on any sequence of examples labeled with some $h \in \mathcal{H}$:

$$M(\mathcal{A}, \mathcal{H}) := \sup_{x_1, x_2, \cdots \in \mathcal{X}} \sup_{h \in \mathcal{H}} \sum_{t=1}^{\infty} \mathbb{1}\left[\mathcal{A}(\{(x_i, h(x_i))\}_{i=1}^{t-1})(x_t) \neq h(x_t)\right]. \tag{3}$$

We say that a class $\mathcal{H}$ is online learnable if there exists an online learning algorithm $\mathcal{A}$ such that $M(\mathcal{A}, \mathcal{H}) < \infty$. A class $\mathcal{H}$ is online learnable if and only if the Littlestone dimension of $\mathcal{H}$ denoted $\mathrm{lit}(\mathcal{H})$ is finite (Littlestone, 1987). Furthermore, Littlestone (1987) proposed the Standard Optimal Algorithm (SOA) and showed that $M(\mathsf{SOA}, \mathcal{H}) \leq \mathrm{lit}(\mathcal{H})$. We now briefly recall the definition of Littlestone dimension by introducing the notion of Littlestone trees:

Definition 2 (Littlestone trees) *A Littlestone tree for $\mathcal{H}$ is a complete binary tree of depth $d \leq \infty$ whose internal nodes are labeled by instances from $\mathcal{X}$, and whose two edges connecting a node to its children are labeled with $+1$ and -1 such that every finite path emanating from the root is consistent with some concept in $\mathcal{H}$. That is, a Littlestone tree is a collection $\left\{x_{\boldsymbol{u}} : 0 \leq k < d, \boldsymbol{u} \in \{\pm 1\}^k\right\} \subseteq \mathcal{X}$ such that for every $\boldsymbol{y} \in \{\pm 1\}^d$, there exists $h \in \mathcal{H}$ such that $h(x_{\boldsymbol{y}_{1:k}}) = y_{k+1}$ for $0 \leq k < d$.*

Definition 3 (Littlestone dimension) *The Littlestone dimension of $\mathcal{H}$, denoted $\mathrm{lit}(\mathcal{H})$, is the largest integer d such that there exists a Littlestone tree for $\mathcal{H}$ of depth d (see Definition 2). If no such d exists, then $\mathrm{lit}(\mathcal{H})$ is said to be infinite.*

3. Access to a Perfect Attack Oracle

In this section, we study robust learning with algorithms that are only allowed access to a perfect attack oracle for $\mathcal{U}$ at training-time. Formally,

Definition 4 (Perfect Attack Oracle) *Denote by $\mathsf{O}_{\mathcal{U}}$ a perfect attack oracle for $\mathcal{U}$. $\mathsf{O}_{\mathcal{U}}$ takes as input a predictor $f : \mathcal{X} \to \mathcal{Y}$ and an example $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and either: (a) asserts that f is robust on (x, y) (i.e. $\forall z \in \mathcal{U}(x), f(z) = y$), or (b) returns a perturbation $z \in \mathcal{U}(x)$ such that $f(z) \neq y$.*[1]

In the Perfect Attack Oracle model, a learning algorithm $\mathcal{B}$ takes as input iid distributed training samples $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ drawn from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, and a black-box perfect attack oracle $\mathsf{O}_{\mathcal{U}}$. Learner $\mathcal{B}$ can query $\mathsf{O}_{\mathcal{U}}$ by calling it T times with queries of the form: $(g_t, (x'_t, y'_t))$, where $g_t : \mathcal{X} \to \mathcal{Y}$ is a predictor and (x'_t, y'_t) is a labeled example. The goal of learner $\mathcal{B}$ is to output a predictor $\hat{h} \in \mathcal{Y}^{\mathcal{X}}$ with small robust risk $\mathrm{R}_{\mathcal{U}}(\hat{h}; \mathcal{D}) \leq \varepsilon$ (see Equation 1). Learner $\mathcal{B}$ (ε, δ)-robustly PAC learns $\mathcal{H}$ in the Perfect Attack Oracle model with oracle complexity $T(\varepsilon, \delta)$ if for any perturbation set $\mathcal{U}$, learner $\mathcal{B}$ (ε, δ)-robustly PAC learns $\mathcal{H}$ with at most $T(\varepsilon, \delta)$ calls to $\mathsf{O}_{\mathcal{U}}$.

From a practical or engineering perspective, to establish robust generalization guarantees with respect to $\mathcal{U}$ in the Perfect Attack Oracle model, it suffices to build a perfect attack oracle for $\mathcal{U}$. Furthermore, to achieve robustness guarantees to multiple perturbation sets $\mathcal{U}_1, \ldots, \mathcal{U}_k$ concurrently, which is a goal of interest in practice (see e.g., Kang et al., 2019; Tramèr and Boneh, 2019; Maini et al., 2020), it suffices to *separately* build perfect attack oracles $\mathsf{O}_{\mathcal{U}_1}, \ldots, \mathsf{O}_{\mathcal{U}_k}$, and then implement a perfect attack oracle for the union $\cup_{i \leq k} \mathcal{U}_i$ by calling each attack oracle $\mathsf{O}_{\mathcal{U}_1}, \ldots, \mathsf{O}_{\mathcal{U}_k}$ separately.

Questions:

What hypothesis classes $\mathcal{H}$ are robustly PAC learnable in the Perfect Attack Oracle model?
How can we learn a generic $\mathcal{H}$ using such access?
With how many samples m and oracle calls T?

Summary of Results We begin in Section 3.1 with the realizable setting under which it is assumed that there is a predictor $h^* \in \mathcal{H}$ with zero robust risk, i.e. $\inf_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; \mathcal{D}) = 0$. In Theorem 1, we give a simple algorithm (Algorithm 1) CycleRobust that robustly learns $\mathcal{H}$ in the Perfect Attack Oracle model with sample complexity $m = O(\mathrm{lit}(\mathcal{H}))$ and oracle complexity $T = O(\mathrm{lit}^2(\mathcal{H}))$. In Theorem 2, we give an alternative algorithm (Algorithm 1) RLUA to robustly learn $\mathcal{H}$ in the Perfect Attack Oracle model with reduced sample complexity $m = \tilde{O}(\mathrm{vc}(\mathcal{H})\mathrm{vc}^{*2}(\mathcal{H}))$ depending only the VC and dual VC dimension but at the cost of higher oracle complexity $T \approx 2^{\tilde{O}(\mathrm{vc}^2(\mathcal{H})\mathrm{vc}^{*2}(\mathcal{H}))}\mathrm{lit}(\mathcal{H})$. Then, in Section 3.2, we extend our algorithmic results in Theorem 3 and Theorem 4 to the more general agnostic setting where we want to compete with the best attainable robust risk $\inf_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; \mathcal{D})$. Finally, in Section 3.3, we give a lower bound on the oracle complexity necessary to robustly learn in the Perfect Attack Oracle model. In Corollary 6, we show that for any class $\mathcal{H}$, the oracle complexity to robustly learn $\mathcal{H}$ is at least $\Omega(\log\log \mathrm{lit}(\mathcal{H}))$. Furthermore, Corollary 7 gives a specific hypothesis class $\mathcal{H}$ with $\mathrm{vc}(\mathcal{H}) = O(1) \ll \mathrm{lit}(\mathcal{H})$ such that the oracle complexity to robustly learn $\mathcal{H}$ is at least $\Omega(\mathrm{lit}(\mathcal{H}))$. These results are summarized in Table 1 on page 2.

1. To be clear, we suppose $\mathsf{O}_{\mathcal{U}}$ acts as a function so that the z it returns from calling $\mathsf{O}_{\mathcal{U}}(g, (x, y))$ is deterministic and oblivious to the history of interactions.

Related Work Montasser et al. (2020a) recently gave an algorithm based on the Ellipsoid method to efficiently robustly learn halfspaces (linear predictors) in the Perfect Attack Oracle model in the realizable setting, for a broad range of perturbation sets $\mathcal{U}$ given access to a separation oracle for $\mathcal{U}$, with oracle complexity that depends on the bit complexity. Furthermore, using a conservative online learner $\mathcal{A}$ for $\mathcal{H}$, Montasser et al. (2020b) gave an algorithm to robustly PAC learn $\mathcal{H}$ in the Perfect Attack Oracle model in the realizable setting with sample complexity $m(\varepsilon, \delta)$ and oracle complexity $T(\varepsilon, \delta)$ at most $O\left(\frac{\mathrm{lit}(\mathcal{H})\log(\mathrm{lit}(\mathcal{H})/\delta)}{\varepsilon}\right)$. We consider robustly learning a general class $\mathcal{H}$, and the more general agnostic setting.

3.1. Algorithms and guarantees in the realizable setting

We begin in Theorem 1 with a simple algorithm CycleRobust (Algorithm 1) based on an online-to-batch conversion that robustly PAC learns a class $\mathcal{H}$ with sample complexity and oracle complexity depending on the Littlestone dimension $\mathrm{lit}(\mathcal{H})$. Specifically, CycleRobust (Algorithm 1) cycles an online learner $\mathcal{A}$ for $\mathcal{H}$ on the training set S until it robustly correctly classifies all training examples. To establish a robust generalization guarantee, we show that CycleRobust (Algorithm 1) can be viewed as a stable compression scheme for the robust loss. This conversion technique and its connection to stable sample compression schemes have been recently studied in the standard 0-1 loss setting (Bousquet et al., 2020). The proof is provided in Appendix A.

Theorem 1 *For any class $\mathcal{H}$,* CycleRobust *(Algorithm 1) robustly PAC learns $\mathcal{H}$ w.r.t. any $\mathcal{U}$ with:*

1. *Sample complexity* $m(\varepsilon, \delta) = O\left(\frac{\mathrm{lit}(\mathcal{H})+\log(1/\delta)}{\varepsilon}\right)$.
2. *Oracle complexity* $T(\varepsilon, \delta) = m(\varepsilon, \delta)\mathrm{lit}(\mathcal{H})$.

Furthermore, the output of CycleRobust *achieves zero robust loss on the training sample.*

CycleRobust (Algorithm 1) robustly PAC learns $\mathcal{H}$ in the Perfect Attack Oracle model with sample complexity and oracle complexity both depending on the Littlestone dimension $\mathrm{lit}(\mathcal{H})$. But are there robust learning algorithms with better sample complexity and/or oracle complexity? At least with explicit knowledge of $\mathcal{U}$, we know that we can robustly PAC learn $\mathcal{H}$ with $\tilde{O}(\mathrm{vc}(\mathcal{H})\mathrm{vc}^*(\mathcal{H}))$ sample complexity (Montasser et al., 2019) which is much smaller than $\mathrm{lit}(\mathcal{H})$ for many natural classes (e.g., halfspaces). Can we obtain a similar sample complexity bound in the Perfect Attack Oracle model, where explicit knowledge of $\mathcal{U}$ is not allowed? We prove *yes* in Theorem 2. Specifically, we give an algorithm that can robustly PAC learn $\mathcal{H}$ in Perfect Attack Oracle model with sample complexity $\tilde{O}(\mathrm{vc}(\mathcal{H})\mathrm{vc}^{*2}(\mathcal{H}))$ independent of $\mathrm{lit}(\mathcal{H})$.

Theorem 2 *For any class $\mathcal{H}$ with $\mathrm{vc}(\mathcal{H}) = d$ and $\mathrm{vc}^*(\mathcal{H}) = d^*$, there exists a learning algorithm $\tilde{\mathcal{B}}$ that robustly PAC learns $\mathcal{H}$ w.r.t any $\mathcal{U}$ with:*

1. *Sample Complexity* $m(\varepsilon, \delta) = O\left(\frac{dd^{*2}\log^2 d^*}{\varepsilon}\log^2\left(\frac{dd^{*2}\log^2 d^*}{\varepsilon}\right) + \frac{1}{\varepsilon}\log(\frac{2}{\delta})\right)$.
2. *Oracle Complexity* $T_{\mathrm{RE}}(\varepsilon, \delta) = \left(2^{O(d^2d^{*2}\log^2 d^*)}\mathrm{lit}(\mathcal{H}) + m(\varepsilon, \delta)\right)\left(\log\left(m(\varepsilon, \delta)\right)\left(\log\left(\frac{\log(m(\varepsilon,\delta))}{\delta}\right)\right)\right)$.

The full proof is deferred to Appendix B, but we briefly describe the main building blocks of this result. We will adapt an algorithm due to Montasser et al. (2019) and establish a robust generalization

guarantee that depends only on $\mathrm{vc}(\mathcal{H})$ and $\mathrm{vc}^*(\mathcal{H})$. In particular, the learning algorithm of Montasser et al. (2019) required explicit knowledge of $\mathcal{U}$, this knowledge was used to implement a $\mathrm{RERM}_{\mathcal{U}}$ oracle for $\mathcal{H}$, and for a sample inflation and discretization step which is crucial to establish robust generalization based on sample compression. As explicit knowledge is not allowed in the Perfect Attack Oracle model, we show that we can avoid these limitations and use only queries to $\mathsf{O}_{\mathcal{U}}$. Specifically, observe that CycleRobust (Algorithm 1) implements a $\mathrm{RERM}_{\mathcal{U}}$ oracle for $\mathcal{H}$ using only black-box access to $\mathsf{O}_{\mathcal{U}}$, since by Theorem 1 the output of CycleRobust achieves zero robust loss on its input dataset S. Similarly, the discretization step can be carried using only queries to $\mathsf{O}_{\mathcal{U}}$, by constructing queries using the output predictors of CycleRobust to force the oracle to reveal perturbations of the empirical sample S, we leave further details to the proof. While this suffices to establish a result of robust PAC learning in the Perfect Attack Oracle model with sample complexity completely independent of $\mathrm{lit}(\mathcal{H})$ (Theorem 18 and its proof in Appendix B), we can further improve the dependence on ε and δ in the oracle complexity. To this end, we treat the algorithm (Algorithm 1) RLUA from Theorem 18 as a *weak* robust learner with fixed ε_0 and δ_0 and boost its robust error guarantee to improve the oracle complexity and obtain the result in Theorem 2.

3.2. Algorithms and guarantees in the agnostic setting

We now consider the more general agnostic setting where we want to compete with the best attainable robust risk $\inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U}}(h;\mathcal{D})$. Mirroring the results from the realizable section, we begin in Theorem 3 with a simple algorithm that can only guarantee a robust error at most $2\inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U}}(h;\mathcal{D})+\varepsilon$ with sample and oracle complexity depending on the Littlestone dimension $\mathrm{lit}(\mathcal{H})$. Then, in Theorem 4, we give a reduction to the realizable setting of Theorem 2, that agnostically robustly PAC learns $\mathcal{H}$ in the Perfect Attack Oracle model with sample complexity depending only on the $\mathrm{vc}(\mathcal{H})$ and $\mathrm{vc}^*(\mathcal{H})$.

Theorem 3 *For any class $\mathcal{H}$,* Weighted Majority *(Algorithm 2) guarantees that for any perturbation set $\mathcal{U}$ and any distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$, with sample complexity $m(\varepsilon,\delta)=O\left(\frac{\mathrm{lit}(\mathcal{H})\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon^2}\right)$ and oracle complexity $T(\varepsilon,\delta)=O(m(\varepsilon,\delta)^2)$, with probability at least $1-\delta$ over $S\sim\mathcal{D}^{m(\varepsilon,\delta)}$,*

$$\mathrm{R}_{\mathcal{U}}(\mathsf{WM}(S,\mathsf{O}_{\mathcal{U}});\mathcal{D}) \le 2\inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U}}(h;\mathcal{D})+\varepsilon.$$

We briefly describe the main ingredients of this result. First, in Lemma 19, we show that for any class $\mathcal{H}$ with finite cardinality, a variant of the Weighted Majority algorithm (Littlestone and Warmuth, 1994) presented in (Algorithm 2) has a regret guarantee with respect to the robust loss. Then, in Lemma 20, we extend this regret guarantee for infinite $\mathcal{H}$ using a technique due to Ben-David et al. (2009) for agnostic online learning. Finally, we apply a standard online-to-batch conversion (Cesa-Bianchi et al., 2004) to convert the regret guarantee to a robust generalization guarantee. These helper lemmas and proofs are deferred to Appendix C.

Similarly to the realizable setting, we can establish an upper bound with sample complexity independent of $\mathrm{lit}(\mathcal{H})$. This is achieved via a reduction to the realizable setting Theorem 2, following an argument of David et al. (2016); Montasser et al. (2019). The proof is deferred to Appendix C.

Theorem 4 (Reduction to Realizable Setting) *For any class $\mathcal{H}$ with $\mathrm{vc}(\mathcal{H})=d$ and $\mathrm{vc}^*(\mathcal{H})=d^*$, there is a learning algorithm $\tilde{\mathcal{B}}$ that robustly* agnostically *PAC learns $\mathcal{H}$ w.r.t any $\mathcal{U}$ with:*

1. *Sample Complexity* $m(\varepsilon,\delta)=O\left(\frac{dd^{*2}\log^2 d^*}{\varepsilon^2}\log^2\left(\frac{dd^{*2}\log^2 d^*}{\varepsilon}\right)+\frac{1}{\varepsilon^2}\log(\frac{2}{\delta})\right)$.
2. *Oracle Complexity* $T(\varepsilon,\delta)=2^{m(\varepsilon,\delta)}\mathrm{lit}(\mathcal{H})+T_{\mathrm{RE}}(\varepsilon,\delta)$.

3.3. Lowerbound on Oracle Complexity

In Section 3.1 and Section 3.2, we have shown that it is possible to robustly PAC learn in the Perfect Attack Oracle model with sample complexity that is completely *independent* of the Littlestone dimension, and with oracle complexity that *depends* on the Littlestone dimension. It is natural to ask whether the oracle complexity can be improved. Perhaps, we can avoid dependence on Littlestone dimension altogether? In Theorem 5, we prove that the answer is *no*.

Specifically, we will first establish a lower bound in terms of another complexity measure known as the Threshold dimension of $\mathcal{H}$, denoted by $\mathrm{Tdim}(\mathcal{H})$. Informally, $\mathrm{Tdim}(\mathcal{H})$ is the largest number of thresholds that can be embedded in class $\mathcal{H}$ (see Definition 5 below). Importantly, the Threshold dimension of $\mathcal{H}$ is related to the Littlestone dimension of $\mathcal{H}$ and is known to satisfy: $\lfloor \log_2 \mathrm{lit}(\mathcal{H}) \rfloor \leq \mathrm{Tdim}(\mathcal{H}) \leq 2^{\mathrm{lit}(\mathcal{H})}$ (Shelah, 1990; Hodges, 1997; Alon et al., 2019). This relationship was recently used to establish that *private* PAC learnability implies online learnability (Alon et al., 2019).

Definition 5 (Threshold Dimension) *We say that a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ contains k thresholds if $\exists x_1, \ldots, x_k \in \mathcal{X}$ and $\exists h_1, \ldots, h_k \in \mathcal{H}$ such that $h_i(x_j) = +1$ if and only if $j \leq i, \forall i, j \leq k$. The Threshold dimension of $\mathcal{H}$, $\mathrm{Tdim}(\mathcal{H})$, is the largest integer k such that $\mathcal{H}$ contains k thresholds.*

Theorem 5 *For any class $\mathcal{H}$, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, such that for any learner $\mathcal{B}$, there exists a perturbation set $\mathcal{U} : \mathcal{X} \to 2^{\mathcal{X}}$ where $\inf_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; \mathcal{D}) = 0$ and a perfect attack oracle $\mathsf{O}_{\mathcal{U}}$ such that $\mathcal{B}$ needs to make $\frac{\log_2(\mathrm{Tdim}(\mathcal{H})-1)}{2}$ oracle queries to $\mathsf{O}_{\mathcal{U}}$ to robustly learn $\mathcal{D}$.*

The full proof is deferred to Appendix D, but we will provide some intuition behind the proof. The main idea is to use $h_1, \ldots, h_{\mathrm{Tdim}(\mathcal{H})}$ thresholds to construct $\mathcal{U}_1, \ldots, \mathcal{U}_{\mathrm{Tdim}(\mathcal{H})}$ perturbation sets. We will setup a single source distribution $\mathcal{D}$ that is known to the learner, but choose a random perturbation set from $\mathcal{U}_1, \ldots, \mathcal{U}_{\mathrm{Tdim}(\mathcal{H})}$. In order for the learner to *robustly* learn $\mathcal{D}$, it needs to figure out which perturbation set is picked, and that requires $\Omega(\log \mathrm{Tdim}(\mathcal{H}))$ queries to the oracle $\mathsf{O}_{\mathcal{U}}$. Since $\mathrm{Tdim}(\mathcal{H}) \geq \lfloor \log_2 \mathrm{lit}(\mathcal{H}) \rfloor$, Theorem 5 implies the following corollaries.

Corollary 6 *For any class $\mathcal{H}$, the oracle complexity to robustly learn $\mathcal{H}$ in the Perfect Attack Oracle model is at least $\Omega(\log \log \mathrm{lit}(\mathcal{H}))$.*

Corollary 7 *For any $n \in \mathbb{N}$, the class $\mathcal{H}_n$ consisting of n thresholds satisfies $\mathrm{lit}(\mathcal{H}_n) = \log_2 \mathrm{Tdim}(\mathcal{H}_n) = \log_2 n$[2] and $\mathrm{vc}(\mathcal{H}_n) = O(1)$. Thus, the oracle complexity to robustly learn $\mathcal{H}_n$ in the Perfect Attack Oracle model is $\Omega(\mathrm{lit}(\mathcal{H}_n))$.*

A couple of remarks are in order. First, the lower bound of $\Omega(\log \log \mathrm{lit}(\mathcal{H}))$ applies to any hypothesis class $\mathcal{H}$, but a stronger lower bound for the special case of thresholds can be shown where $\Omega(\mathrm{lit}(\mathcal{H}))$ oracle queries are needed. Second, observe that these lower bounds apply to learning algorithms that know the distribution $\mathcal{D}$, and so even with infinite sample complexity, it is not possible to have oracle complexity independent of Littlestone dimension.

2. We learned about this fact from the following talk: https://youtu.be/NPpPiWYcmPk

3.4. Gaps and Open Questions

We have established that $\mathcal{H}$ is robustly PAC learnable in the Perfect Attack Oracle model if and only if $\mathcal{H}$ is online learnable. We provided a simple online-to-batch conversion scheme CycleRobust (Algorithm 1) with sample and oracle complexity scaling with $\mathrm{lit}(\mathcal{H})$. Then, with a more sophisticated algorithm, RLUA (Algorithm 1), we get an improved sample complexity depending only on $\mathrm{vc}(\mathcal{H})$ and $\mathrm{vc}^*(\mathcal{H})$, but at the expense of higher oracle complexity with an exponential dependence on $\mathrm{vc}(\mathcal{H})$ and $\mathrm{vc}^*(\mathcal{H})$ and linear dependence on $\mathrm{lit}(\mathcal{H})$. We also showed that for any class $\mathcal{H}$, an oracle complexity of $\log\log\mathrm{lit}(\mathcal{H})$ is unavoidable, and furthermore, exhibit a class $\mathcal{H}$ with $\mathrm{vc}(\mathcal{H}) = O(1)$ and $\mathrm{lit}(\mathcal{H}) \gg \mathrm{vc}(\mathcal{H})$ where an oracle complexity of $\Omega(\mathrm{lit}(\mathcal{H}))$ is unavoidable.

An interesting direction is to improve the oracle complexity to perhaps a polynomial dependence $\mathrm{poly}(\mathrm{vc}(\mathcal{H}), \mathrm{vc}^*(\mathcal{H}))\mathrm{lit}(\mathcal{H})$, or more ambitiously $\mathrm{poly}(\mathrm{vc}(\mathcal{H}))\mathrm{lit}(\mathcal{H})$. It would also be interesting to establish a finer characterization for the oracle complexity that is adaptive to the perturbation sets $\mathcal{U}$, perhaps depending on some notion measuring the complexity of $\mathcal{U}$. Also, can we strengthen the lower bound and show that for any $\mathcal{H}$, $\Omega(\mathrm{lit}(\mathcal{H}))$ oracle complexity is necessary to robustly learn $\mathcal{H}$ or is there another complexity measure that tightly characterizes the oracle complexity?

4. Bounding the number of successful attacks

In Section 3, we considered having access to a *perfect* attack oracle $\mathsf{O}_{\mathcal{U}}$. But in many settings, our practical attack oracle attack engines, e.g., PGD (Madry et al., 2018), are not perfect—they might not always find miss-classified adversarial perturbations even when they do exist. Also, the perturbation set $\mathcal{U}$ might be fairly abstract, like "images indistinguishable to the human eye", and so there isn't really a perfect attack oracle, but rather just approximations to it. Can we still provide meaningful robustness guarantees even with *imperfect* attackers?

In this section, we introduce a model where we consider working with an actual adversary or attack algorithm that is possibly imperfect, and the goal is to bound the number of successful attacks. In this model, a learning algorithm $\mathcal{B}$ first receives as input iid distributed training samples $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ drawn from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Then, the learning algorithm $\mathcal{B}$ makes predictions on examples $z_t \in \mathcal{U}(x'_t)$ where z_t is an adversarial perturbation chosen by an adversary $\mathbb{A}$, and (x'_t, y'_t) is an iid sample drawn from $\mathcal{D}$. The adversary $\mathbb{A}$ has access to the random sample (x'_t, y'_t) and the predictor used by learner $\mathcal{B}$, $h_t = \mathcal{B}(S \cup \{(z_j, y_j)_{j=1}^{t-1}\})$, but learner $\mathcal{B}$ only sees the perturbation z_t. After learner $\mathcal{B}$ makes its prediction $\hat{y}_t = h_t(z_t)$, the true label y_t is revealed to $\mathcal{B}$. The goal is to bound the number of successful adversarial attacks where $\hat{y}_t \neq y_t$. For a class $\mathcal{H}$ and a learner $\mathcal{B}$, the maximum number of successful attacks caused by an adversary $\mathbb{A}$ w.r.t. $\mathcal{U}$ on a distribution $\mathcal{D}$ satisfying $\inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; \mathcal{D}) = 0$ is defined as

$$M_{\mathcal{U},\mathbb{A}}(\mathcal{B}, \mathcal{H}; \mathcal{D}) := \sum_{t=1}^{\infty} \mathbb{1}\left[\mathcal{B}(S \cup \{(z_i, y_i)\}_{i=1}^{t-1})(z_t) \neq y_t\right], \tag{4}$$

where $z_t = \mathbb{A}(\mathcal{B}(S \cup \{(z_i, y_i)\}_{i=1}^{t-1}), (x'_t, y'_t))$ and $\{(x'_t, y'_t)\}_{t=1}^{\infty}$ are iid samples from $\mathcal{D}$.

Questions:

Can we obtain upper bounds and lower bounds on the maximum number of successful attacks for generic classes $\mathcal{H}$? Can additional training samples from $\mathcal{D}$ help?

First, we show that we can upper bound the maximum number of successful attacks on any online learner $\mathcal{B}$ for $\mathcal{H}$ by the online mistake bound of $\mathcal{B}$.

Theorem 8 (Upper Bound) *For any class $\mathcal{H}$ and any online learner $\mathcal{B}$, for any perturbation set $\mathcal{U}$, adversary $\mathbb{A}$, and distribution $\mathcal{D}$, $M_{\mathcal{U},\mathbb{A}}(\mathcal{B},\mathcal{H};\mathcal{D}) \leq M(\mathcal{B},\mathcal{H})$. In particular, the Standard Optimal Algorithm (*SOA*) has an attack bound of at most $\operatorname{lit}(\mathcal{H})$.*

Proof The proof follows directly from the definition of the online mistake bound (see Equation 3) and the online attack bound (see Equation 4). ■

Is this the best achievable upper bound on the number of successful attacks? Perhaps there are learning algorithms with an attack bound that is much smaller than $\operatorname{lit}(\mathcal{H})$, maybe an attack bound that scales with $\operatorname{vc}(\mathcal{H})$? We next answer this question in the negative. Using the same the lower bound construction from Theorem 5 in Section 3.3, we first establish a lower bound on the number of the successful attacks based on the Threshold dimension of $\mathcal{H}$ (see Definition 5) (proof deferred to Appendix E). We then utilize the relationship $\operatorname{Tdim}(\mathcal{H}) \geq \lfloor \log_2 \operatorname{lit}(\mathcal{H}) \rfloor$ to get the corollaries.

Theorem 9 (Lower Bound) *For any class $\mathcal{H}$, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, such that for any learner $\mathcal{B}$, there is a perturbation set $\mathcal{U}$ where $\inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U}}(h;\mathcal{D}) = 0$ and an adversary $\mathbb{A}$ that makes at least $\frac{\log_2(\operatorname{Tdim}(\mathcal{H})-1)}{2}$ successful attacks on learner $\mathcal{B}$.*

Corollary 10 *For any class $\mathcal{H}$, there is a distribution $\mathcal{D}$, such that for any learner $\mathcal{B}$, there is a perturbation set $\mathcal{U}$ and adversary $\mathbb{A}$ such that $M_{\mathcal{U},\mathbb{A}}(\mathcal{B},\mathcal{H};\mathcal{D}) \geq \Omega(\log\log\operatorname{lit}(\mathcal{H}))$.*

Corollary 11 *For any $n \in \mathbb{N}$, the class $\mathcal{H}_n$ consisting of n thresholds satisfies $\operatorname{lit}(\mathcal{H}_n) = \log_2 \operatorname{Tdim}(\mathcal{H}_n) = \log_2 n$ and $\operatorname{vc}(\mathcal{H}_n) = O(1)$. Thus, $\exists_{\mathcal{D}}\forall_{\mathcal{B}}\exists_{\mathcal{U},\mathbb{A}} M_{\mathcal{U},\mathbb{A}}(\mathcal{B},\mathcal{H}_n;\mathcal{D}) \geq \Omega(\operatorname{lit}(\mathcal{H}_n))$.*

We remark that these lower bounds hold even for learning algorithms $\mathcal{B}$ that perfectly know the source distribution $\mathcal{D}$. For the class $\mathcal{H}_n$ of n thresholds, we cannot expect a learning algorithm that leverages extra training data that avoids the $\Omega(\operatorname{lit}(\mathcal{H}_n))$ lower bound. But it might be that for other classes $\mathcal{H}$, additional training data might help reduce the attack bound to $\log\operatorname{lit}(\mathcal{H})$ or $\log\log\operatorname{lit}(\mathcal{H})$.

Gaps and Open Questions We have only considered the realizable setting, where there is a predictor $h \in \mathcal{H}$ that is perfectly robust to the attacker $\mathbb{A}$. It would be interesting to extend the guarantees to the agnostic setting. Can we strengthen the lower bound and show that for any $\mathcal{H}$, an attack bound of $\Omega(\operatorname{lit}(\mathcal{H}))$ is unavoidable or is there another complexity measure that tightly characterizes the attack bound? Are there examples of classes $\mathcal{H}$ where collecting additional samples from $\mathcal{D}$ helps reduce the number of successful attacks?

5. Robust generalization to imperfect attack algorithms

In Section 4, given an online learning algorithm, we can guarantee a finite number of successful attacks from any attacking algorithm even if it was imperfect. But what if we want to work in a more traditional train-then-ship approach, where we first ensure adversarial robustness without releasing anything, and only then release? Can we provide any robust generalization guarantees when we only have access to an *imperfect* attacking algorithm such as PGD (Madry et al., 2018) at training-time?

In this model, a learning algorithm $\mathcal{B}$ takes as input a black-box (possibly imperfect) attacker $\mathbb{A}$, and iid distributed training samples $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Learner $\mathcal{B}$ can query $\mathbb{A}$ by calling it T times with queries of the form: $(g_t, (x'_t, y'_t))$, where $g_t : \mathcal{X} \to \mathcal{Y}$ is a predictor and (x'_t, y'_t) is a labeled example. The goal of learner $\mathcal{B}$ is to output a predictor $\hat{h} \in \mathcal{Y}^{\mathcal{X}}$ with small error w.r.t. future attacks from $\mathbb{A}$, $\mathrm{err}_{\mathbb{A}}(\hat{h}; \mathcal{D}) \leq \varepsilon$ (see Equation 2).

Definition 6 (Robust PAC Learnability with Imperfect Attackers) *Learner $\mathcal{B}$ (ε, δ)-robustly PAC learns $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with sample complexity $m(\varepsilon, \delta) : (0,1)^2 \to \mathbb{N}$ and oracle complexity $T(\varepsilon, \delta) : (0,1)^2 \to \mathbb{N}$ if for any (possibly randomized and imperfect) attacker $\mathbb{A} : \mathcal{Y}^{\mathcal{X}} \times (\mathcal{X} \times \mathcal{Y}) \to \mathcal{X}$, any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, with at most $T(\varepsilon, \delta)$ oracle calls to $\mathbb{A}$ and with probability at least $1 - \delta$ over $S \sim \mathcal{D}^{m(\varepsilon,\delta)}$: $\mathrm{err}_{\mathbb{A}}(\mathcal{B}(S, \mathbb{A})) \leq \inf_{h \in \mathcal{H}} \mathrm{err}_{\mathbb{A}}(h) + \epsilon$.*

In this model, access to an imperfect attacker $\mathbb{A}$ at training-time ensures robust generalization to this *specific* attacker $\mathbb{A}$ at test-time. This is a different guarantee from robust generalization to a perturbation set $\mathcal{U}$, because it might well be that that there is a stronger attack algorithm $\mathbb{A}'$ than $\mathbb{A}$ such that $\mathrm{err}_{\mathbb{A}'}(\hat{h}; \mathcal{D}) \gg \mathrm{err}_{\mathbb{A}}(\hat{h}; \mathcal{D})$. Furthermore, since the "strength" of the attack algorithm $\mathbb{A}$ is a function of the predictor $\hat{h}$ it is attacking, establishing a generalization guarantee w.r.t. $\mathbb{A}$ is not immediate, and does not follow for example from our results in Section 3.

We relate robust learnability in this model to the online model from Section 4. In Theorem 12, we observe that we can apply a standard online-to-batch conversion based on the longest survivor technique (GALLANT, 1986) to establish generalization guarantees with respect to future attacks made by $\mathbb{A}$. Specifically, we simply output a predictor $\hat{h}$ that has survived a sufficient number of attacks from $\mathbb{A}$. The full algorithm and proof are presented in Appendix F.

Theorem 12 *For any class $\mathcal{H}$, Algorithm 4 robustly PAC learns $\mathcal{H}$ w.r.t. any (possibly randomized and imperfect) attacker $\mathbb{A}$ and any distribution $\mathcal{D}$ such that $\inf_{h \in \mathcal{H}} \mathrm{err}_{\mathbb{A}}(h; \mathcal{D}) = 0$, with sample complexity $m(\varepsilon, \delta)$ and oracle complexity $T(\varepsilon, \delta)$ at most $O\left(\frac{\mathrm{lit}(\mathcal{H}) \log(\mathrm{lit}(\mathcal{H})/\delta)}{\varepsilon}\right)$.*

Gaps and Open Questions Currently we only provide generalization guarantees in the realizable setting. It would be interesting to extend our guarantees to the agnostic setting. Are there algorithms with better sample complexity perhaps depending only on the VC dimension? We established such a result in Section 3 with access to a perfect attack oracle, but the same approach does not go through when using an imperfect attacker. What about better oracle complexity? Can we obtain similar generalization guarantees for adaptive attacking algorithms that change over time? Can we obtain generalization guarantees against a family of attacking algorithms (e.g., first order attacks)?

6. Discussion

In this paper, we consider robust learning with respect to *unknown* perturbation sets $\mathcal{U}$. We initiate the quest to find the "right" model of access to $\mathcal{U}$ by considering different forms of access and studying the robustness guarantees achievable in each case. One of the main takeaways from this work is that we need to be mindful about what form of access to $\mathcal{U}$ we are assuming, because the guarantees that can be achieved can be different. So knowledge about $\mathcal{U}$ should not be thought of as a free resource, but rather we should quantify the complexity of the information we are asking about $\mathcal{U}$.

In some ways, adversarial learning is an arms race, and Athalye et al. (2018) have illustrated that predictors trained to be secure against a specific attack, might be easily defeated by a different attack.

Our *imperfect* attack oracle model in Section 5 certainly suffers from this problem. But, it can also be viewed as taking a step towards addressing it, as it provides a generic way of turning any new attack into a defence, and thus defending against it, and since this is done in a black-box manner, could at the very least hasten the development time needed to defend against a new attack. The online model in Section 4 in a sense does so explicitly, and can indeed handle arbitrary new attacks.

In Section 3 we establish robust generalization guarantees w.r.t. any attacking algorithm $\mathbb{A}$ for $\mathcal{U}$, but it requires a perfect adversarial attack oracle for $\mathcal{U}$: $\mathsf{O}_{\mathcal{U}}$, and in Section 5, we establish a generalization guarantee w.r.t. a *specific* attack algorithm $\mathbb{A}$ when given black-box access to $\mathbb{A}$ at training-time. These are in a sense two opposing ends of the spectrum. Are there other interesting models that provide weaker access than a perfect attack oracle $\mathsf{O}_{\mathcal{U}}$, but also provide a stronger guarantee than that of generalization to a particular attack? For example, under what conditions, can we generalize to a test-time attacker $\mathbb{A}_{\text{test}}$ that is different from the attacker $\mathbb{A}_{\text{train}}$ used at training-time. What if we are interested in providing guarantees to a family of test-time attackers (e.g. first-order algorithms), what form of access would be sufficient and necessary?

Under explicit knowledge of the perturbation set $\mathcal{U}$, Montasser et al. (2019) showed that we can robustly learn any hypothesis class $\mathcal{H}$ that is PAC learnable, i.e., finite $\text{vc}(\mathcal{H})$. Given only a perfect attack oracle to $\mathcal{U}$, we show in this paper that we can robustly learn any hypothesis class $\mathcal{H}$ that is online learnable, i.e., finite $\text{lit}(\mathcal{H})$, and we give lower bounds showing that online learnability is necessary. Are there other models of access to $\mathcal{U}$ that would allow us to robustly learn a broader family of hypothesis classes beyond those that are online learnable?

Our approach to robust learning in this work is modular, in particular, the *perfect* and *imperfect* attack oracles that we consider are independent of the hypothesis class $\mathcal{H}$ since they just receive a predictor $g : \mathcal{X} \to \mathcal{Y}$ as input. But how is g provided? And do we expect the oracles to accept as input any predictor g regardless of its complexity? A more careful look reveals that for the simple algorithm CycleRobust (Algorithm 1) it suffices for the perfect attack oracle $\mathsf{O}_{\mathcal{U}}$ to accept only predictors $h \in \mathcal{H}$. This seems like it creates a dependency and breaks the modularity, but it does not, since, e.g., the oracle might be implemented in terms of a much larger and more generic class, such as neural nets with any architecture, as opposed to the specific architecture we are trying to learn, and which the oracle need not be aware of. But the more sophisticated algorithm RLUA (Algorithm 1) requires calling the oracle on predictors outside $\mathcal{H}$ and so we do need the oracle to accept arbitrary predictors, or at least predictors from a much broader class than $\mathcal{H}$. How can this be translated to a computational rather than purely mathematical framework, and implemented in practice?

Are there sensible but generic assumptions on the perturbation set $\mathcal{U}$ that can lead to improved guarantees? Either assumptions that are on $\mathcal{U}$ separate from the class $\mathcal{H}$, i.e., that hold even if $\mathcal{H}$ is applied to one relabeling or permutation of $\mathcal{X}$ and $\mathcal{U}$ is applied to a different relabeling, or that rely on simple and generic relationships between $\mathcal{U}$ and $\mathcal{H}$.

Acknowledgments

This work was supported in part by DARPA under cooperative agreement HR00112020003[3] and NSF BIGDATA award 1546500. Part of this work was done at the IDEAL Fall 2020 special quarter on *Theory of Deep Learning* funded by NSF TRIPOD award 1934843.

3. The views expressed in this work do not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred. Approved for public release; distribution is unlimited.

References

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 852–860. ACM, 2019. doi: 10.1145/3313276.3316312. URL https://doi.org/10.1145/3313276.3316312.

Hassan Ashtiani, Vinayak Pathak, and Ruth Urner. Black-box certification and learning under adversarial perturbations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 388–398. PMLR, 2020. URL http://proceedings.mlr.press/v119/ashtiani20a.html.

P. Assouad. Densité et dimension. *Annales de l'Institut Fourier (Grenoble)*, 33(3):233–282, 1983.

Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR, 2018. URL http://proceedings.mlr.press/v80/athalye18a.html.

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009. URL http://www.cs.mcgill.ca/%7Ecolt2009/papers/032.pdf#page=1.

Avrim Blum and Yishay Monsour. Learning, regret minimization, and equilibria. 2007.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4):929–965, 1989.

Olivier Bousquet, Steve Hanneke, Shay Moran, and Nikita Zhivotovskiy. Proper learning, helly number, and an optimal SVM bound. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 582–609. PMLR, 2020. URL http://proceedings.mlr.press/v125/bousquet20a.html.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inf. Theory*, 50(9):2050–2057, 2004. doi: 10.1109/TIT.2004.833339. URL https://doi.org/10.1109/TIT.2004.833339.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 2019. URL http://proceedings.mlr.press/v97/cohen19c.html.

O. David, S. Moran, and A. Yehudayoff. Supervised learning through the lens of compression. In *Advances in Neural Information Processing Systems 29*, pages 2784–2792, 2016.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1802–1811, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/engstrom19a.html.

S. I. GALLANT. Optimal linear discriminants. *Eighth International Conference on Pattern Recognition*, pages 849–852, 1986. URL https://ci.nii.ac.jp/naid/10008965845/en/.

Shafi Goldwasser, Adam Tauman Kalai, Yael Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b6c8cf4c587f2ead0c08955ee6e2502b-Abstract.html.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

T. Graepel, R. Herbrich, and J. Shawe-Taylor. PAC-Bayesian compression bounds on the prediction error of learning algorithms for classification. *Machine Learning*, 59(1-2):55–76, 2005.

Steve Hanneke, Roi Livni, and Shay Moran. Online learning with simple predictors and a combinatorial characterization of minimax in 0/1 games. In *Proceedings of the* 34^{th} *Conference on Learning Theory*, 2021.

Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997. ISBN 978-0-521-58713-6.

Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *CoRR*, abs/1908.08016, 2019. URL http://arxiv.org/abs/1908.08016.

Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *CoRR*, abs/2006.12655, 2020. URL https://arxiv.org/abs/2006.12655.

Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 656–672. IEEE, 2019. doi: 10.1109/SP.2019.00044. URL https://doi.org/10.1109/SP.2019.00044.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2(4):285–318, 1987. doi: 10.1007/BF00116827. URL https://doi.org/10.1007/BF00116827.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2): 212–261, 1994. doi: 10.1006/inco.1994.1009. URL https://doi.org/10.1006/inco.1994.1009.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Pratyush Maini, Eric Wong, and J. Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6640–6650. PMLR, 2020. URL http://proceedings.mlr.press/v119/maini20a.html.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2512–2530, Phoenix, USA, 25–28 Jun 2019. PMLR.

Omar Montasser, Surbhi Goel, Ilias Diakonikolas, and Nathan Srebro. Efficiently learning adversarially robust halfspaces with noise. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7010–7021. PMLR, 2020a. URL http://proceedings.mlr.press/v119/montasser20a.html.

Omar Montasser, Steve Hanneke, and Nati Srebro. Reducing adversarially robust learning to non-robust PAC learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/a822554e5403b1d370db84cfbc530503-Abstract.html.

S. Moran and A. Yehudayoff. Sample compression schemes for VC classes. *Journal of the ACM*, 63 (3):21:1–21:10, 2016.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018a. URL https://openreview.net/forum?id=Bys4ob-Rb.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10900–10910, 2018b. URL https://proceedings.neurips.cc/paper/2018/hash/29c0605a3bab4229e46723f89cf59d83-Abstract.html.

Hadi Salman, Jerry Li, Ilya P. Razenshteyn, Pengchuan Zhang, Huan Zhang, Sébastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11289–11300, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/3a24b25a7b092a252166a1641ae953e7-Abstract.html.

N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory (A)*, 13(1):145–147, 1972.

R. E. Schapire and Y. Freund. *Boosting*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2012.

Rob Schapire. Lecture notes - cos 511: Foundations of machine learning. March 2006.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5019–5031, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/f708f064faaf32a43e4d3c784e6af9ea-Abstract.html.

Saharon Shelah. *Classification theory - and the number of non-isomorphic models, Second Edition*, volume 92 of *Studies in logic and the foundations of mathematics*. North-Holland, 1990. ISBN 978-0-444-70260-9.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5858–5868, 2019. URL http://papers.nips.cc/paper/8821-adversarial-training-and-robustness-for-multiple-perturbations.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5283–5292, 2018.

Appendix A. Lemma and Proof of Theorem 1

Lemma 13 (Robust Generalization with Stable Sample Compression) *Let (κ, ρ) be a stable sample compression scheme of size k for $\mathcal{H}$ with respect to the robust loss $\sup_{z \in \mathcal{U}(x)} \mathbb{1}[h(z) \neq y]$.*

Then, for any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ such that $\inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U}}(h;\mathcal{D}) = 0$, any integer $m > 2k$, and any $\delta \in (0,1)$, with probability at least $1-\delta$ over $S = \{(x_1,y_1),\ldots,(x_m,y_m)\}$ iid $\mathcal{D}$-distributed random variables,

$$\mathrm{R}_{\mathcal{U}}(\rho(\kappa(S));\mathcal{D}) \leq \frac{2}{m-2k}\left(k\ln(4) + \ln\left(\frac{1}{\delta}\right)\right).$$

Proof The argument follows an analogous proof from Bousquet et al. (2020) for the 0-1 loss. We observe that the same argument applies to the robust loss, and we provide an explicit proof for completeness. Split the m samples of S into $2k$ sets $S_1,\ldots,S_{2k}$ each of size $\frac{m}{2k}$. Observe that the k compression points chosen by κ, $\kappa(S)$, are in at most k of these sets $S_{i_1^*},\ldots,S_{i_k^*}$ where $i_1^*,\ldots,i_k^* \in \{1,\ldots,2k\}$. Stability of (κ,ρ) implies that $\rho(\kappa(\cup_{j=1}^k S_{i_j^*})) = \rho(\kappa(S))$. Since by definition of (κ,ρ), the robust risk $\mathrm{R}_{\mathcal{U}}(\rho(\kappa(S));S) = 0$, it follows that $\mathrm{R}_{\mathcal{U}}(\rho(\kappa(\cup_{j=1}^k S_{i_j^*})));S) = 0$. This implies that $\rho(\kappa(\cup_{j=1}^k S_{i_j^*}))$ is robustly correct on the remaining sets $\cup_{j\notin\{i_1^*,\ldots,i_k^*\}} S_j$.

Observe that the event that $\mathrm{R}_{\mathcal{U}}(\rho(\kappa(S));\mathcal{D}) > \varepsilon$ implies the event that there exists $i_1,\ldots,i_k \in \{1,\ldots,2k\}$ such that $\mathrm{R}_{\mathcal{U}}(\rho(\kappa(\cup_{j=1}^k S_{i_j}));\mathcal{D}) > \varepsilon$ and $\rho(\kappa(\cup_{j=1}^k S_{i_j}))$ robustly correct on $\cup_{j\notin\{i_1,\ldots,i_k\}} S_j$. Thus,

$$\begin{aligned}
&\Pr_{S\sim\mathcal{D}^m}\left[\mathrm{R}_{\mathcal{U}}(\rho(\kappa(S));\mathcal{D}) > \varepsilon\right] \\
&\leq \Pr_{S\sim\mathcal{D}^m}\left[\exists i_1,\ldots,i_k : \mathrm{R}_{\mathcal{U}}(\rho(\kappa(\cup_{j=1}^k S_{i_j}));\mathcal{D}) > \varepsilon \wedge \mathrm{R}_{\mathcal{U}}(\rho(\kappa(\cup_{j=1}^k S_{i_j}));\cup_{j\notin\{i_1,\ldots,i_k\}} S_j) = 0\right] \\
&\overset{(i)}{\leq} \binom{2k}{k} \Pr_{S\sim\mathcal{D}^m}\left[\mathrm{R}_{\mathcal{U}}(\rho(\kappa(\cup_{j=1}^k S_{i_j}));\mathcal{D}) > \varepsilon \wedge \mathrm{R}_{\mathcal{U}}(\rho(\kappa(\cup_{j=1}^k S_{i_j}));\cup_{j\notin\{i_1,\ldots,i_k\}} S_j) = 0\right] \\
&\overset{(ii)}{\leq} \binom{2k}{k}(1-\varepsilon)^{m/2} < 4^k e^{-\varepsilon m/2},
\end{aligned}$$

where inequality (i) follows from a union bound, and inequality (ii) follows from observing that the $\frac{m}{2}$ samples in $\cup_{j\notin\{i_1,\ldots,i_k\}} S_j$ are independent of $\rho(\kappa(\cup_{j=1}^k S_{i_j})$. Setting $4^k e^{-\varepsilon m/2} = \delta$ and solving for ε yields the stated bound. ■

Proof [of Theorem 1] Let $\mathcal{A} : (\mathcal{X}\times\mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$ be a conservative online learner for $\mathcal{H}$ with mistake bound equal to $\mathrm{lit}(\mathcal{H})$. Let $\mathcal{U} : \mathcal{X} \to 2^{\mathcal{X}}$ be an arbitrary adversarial set that is unknown to the learning algorithm and $\mathsf{O}_{\mathcal{U}}$ a black-box perfect attack oracle for $\mathcal{U}$. Let $\mathcal{D}$ be an arbitrary distribution over $\mathcal{X}\times\mathcal{Y}$ that is robustly realizable with some concept $h^* \in \mathcal{H}$, i.e., $\mathrm{R}_{\mathcal{U}}(h^*;\mathcal{D}) = 0$. 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 iid sample from $\mathcal{D}$. Our proof will be divided into two main parts.

Zero Empirical Robust Loss Observe that the output of CycleRobust (Algorithm 1): $\hat{h} = \mathcal{B}(S,\mathsf{O}_{\mathcal{U}},\mathcal{A})$, achieves zero robust loss on the training data, $\mathrm{R}_{\mathcal{U}}(\hat{h};S) = 0$. This follows because whenever CycleRobust (Algorithm 1) terminates, Steps 4-11 imply that it made a full pass on dataset S without encountering any example (x_i,y_i) where predictor $\hat{h}$ is not robustly correct. Furthermore, since conservative online learner $\mathcal{A}$ has a finite mistake bound of $\mathrm{lit}(\mathcal{H})$, it implies that the number of full passes (execution of Step 3) Algorithm 1 makes over S is at most $\mathrm{lit}(\mathcal{H})$, and in each pass m oracle queries to $\mathsf{O}_{\mathcal{U}}$ are made. Thus, with at most $m\mathrm{lit}(\mathcal{H})$ oracle queries to $\mathsf{O}_{\mathcal{U}}$, CycleRobust (Algorithm 1) outputs a predictor $\hat{h}$ with zero robust loss on S, $\mathrm{R}_{\mathcal{U}}(\hat{h};S) = 0$.

Robust Generalization through Stable Sample Compression CycleRobust (Algorithm 1) can be viewed as a stable compression scheme for the robust loss. Specifically, the output of the compression function $\kappa(S, \mathsf{O}_{\mathcal{U}}, \mathcal{A})$ is an order-dependent sequence that contains all examples (x_i, y_i) on which $\hat{h}$ was not robustly correct while cycling through dataset S (Steps 6-7), since $\mathcal{A}$ has a finite mistake bound of $\mathrm{lit}(\mathcal{H})$, it follows that $|\kappa(S, \mathsf{O}_{\mathcal{U}}, \mathcal{A})| \leq \mathrm{lit}(\mathcal{H})$. The reconstruction function ρ simply runs CycleRobust (Algorithm 1) on the compressed dataset $S' = \kappa(S, \mathsf{O}_{\mathcal{U}}, \mathcal{A})$. The fact that $\mathcal{A}$ is a conservative online learner implies that $\hat{h} = \mathcal{B}(S, \mathsf{O}_{\mathcal{U}}, \mathcal{A}) = \mathcal{B}(S', \mathsf{O}_{\mathcal{U}}, \mathcal{A})$. Since $\mathrm{R}_{\mathcal{U}}(\hat{h}; S) = 0$, this establishes that (κ, ρ) is a sample compression scheme for the robust loss. Furthermore, since $\mathcal{A}$ is a conservative online learner, observe that for any S'' such that $\kappa(S, \mathsf{O}_{\mathcal{U}}, \mathcal{A}) \subseteq S'' \subseteq S$ it holds that $\kappa(S, \mathsf{O}_{\mathcal{U}}, \mathcal{A}) = \kappa(S'', \mathsf{O}_{\mathcal{U}}, \mathcal{A})$. That is, removing any of the examples from S on which $\hat{h}$ was robustly correct in Step 6 will not change the output of the compression function κ. Thus, the pair (κ, ρ) is a stable sample compression scheme for the robust loss of size $\mathrm{lit}(\mathcal{H})$. To conclude the proof, Lemma 13 guarantees that for a sample size $m(\varepsilon, \delta) = O\left(\frac{\mathrm{lit}(\mathcal{H}) + \log(1/\delta)}{\varepsilon}\right)$, the robust risk $\mathrm{R}_{\mathcal{U}}(\hat{h}; \mathcal{D}) \leq \varepsilon$. ■

Appendix B. Auxiliary Lemmas and Proof of Theorem 2

Lemma 14 (Properties of α-Boost, 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 $\mathcal{A}$ an arbitrary PAC learner for $\mathcal{C}$ (for $\varepsilon = 1/3$, $\delta = 1/3$). Then, running α-Boost on S with black-box oracle access to $\mathcal{A}$ with $\alpha = \frac{1}{2}\ln\left(1 + \sqrt{\frac{2\ln m}{T}}\right)$ for $T = \lceil 112\ln(m)\rceil = O(\log m)$ rounds suffices to produce a sequence of hypotheses $h_1, \ldots, h_T \in \mathrm{im}(\mathcal{A})$ such that*

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

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

Lemma 15 (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 $N = O\left(\frac{\mathrm{vc}^*(\mathcal{H}) + \log(1/\delta)}{\varepsilon^2}\right)$ independent random indices $i_1, \ldots, i_N \sim \mathrm{Uniform}(\{1, \ldots, T\})$, we have:*

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

Lemma 16 (Montasser et al. (2019)) *For any $k \in \mathbb{N}$ and fixed function $\phi : (\mathcal{X} \times \mathcal{Y})^k \to \mathcal{Y}^{\mathcal{X}}$, for any distribution P over $\mathcal{X} \times \mathcal{Y}$ and any $m \in \mathbb{N}$, for $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ iid P-distributed random variables, with probability at least $1 - \delta$, if $\exists i_1, \ldots, i_k \in \{1, \ldots, m\}$ s.t. $\hat{R}_{\mathcal{U}}(\phi((x_{i_1}, y_{i_1}), \ldots, (x_{i_k}, y_{i_k})); S) = 0$, then*

$$\mathrm{R}_{\mathcal{U}}(\phi((x_{i_1}, y_{i_1}), \ldots, (x_{i_k}, y_{i_k})); P) \leq \frac{1}{m - k}(k\ln(m) + \ln(1/\delta)).$$

Algorithm 1: Robust Learner against Unknown Adversaries (RLUA)

Input: Training dataset $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, black-box online learner $\mathcal{A}$ for $\mathcal{H}$, black-box perfect attack oracle $\mathsf{O}_{\mathcal{U}}$.

Set $n = O(\mathrm{vc}(\mathcal{A}))$.

Foreach $L \subset S$ such that $|L| = n$, run `CycleRobust` on $(L, \mathcal{A}, \mathsf{O}_{\mathcal{U}})$, and denote by $\hat{\mathcal{H}}$ the resulting set of predictors.

Call `Discretizer` on $(S, \hat{\mathcal{H}}, \mathsf{O}_{\mathcal{U}})$, and denote by $\hat{S}_{\mathcal{U}}$ its output.

Initialize D_1 to be uniform over $\hat{S}_{\mathcal{U}}$, and set $T = O(\log |S_{\mathcal{U}}|)$.

for $1 \le t \le T$ **do**

  Sample $S' \sim D_t^n$, and project S' to dataset $L_t \subseteq S$ by replacing each perturbation z with its corresponding example x.

  Call `CycleRobust` on $(L_t, \mathcal{A}, \mathsf{O}_{\mathcal{U}})$, and denote by f_t its output predictor.

  Compute a new distribution D_{t+1} by applying the following update for each $(z, y) \in \hat{S}_{\mathcal{U}}$:

$$D_{t+1}(\{(z,y)\}) = \frac{D_t(\{(z,y)\})}{Z_t} \times \begin{cases} e^{-2\alpha} & \text{if } f_t(z) = y \\ 1 & \text{otherwise} \end{cases}$$

  where Z_t is a normalization factor and α is set as in Lemma 14.

Sample $N = O(\mathrm{vc}^*(\mathcal{A}))$ i.i.d. indices $i_1, \ldots, i_N \sim \mathrm{Uniform}(\{1, \ldots, T\})$.

(repeat previous step until $g = \mathrm{MAJ}(f_{i_1}, \ldots, f_{i_N})$ satisfies $\mathrm{R}_{\mathcal{U}}(g; S) = 0$)

Output: A majority-vote $\mathrm{MAJ}(f_{i_1}, \ldots, f_{i_N})$ predictor.

`CycleRobust` (*Dataset L, Learner $\mathcal{A}$, Oracle $\mathsf{O}_{\mathcal{U}}$*) **:**

  Initialize $Z = \{\}$, and initialize $\hat{h} = \mathcal{A}(Z)$.

  Set `FullRobustPass` = `False`.

  while *`FullRobustPass` is `False`* **do**

    Set `FullRobustPass` = `True`.

    for $1 \le i \le m$ **do**

      Certify the robustness of $\hat{h}$ on (x_i, y_i) by sending the query $(\hat{h}, (x_i, y_i))$ to the perfect attack oracle $\mathsf{O}_{\mathcal{U}}$.

      if $\hat{h}$ *is not robustly correct on* (x_i, y_i) **then**

        Let z be the perturbation returned by $\mathsf{O}_{\mathcal{U}}$ where $\hat{h}(z) \neq y_i$.

        Add (z, y_i) to the set Z.

        Update $\hat{h}$ by running $\mathcal{A}$ on example (z, y_i), or equivalently, set $\hat{h} = \mathcal{A}(Z)$.

        Set `FullRobustPass` = `False`.

  return Predictor $\hat{h}$.

`Discretizer` (*Dataset S, Predictors $\hat{\mathcal{H}}$, Oracle $\mathsf{O}_{\mathcal{U}}$*) **:**

  Initialize **for** $(x, y) \in S$ **do**

    Initialize $P = \{(x, y)\}$.

    Let $f_P^y : \mathcal{X} \to \mathcal{Y}$ be a predictor of the form:

    $f_P^y(x') = y$ if and only if $\left(\exists_{(z,y)\in P}\right) \left(\forall_{h\in\hat{\mathcal{H}}}\right) \mathbb{1}_{[h(z)\neq y]} = \mathbb{1}_{[h(x')\neq y]}$.

    Send the query $(f_P^y, (x, y))$ to the perfect attack oracle $\mathsf{O}_{\mathcal{U}}$.

    while f_P^y *is not robustly correct on* (x, y) **do**

      Let z be the perturbation returned by $\mathsf{O}_{\mathcal{U}}$ where $f_P^y(z) \neq y$.

      Append (z, y) to the set P.

      Send an updated query $(f_P^y, (x, y))$ to the perfect attack oracle $\mathsf{O}_{\mathcal{U}}$.

Lemma 17 (Montasser et al. (2020b)) *Let* $\mathrm{co}^k(\mathcal{H}) = \{x \mapsto \mathrm{MAJ}(h_1, \ldots, h_k)(x) : h_1, \ldots, h_k \in \mathcal{H}\}$. *Then, the dual VC dimension of* $\mathrm{co}^k(\mathcal{H})$ *satisfies* $\mathrm{vc}^*(\mathrm{co}^k(\mathcal{H})) \leq O(d^* \log k)$.

Theorem 18 (Weak Robust Learner) *For any class* $\mathcal{H}$ *with* $\mathrm{vc}(\mathcal{H}) = d$ *and* $\mathrm{vc}^*(\mathcal{H}) = d$, RLUA *(Algorithm 1) robustly PAC learns* $\mathcal{H}$ *w.r.t any* $\mathcal{U}$ *with:*

1. *Sample Complexity* $m(\varepsilon, \delta) = O\left(\frac{dd^{*2}\log^2 d^*}{\varepsilon}\log\left(\frac{dd^{*2}\log^2 d^*}{\varepsilon}\right) + \frac{\log(1/\delta)}{\varepsilon}\right)$.
2. *Oracle Complexity* $T(\varepsilon, \delta) = O\left(m(\varepsilon, \delta)^{dd^{*2}\log^2 d^*} + m(\varepsilon, \delta)^{dd^* \log d^*}\mathrm{lit}(\mathcal{H})\right)$.

Proof Let $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$ be an online learner for $\mathcal{H}$ with mistake bound $M(\mathcal{A}, \mathcal{H}) = O(\mathrm{lit}(\mathcal{H}))$. We do not require $\mathcal{A}$ to be "proper" (i.e. returns a predictor in $\mathcal{H}$), 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 $\mathrm{vc}(\mathcal{A}) = \mathrm{vc}(\mathrm{im}(\mathcal{A}))$ and $\mathrm{vc}^*(\mathcal{A}) = \mathrm{vc}^*(\mathrm{im}(\mathcal{A}))$ the primal and dual VC dimension of the image of $\mathcal{A}$, i.e. the class $\mathrm{im}(\mathcal{A}) = \{\mathcal{A}(S) | S \in (\mathcal{X} \times \mathcal{Y})^*\}$ of the possible hypothesis $\mathcal{A}$ might return. We will first prove a sample and oracle complexity bound stated in terms of $\mathrm{vc}(\mathcal{A})$ and $\mathrm{vc}^*(\mathcal{A})$, and later, at the end of the proof, we will use a result due to (Hanneke et al., 2021) to bound $\mathrm{vc}(\mathcal{A})$ and $\mathrm{vc}^*(\mathcal{A})$ in terms of $d = \mathrm{vc}(\mathcal{H})$ and $d^* = \mathrm{vc}(\mathcal{H})$ for a specific online learner $\mathcal{A}$.

Let $\mathcal{U} : \mathcal{X} \to 2^{\mathcal{X}}$ be an arbitrary adversary that is unknown to the learner. Let $\mathcal{D}$ be an arbitrary distribution over $\mathcal{X} \times \mathcal{Y}$ that is robustly realizable with some concept $h^* \in \mathcal{H}$, i.e., $\mathrm{R}_{\mathcal{U}}(h^*; \mathcal{D}) = 0$. 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 iid sample from $\mathcal{D}$.

Zero Empirical Robust Loss. Let $L \subseteq S$. Let $\mathcal{A}_{\mathrm{cyc}}$ be CycleRobust (Algorithm 1) from Theorem 1. By Theorem 1, running $\mathcal{A}_{\mathrm{cyc}}$ on input L with black-box access to $\mathsf{O}_{\mathcal{U}}$ and black-box access to $\mathcal{A}$, guarantees that the output $\hat{h} = \mathcal{A}_{\mathrm{cyc}}(L, \mathsf{O}_{\mathcal{U}}, \mathcal{A})$ satisfies $\mathrm{R}_{\mathcal{U}}(\hat{h}; L) = 0$ with at most $|L|\,\mathrm{lit}(\mathcal{H})$ oracle queries to $\mathsf{O}_{\mathcal{U}}$.

Discretization Before we can apply the compression approach, we will inflate dataset S to a (potentially infinite) larger dataset $S_{\mathcal{U}} = \bigcup_{i \leq m}\{(z, y_i) : z \in \mathcal{U}(x_i)\}$ that includes all possible adversarial perturbations under $\mathcal{U}$. There are two challenges that need to be addressed. First, $S_{\mathcal{U}}$ can be potentially infinite, and so we would need to discretize it somehow. Second, the learner does not know $\mathcal{U}$ and so the inflation can be carried only through interaction with the perfect attack oracle $\mathsf{O}_{\mathcal{U}}$. Denote by $\hat{\mathcal{H}} = \{\mathcal{A}_{\mathrm{cyc}}(L) : L \subseteq S, |L| = n\}$ where $n = O(\mathrm{vc}(\mathcal{A}))$. Think of $\hat{\mathcal{H}}$ as the effective hypothesis class that is used by our robust learning algorithm $\mathcal{B}$ that we are constructing. Note that $|\hat{\mathcal{H}}| \leq |\{L : L \subseteq S, |L| = n\}| = \binom{m}{n} \leq \left(\frac{em}{n}\right)^n$. We will now apply classic tools from VC theory to argue that there is a finite number of behaviors when projecting the infinite unknown set $S_{\mathcal{U}}$ onto $\hat{\mathcal{H}}$. Specifically, consider a *dual class* $\mathcal{G}$: a set of functions $g_{(x,y)} : \hat{\mathcal{H}} \to \{0, 1\}$ defined as $g_{(x,y)}(h) = \mathbb{1}[h(x) \neq y]$, for each $h \in \hat{\mathcal{H}}$ and each $(x, y) \in S_{\mathcal{U}}$. The VC dimension of $\mathcal{G}$ is at most the *dual VC dimension* of $\hat{\mathcal{H}}$: $\mathrm{vc}^*(\hat{\mathcal{H}})$, which is at most $\mathrm{vc}^*(\mathcal{A})$ since $\hat{\mathcal{H}} \subseteq \mathrm{im}(\mathcal{A})$. The set of behaviors when projecting $S_{\mathcal{U}}$ onto $\hat{\mathcal{H}}$ is defined as follows:

$$S_{\mathcal{U}}|_{\hat{\mathcal{H}}} = \left\{\left(g_{(z,y)}(h_1), \ldots, g_{(z,y)}(h_{|\hat{\mathcal{H}}|})\right) : (z, y) \in S_{\mathcal{U}}\right\}.$$

Now denote by $\hat{S}_{\mathcal{U}}$ a subset of $S_{\mathcal{U}}$ which includes exactly one $(z, y) \in S_{\mathcal{U}}$ for each distinct classification $\left(g_{(z,y)}(h)\right)_{h \in \hat{\mathcal{H}}}$ of $\hat{\mathcal{H}}$ realized by some $(z, y) \in S_{\mathcal{U}}$. In particular, by applying Sauer's

lemma Vapnik and Chervonenkis (1971); Sauer (1972) on the dual class $\mathcal{G}$, $|\hat{S}_{\mathcal{U}}| = \left|S_{\mathcal{U}}|_{\hat{\mathcal{H}}}\right| \leq \left(\frac{e|\hat{\mathcal{H}}|}{d^*}\right)^{d^*}$, which is at most m^{nd^*}. In particular, note that for any $T \in \mathbb{N}$ and $h_1, \ldots, h_T \in \hat{\mathcal{H}}$, if $\frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[h_t(x) = y] > \frac{1}{2}$ for every $(z, y) \in \hat{S}_{\mathcal{U}}$, then $\frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[h_t(x) = y] > \frac{1}{2}$ for every $(z, y) \in S_{\mathcal{U}}$ as well, which would further imply $\mathrm{R}_{\mathcal{U}}(\mathrm{MAJ}(h_1, \ldots, h_T); S) = 0$. Thus, we have shown that there *exists* a finite discretization $\hat{S}_{\mathcal{U}}$ of $S_{\mathcal{U}}$ where it suffices to find predictors $h_1, \ldots, h_T \in \hat{\mathcal{H}}$ that achieve zero loss on $\hat{S}_{\mathcal{U}}$.

It remains to show how to construct the discretization $\hat{S}_{\mathcal{U}}$ using only interactions with the perfect attack oracle $\mathsf{O}_{\mathcal{U}}$. To this end, for each $(x, y) \in S$, initialize $P = \{(x, y)\}$. The robust learner $\mathcal{B}$ constructs a query $(f_P, (x, y))$ where $f_P : \mathcal{X} \to \mathcal{Y}$ is a predictor of the form:

$$f_P(x') = y \text{ if and only if } \left(\exists_{(z,y)\in P}\right)\left(\forall_{h\in\hat{\mathcal{H}}}\right) g_{(z,y)}(h) = g_{(x',y)}(h).$$

By the definition of f_P, if there is a perturbation $z' \in \mathcal{U}(x)$ such that the classification pattern $\left(g_{(z',y)}(h)\right)_{h\in\hat{\mathcal{H}}}$ is distinct from the classification pattern $\left(g_{(z,y)}(h)\right)_{h\in\hat{\mathcal{H}}}$ of any of the points $(z, y) \in P$, then $f_P(z') \neq y$, and therefore the oracle $\mathsf{O}_{\mathcal{U}}$ would reveal to the learner perturbation z'. Next, the learner adds the point (z', y) to P, and repeats the procedure again until f_P is robustly correct on example (x, y). In each oracle query, the learner is forcing the oracle $\mathsf{O}_{\mathcal{U}}$ to reveal perturbations $z \in \mathcal{U}(x)$ with distinct classification patterns that the learner did not see before. Since we know that $\left|\hat{S}_{\mathcal{U}}\right| \leq m^{n\mathrm{vc}^*(\mathcal{A})}$, the learner makes at most $m^{n\mathrm{vc}^*(\mathcal{A})}$ oracle calls to $\mathsf{O}_{\mathcal{U}}$ before f_P is robustly correct on (x, y). This process is repeated for each training example $(x, y) \in S$, and so the total number of oracle calls to $\mathsf{O}_{\mathcal{U}}$ is at most $m^{n\mathrm{vc}^*(\mathcal{A})+1}$.

Oracle Complexity Our robust learner $\mathcal{B}$ makes $\left(\frac{em}{n}\right)^n n\mathrm{lit}(\mathcal{H})$ oracle calls to $\mathsf{O}_{\mathcal{U}}$ to construct $\hat{\mathcal{H}}$ and $m^{n\mathrm{vc}^*(\mathcal{A})+1}$ oracle calls to $\mathsf{O}_{\mathcal{U}}$ to construct $\hat{S}_{\mathcal{U}}$.

Sample Complexity and Robust Generalization We proceed by running the sample compression scheme from Montasser et al. (2019) on the discretized dataset $\hat{S}_{\mathcal{U}}$. In this stage no more oracle queries to $\mathsf{O}_{\mathcal{U}}$ are needed since the learner has already precomputed $\hat{\mathcal{H}}$ and the discretized dataset $\hat{S}_{\mathcal{U}}$. As mentioned above, our goal in this stage is to find predictors $h_1, \ldots, h_T \in \hat{\mathcal{H}}$ where the majority-vote $\mathrm{MAJ}(h_1, \ldots, h_T)$ achieves zero loss on $\hat{S}_{\mathcal{U}}$. This implies that $\mathrm{MAJ}(h_1, \ldots, h_T)$ achieves zero robust loss on S, $\mathrm{R}_{\mathcal{U}}(\mathrm{MAJ}(h_1, \ldots, h_T); S) = 0$, by properties of $\hat{\mathcal{H}}$ and $\hat{S}_{\mathcal{U}}$. We will next go about finding such a set of h_t predictors.

We run the α-Boost algorithm on the discretized dataset $\hat{S}_{\mathcal{U}}$, this time with $\mathcal{A}_{\mathrm{cyc}}$ (CycleRobust (Algorithm 1)) as the subprocedure. Specifically, on each round of boosting, α-Boost computes an empirical distribution D_t over $\hat{S}_{\mathcal{U}}$. We draw $n = O(\mathrm{vc}(\mathcal{A}))$ 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 $\mathcal{A}_{\mathrm{cyc}}$ on dataset L_t (this is already precomputed since $\mathcal{A}_{\mathrm{cyc}}(L_t) \in \hat{\mathcal{H}}$ by definition of $\hat{\mathcal{H}}$). 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 $\hat{S}_{\mathcal{U}}$.

By classic PAC learning guarantees Vapnik and Chervonenkis (1974); Blumer et al. (1989), with $n = O(\mathrm{vc}(\mathcal{A}))$, we are guaranteed uniform convergence of 0-1 risk over predictors in $\hat{\mathcal{H}}$. So, for any distribution D over $\mathcal{X} \times \mathcal{Y}$ with $\inf_{h\in\mathcal{H}} \mathrm{err}(h; \mathcal{D}) = 0$, with nonzero probability over $S' \sim \mathcal{D}^n$, every $h' \in \hat{\mathcal{H}}$ satisfying $\mathrm{err}_{S'}(h') = 0$, also has $\mathrm{err}_D(h') < 1/3$. As discussed above, we know that $h_t = \mathcal{A}_{\mathrm{cyc}}(L_t)$ achieves zero robust loss on L_t, $\mathrm{R}_{\mathcal{U}}(h_t; L_t) = 0$, which by definition of the

projection means that $\mathrm{err}_{S'}(h_t) = 0$, and thus $\mathrm{err}_{D_t}(h_t) < 1/3$. This allows us to use $\mathcal{A}_{\mathrm{cyc}}$ with α-Boost to establish a *robust* generalization guarantee. Specifically, Lemma 14 implies that running the α-Boost algorithm with $\hat{S}_{\mathcal{U}}$ as its dataset for $T = O(\log(|\hat{S}_{\mathcal{U}}|))$ rounds, using $\mathcal{A}_{\mathrm{cyc}}$ to produce the hypotheses $h_t \in \hat{\mathcal{H}}$ for the distributions D_t produced on each round of the algorithm, will produce a sequence of hypotheses $h_1, \ldots, h_T \in \hat{\mathcal{H}}$ such that:

$$\forall (z, y) \in \hat{S}_{\mathcal{U}}, \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}[h_i(z) = y] \geq \frac{5}{9}.$$

Specifically, this implies that the majority-vote over hypotheses $h_1, \ldots, h_T$ achieves zero *robust* loss on dataset S, $\mathrm{R}_{\mathcal{U}}(\mathrm{MAJ}(h_1, \ldots, h_T); S) = 0$. Note that each of these classifiers h_t is equal to $\mathcal{A}(L_t, \mathsf{O}_{\mathcal{U}})$ for some $L_t \subseteq S$ with $|L_t| = n$. Thus, the classifier $\mathrm{MAJ}(h_1, \ldots, h_T)$ is representable as the value of an (order-dependent) reconstruction function ϕ with a compression set size

$$nT = O(n \log(|S_{\mathcal{U}}|)).$$

We can further reduce the compression set size by sparsifying the majority-vote. Lemma 15 (with $\varepsilon = 1/18, \delta = 1/3$) guarantees that for $N = O(\mathrm{vc}^*(\mathcal{A}))$, the sampled predictors $h_{i_1}, \ldots, h_{i_N} \in \hat{\mathcal{H}}$ satisfy:

$$\forall (z, y) \in \hat{S}_{\mathcal{U}}, \frac{1}{N}\sum_{j=1}^{N} \mathbb{1}[h_{i_j}(z) = y] > \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}[h_i(z) = y] - \frac{1}{18} > \frac{5}{9} - \frac{1}{18} = \frac{1}{2},$$

so that the majority-vote achieves zero robust loss on S, $\mathrm{R}_{\mathcal{U}}(\mathrm{MAJ}(h_{i_1}, \ldots, h_{i_N}); S) = 0$. Since again, each h_{i_j} is the result of $\mathcal{A}(L_t, \mathsf{O}_{\mathcal{U}})$ for some $L_t \subseteq S$ with $|L_t| = m_0$, we have that the classifier $\mathrm{MAJ}(h_{i_1}, \ldots, h_{i_N})$ can be represented as the value of an (order-dependent) reconstruction function ϕ with a compression set size $nN = O(\mathrm{vc}(\mathcal{A})\mathrm{vc}^*(\mathcal{A}))$. Lemma 16 (Montasser et al. (2019)) which extends to the robust loss the classic compression-based generalization guarantees from the 0-1 loss, implies that for $m \geq c\mathrm{vc}(\mathcal{A})\mathrm{vc}(\mathcal{A})^*$ (for an appropriately large numerical constant c), with probability at least $1 - \delta$ over $S \sim \mathcal{D}^m$,

$$R_{\mathcal{U}}(\mathrm{MAJ}(h_{i_1}, \ldots, h_{i_N}); \mathcal{D}) \leq O\left(\frac{\mathrm{vc}(\mathcal{A})\mathrm{vc}^*(\mathcal{A})}{m}\log(m) + \frac{1}{m}\log(1/\delta)\right). \tag{5}$$

Bounding the complexity of $\mathcal{A}$ A result due to (Hanneke et al., 2021, Theorem 3) states that for any class $\mathcal{H}$ of Littlestone dimension $\mathrm{lit}(\mathcal{H})$ and dual VC dimension d^*, there is an online learner $\mathcal{A}$ with mistake bound $M(\mathcal{A}, \mathcal{H}) = O(\mathrm{lit}(\mathcal{H}))$ which represents its hypotheses as (unweighted) majority votes of $O(d^*)$ predictors of $\mathcal{H}$. In other words,

$$\mathrm{im}(\mathcal{A}) \subseteq \mathrm{co}^{O(d^*)}(\mathcal{H}) \triangleq \left\{x \mapsto \mathrm{MAJ}(h_1, \ldots, h_{O(d^*)})(x) : h_1, \ldots, h_{O(d^*)} \in \mathcal{H}\right\}.$$

By (Blumer et al., 1989), the VC dimension of $\mathrm{co}^{O(d^*)}(\mathcal{H})$ is at most $O(dd^* \log d^*)$, and by Lemma 17, the dual VC dimension of $\mathrm{co}^{O(d^*)}(\mathcal{H})$ is at most $O(d^* \log d^*)$. Since $\mathrm{im}(\mathcal{A}) \subseteq \mathrm{co}^{O(d^*)}(\mathcal{H})$, this implies that $\mathrm{vc}(\mathcal{A}) = O(dd^* \log d^*)$ and $\mathrm{vc}^*(\mathcal{A}) = O(d^* \log d^*)$. Substituting these upper bounds in Equation 5, and setting it less than ε and solving for a sufficient size of m yields the stated sample complexity bound. ■

Proof [of Theorem 2] Let $\mathcal{B}$ be the robust learning algorithm (Algorithm 1) described in Theorem 18. We will use $\mathcal{B}$ as a *weak* robust learner with fixed parameters $\varepsilon_0 = 1/3$ and $\delta_0 = 1/3$. By the guarantee of Theorem 18, with fixed sample complexity $m_0 = O(dd^{*2}\log^2 d^*)$, for any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ such that $\inf_{h\in\mathcal{H}} \mathrm{R}_{\mathcal{U}}(h;\mathcal{D}) = 0$, with probability at least $1/3$ over $S \sim \mathcal{D}^{m_0}$, $\mathrm{R}_{\mathcal{U}}(\mathcal{B}(S);\mathcal{D}) \le 1/3$. Furthermore, $\mathcal{B}$ makes at most $O(dd^{*2}\log^2 d^*)^{O(dd^{*2}\log^2 d^*)} + O(dd^{*2}\log^2 d^*)^{O(dd^*\log d^*)}\mathrm{lit}(\mathcal{H}) = \exp\left\{O(dd^{*2}\log^2 d^*)\right\} + \exp\left\{O(d^2d^{*2}\log^2 d^*)\right\}\mathrm{lit}(\mathcal{H})$ oracle calls to $\mathsf{O}_{\mathcal{U}}$.

We will now boost the confidence and robust error guarantee of the *weak* robust learner $\mathcal{B}$ by running boosting with respect to the *robust* loss (rather than the standard 0-1 loss). Specifically, fix $(\varepsilon,\delta) \in (0,1)$ and a sample size $m(\varepsilon,\delta)$ that will be determined later. Let $S = \{(x_1,y_1),\ldots,(x_m,y_m)\}$ be an iid sample from $\mathcal{D}$. Run the α-Boost algorithm on dataset S using $\mathcal{B}$ as the weak robust learner for a number of rounds $L = O(\log m)$. On each round t, α-Boost computes an empirical distribution D_t over S by applying the following update for each $(x,y) \in S$:

$$D_t(\{(x,y)\}) = \frac{D_{t-1}(\{(x,y)\})}{Z_{t-1}} \times \begin{cases} e^{-2\alpha} & \text{if } \sup_{z\in\mathcal{U}(x)} \mathbb{1}[h_{t-1}(z) \neq y] = 0 \\ 1 & \text{otherwise} \end{cases}$$

where Z_{t-1} is a normalization factor, α is set as in Lemma 14, and h_{t-1} is the *weak* robust predictor outputted by $\mathcal{B}$ on round $t-1$ that satisfies $\mathrm{R}_{\mathcal{U}}(h_{t-1};D_{t-1}) \le 1/3$. Note that computing D_t requires $|S| = m$ oracle calls to $\mathsf{O}_{\mathcal{U}}$. Once D_t is computed, we sample m_0 examples from D_t and run *weak* robust learner $\mathcal{B}$ on these examples to produce a hypothesis h_t with robust error guarantee $\mathrm{R}_{\mathcal{U}}(h_t;D_t) \le 1/3$. This step has failure probability at most $\delta_0 = 1/3$. We will repeat it for at most $\lceil\log(2L/\delta)\rceil$ times, until $\mathcal{B}$ succeeds in finding h_t with robust error guarantee $\mathrm{R}_{\mathcal{U}}(h_t;D_t) \le 1/3$. By a union bound argument, we are guaranteed that with probability at least $1-\delta/2$, foreach $1 \le t \le L$, $\mathrm{R}_{\mathcal{U}}(h_t;D_t) \le 1/3$. Furthermore, by Lemma 14, we are guaranteed that $\mathrm{R}_{\mathcal{U}}(\mathrm{MAJ}(h_1,\ldots,h_L);S) = 0$. Note that each of these classifiers h_t is equal to $\mathcal{B}(S'_t,\mathsf{O}_{\mathcal{U}})$ for some $S'_t \subseteq S$ with $|S'_t| = m_0$. Thus, the classifier $\mathrm{MAJ}(h_1,\ldots,h_L)$ is representable as the value of an (order-dependent) reconstruction function ϕ with a compression set size $m_0L = m_0O(\log m)$. Now, invoking Lemma 16, with probability at least $1-\delta/2$,

$$\mathrm{R}_{\mathcal{U}}(\mathrm{MAJ}(h_1,\ldots,h_L);\mathcal{D}) \le O\left(\frac{m_0\log^2 m}{m} + \frac{\log(2/\delta)}{m}\right),$$

and setting this less than ε and solving for a sufficient size of m yields the stated sample complexity bound.

Oracle Complexity Observe that we run boosting for L rounds, in each round the *weak* robust learner is invoked at most $\lceil\log(2L/\delta)\rceil$ times. In each of these invocations, $\mathcal{B}$ makes at most $\exp\left\{O(dd^{*2}\log^2 d^*)\right\} + \exp\left\{O(d^2d^{*2}\log^2 d^*)\right\}\mathrm{lit}(\mathcal{H})$ oracle calls to $\mathsf{O}_{\mathcal{U}}$, and an additional $m(\varepsilon,\delta)$ oracle calls to $\mathsf{O}_{\mathcal{U}}$ are made by α-Boost to compute the robust error of the h_t hypotheses produced by $\mathcal{B}$. Thus, the total number of calls to $\mathsf{O}_{\mathcal{U}}$ is at most

$$\lceil L\log(2L/\delta)\rceil\left(\exp\left\{O(dd^{*2}\log^2 d^*)\right\} + \exp\left\{O(d^2d^{*2}\log^2 d^*)\right\}\mathrm{lit}(\mathcal{H}) + m(\varepsilon,\delta)\right).$$

■

Algorithm 2: Weighted Majority

Input: paramter $\eta \in [0,1)$, black-box perfect attack oracle $\mathsf{O}_{\mathcal{U}}$, and finite hypothesis class $\mathcal{H}$.

Initialize P_0 to be uniform over $\mathcal{H}$, i.e. $\forall h \in \mathcal{H}, P_0(h) = 1$.
for $1 \le t \le T$ **do**
  Receive (x_t, y_t).
  Certify the robustness of the weighted-majority-vote $\mathrm{MAJ}_{P_{t-1}}$ on (x_t, y_t) by sending the query $(\mathrm{MAJ}_{P_{t-1}}, (x_t, y_t))$ to the perfect attack oracle $\mathsf{O}_{\mathcal{U}}$.
  if $\mathrm{MAJ}_{P_{t-1}}$ *is not robustly correct on* (x_t, y_t) **then**
    Let z_t be the perturbation returned by $\mathsf{O}_{\mathcal{U}}$ where $\mathrm{MAJ}_{P_{t-1}}(z_t) \neq y_t$.
    Foreach $h \in \mathcal{H}$ such that $h(z_t) \neq y_t$, update $P_t(h) = \eta P_{t-1}(h)$.

Output: The weighted-majority-vote MAJ_{P_T} over $\mathcal{H}$.

`Expert` (*Indices* $i_1 < i_2 < \cdots < i_L$*, and hypothesis class* $\mathcal{H}$)**:**
  Initialize $V_1 = \mathcal{H}$.
  for $1 \le t \le T$ **do**
    Receive x_t.
    Let $V_t^y = \{h \in V_t : h(x_t) = y\}$ for $y \in \{\pm 1\}$.
    Let $\tilde{y}_t = \mathrm{argmax}_{y \in \{\pm 1\}} \mathrm{lit}(V_t^y)$ (in case of a tie set $\tilde{y}_t = +1$).
    if $t \in \{i_1, \ldots, i_L\}$ **then**
      Predict $\hat{y}_t = -\tilde{y}_t$.
    else
      Predict $\hat{y}_t = \tilde{y}_t$.
    Update $V_{t+1} = V_t^{\hat{y}_t}$.

Appendix C. Proofs for Agnostic Setting – Section 3.2

Lemma 19 *For any class $\mathcal{H}$ with finite cardinality,* Weighted Majority *(Algorithm 2) guarantees that for any $\mathcal{U}$ and any sequence of examples $(x_1, y_1), \ldots, (x_T, y_T)$:*

$$\sum_{t=1}^{T} \sup_{z \in \mathcal{U}(x_t)} \mathbb{1}[\mathrm{MAJ}_{P_{t-1}}(z) \neq y_t] \le a_\eta \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \sup_{z \in \mathcal{U}(x_t)} \mathbb{1}[h(z) \neq y_t] + b_\eta \ln |\mathcal{H}|\,,$$

where $a_\eta = \frac{\ln(1/\eta)}{\log(2/(1+\eta))}$ and $b_\eta = \frac{1}{\ln(2/(1+\eta))}$. In particular, setting $1 - \eta = \min\{(2\ln|\mathcal{H}|)/T, 1/2\}$ yields

$$\sum_{t=1}^{T} \sup_{z \in \mathcal{U}(x_t)} \mathbb{1}[\mathrm{MAJ}_{P_{t-1}}(z) \neq y_t] \le 2\mathsf{OPT} + 4\sqrt{\mathsf{OPT} \ln |\mathcal{H}|}.$$

Furthermore, Weighted Majority *(Algorithm 2) makes at most T oracle queries to $\mathsf{O}_{\mathcal{U}}$.*

Proof This proof follows from standard analysis for the Weighted Majority algorithm (see e.g. Schapire (2006); Blum and Monsour (2007)). Let $W_t = \sum_{h \in \mathcal{H}} P_t(h)$. Observe that on

round t, if the weighted-majority-vote $\mathrm{MAJ}_{P_{t-1}}$ is not robustly correct on (x_t, y_t), then:

$$W_t = \eta \sum_{h\in\mathcal{H}:h(z_t)\neq y} P_{t-1}(h) + \sum_{h\in\mathcal{H}:h(z_t)=y} P_{t-1}(h) = \eta \sum_{h\in\mathcal{H}:h(z_t)\neq y} P_{t-1}(h) + W_{t-1} - \sum_{h\in\mathcal{H}:h(z_t)\neq y} P_{t-1}(h)$$
$$= W_{t-1} - (1-\eta)\left(\sum_{h\in\mathcal{H}:h(z_t)\neq y} P_{t-1}(h)\right) \leq W_{t-1} - (1-\eta)\frac{1}{2}W_{t-1} = \left(\frac{\eta+1}{2}\right)W_{t-1},$$

where the last inequality follows from the fact that $\sum_{h\in\mathcal{H}:h(z_t)\neq y} P_{t-1}(h) \geq \sum_{h\in\mathcal{H}:h(z_t)=y} P_{t-1}(h)$.

Denote by $M = \sum_{t=1}^{T} \sup_{z\in\mathcal{U}(x_t)} \mathbb{1}[\mathrm{MAJ}_{P_{t-1}}(z) \neq y_t]$ the number of rounds on which the weighted-majority-vote was not robustly correct during the total T rounds. The above implies that $W_T \leq \left(\frac{\eta+1}{2}\right)^M W_0 = \left(\frac{\eta+1}{2}\right)^M |\mathcal{H}|$. On the other hand, denote by $\mathsf{OPT} = \min_{h\in\mathcal{H}} \sum_{t=1}^{T} \sup_{z\in\mathcal{U}(x)} \mathbb{1}[h(z) \neq y]$ the number of rounds on which the best predictor h^* in $\mathcal{H}$ was not robustly correct. Whenever the weighted-majority-vote is not robustly correct, h^* might make a mistake on (z_t, y_t). It follows that after T rounds, $P_T(h^*) \geq \eta^{\mathsf{OPT}}$. Combining the above inequalities, we get

$$\eta^{\mathsf{OPT}} \leq P_T(h^*) \leq W_T \leq \left(\frac{\eta+1}{2}\right)^M |\mathcal{H}|\,,$$

and solving for M yields

$$M \leq \frac{\ln(1/\eta)}{\ln(2/(1+\eta))}\mathsf{OPT} + \frac{1}{\ln(2/(1+\eta))}\ln|\mathcal{H}|\,.$$

To conclude the proof, observe that for $\eta \in [0,1)$, $\ln(2/(1+\eta)) \geq \frac{1-\eta}{2}$, and $\ln(1/\eta) \leq (1-\eta) + (1-\eta)^2$ for $0 \leq 1-\eta \leq 1/2$. Setting $1-\eta = \min\{(2\ln|\mathcal{H}|)/T, 1/2\}$ yields the desired bound. ■

Lemma 20 *For any class $\mathcal{H}$ with finite Littlestone dimension $\mathrm{lit}(\mathcal{H}) < \infty$ and integer T, let $\mathsf{Experts}_{\mathcal{H}} = \{\mathsf{Expert}(i_1, \ldots, i_L) : 1 \leq i_1 < \cdots < i_L \leq T, L \leq \mathrm{lit}(\mathcal{H})\}$ be a set of experts as described in Algorithm 2. Then, running* Weighted Majority *(Algorithm 2) with $\mathsf{Experts}_{\mathcal{H}}$ guarantees that for any perturbation set $\mathcal{U}$ and any sequence of examples $(x_1, y_1), \ldots, (x_T, y_T)$,*

$$\sum_{t=1}^{T} \sup_{z\in\mathcal{U}(x_t)} \mathbb{1}[\mathrm{MAJ}_{P_{t-1}}(z) \neq y_t] \leq 2\mathsf{OPT} + 4\sqrt{\mathsf{OPT}\ln|\mathsf{Experts}_{\mathcal{H}}|},$$

where $\mathsf{OPT} = \min_{h\in\mathcal{H}} \sum_{t=1}^{T} \sup_{z\in\mathcal{U}(x_t)} \mathbb{1}[h(z) \neq y_t]$, and $1-\eta = \min\{(2\ln|\mathsf{Experts}_{\mathcal{H}}|)/T, 1/2\}$. Furthermore, Weighted Majority *(Algorithm 2) makes at most T oracle queries to $\mathsf{O}_{\mathcal{U}}$.*

Proof Let $\mathcal{U}$ be an arbitrary adversary, and $(x_1, y_1), \ldots, (x_T, y_T) \in \mathcal{X}\times\mathcal{Y}$ be an arbitrary sequence. Let $h^* \in \mathcal{H}$ be an optimal robust predictor on this sequence, i.e. $\sum_{t=1}^{T} \sup_{z\in\mathcal{U}(x_t)} \mathbb{1}[h^*(z) \neq y_t] = \mathsf{OPT}$. Let $\texttt{Experts}_{\mathcal{H}} = \{\texttt{Expert}(i_1, \ldots, i_L) : 1 \leq i_1 < \cdots < i_L \leq T, L \leq \mathrm{lit}(\mathcal{H})\}$ denote the set of experts instantiated that simulate the Standard Optimal Algorithm as described in Algorithm 2.

Consider running Weighted Majority (Algorithm 2) with $\mathsf{Experts}_{\mathcal{H}}$ as its finite cardinality set of experts on the sequence $(x_1, y_1), \ldots, (x_T, y_T)$. Consider the set of perturbations returned by the

perfect attack oracle $\mathsf{O}_{\mathcal{U}}$ during the rounds on which the weighted-majority-vote was not robustly correct,

$$Q = \left\{(z_t, y_t) : 1 \le t \le T \wedge \mathrm{MAJ}_{P_{t-1}} \text{ is not robustly correct on } (x_t, y_t)\right\}.$$

By Algorithm 2, there is a choice of rounds $i_1^* < \cdots < i_L^*$ such that $\texttt{Expert}(i_1^*, \ldots, i_L^*) \in \texttt{Experts}_{\mathcal{H}}$ agrees with the predictions of h^* on this particular sequence Q. Observe that the number of mistakes h^* makes on this sequence $M(h^*) := |\{(z, y) \in Q : h^*(z) \neq y\}| \le \mathsf{OPT}$. Thus, the weight of $\texttt{Expert}(i_1^*, \ldots, i_L^*)$ is at least $\eta^{M(h^*)} \ge \eta^{\mathsf{OPT}}$ (since $\eta < 1$). The remainder of the proof follows exactly as in the proof of Theorem 3. ■

Proof [of Theorem 3] Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypotesis class with finite Littlestone dimension $\mathrm{lit}(\mathcal{H}) < \infty$. Let $\mathcal{B} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$ denote the Weighted Majority algorithm running with experts $\mathsf{Experts}_{\mathcal{H}}$ as described in Theorem 3. We will apply a standard online-to-batch conversion Cesa-Bianchi et al. (2004) to get the desired result. Specifically, on input dataset $S = \{(x_j, y_j)\}_{j=1}^m$ that is drawn iid from some unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, output a uniform distribution over hypotheses $\hat{h}_0, \hat{h}_1, \ldots, \hat{h}_{m-1}$ where $\hat{h}_i = \mathcal{B}(\{(x_j, y_j)\}_{j=1}^{i-1})$. We are guaranteed that with probability at least $1 - \delta$ over $S \sim \mathcal{D}^m$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{j \sim \mathsf{Unif}\{0,\ldots,m-1\}} \left[\mathrm{R}_{\mathcal{U}}(\hat{h}_j; \mathcal{D})\right] &\le \frac{1}{m} \sum_{j=1}^{m} \sup_{z \in \mathcal{U}(x_j)} \mathbb{1}[\hat{h}_{j-1}(z) \neq y_j] + \sqrt{\frac{2\ln(1/\delta)}{m}} \\
&\le 2 \min_{h \in \mathcal{H}} \frac{1}{m} \sum_{j=1}^{m} \sup_{z \in \mathcal{U}(x_j)} \mathbb{1}[h(z) \neq y_j] + 4\sqrt{\frac{\ln|\mathsf{Experts}_{\mathcal{H}}|}{m}} + \sqrt{\frac{2\ln(1/\delta)}{m}} \\
&\le 2 \min_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; \mathcal{D}) + 4\sqrt{\frac{\ln|\mathsf{Experts}_{\mathcal{H}}|}{m}} + 2\sqrt{\frac{2\ln(1/\delta)}{m}}.
\end{aligned}$$

This yields a sample complexity bound of $m(\varepsilon, \delta) = O\left(\frac{\ln|\mathsf{Experts}_{\mathcal{H}}| + \ln(1/\delta)}{\varepsilon^2}\right)$. The oracle complexity $T(\varepsilon, \delta) = O(m(\varepsilon, \delta)^2)$ since we invoke learner $\mathcal{B}$ m times on datasets of size at most m. ■

Proof [of Theorem 4] This proof follows an argument originally made by David et al. (2016) to reduce agnostic sample compression to realizable sample compression in the non-robust setting, and later adapted by Montasser et al. (2019) for the robust setting. Let $\mathcal{D}$ be an arbitrary distribution over $\mathcal{X} \times \mathcal{Y}$. 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 iid sample from $\mathcal{D}$. Denote by $\tilde{\mathcal{B}}$ the robust learning algorithm in the realizable setting from Theorem 2, and denote by $\mathcal{A}_{\mathrm{cyc}}$ CycleRobust (Algorithm 1) from Theorem 1. The proof is broken into two parts.

Finding Maximal Subsequence S' with Zero Robust Loss We will use $\mathcal{A}_{\mathrm{cyc}}$ to find a maximal subsequence $S' \subseteq S$ on which the robust loss can be zero, i.e. $\inf_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; S') = 0$. This can be done by running $\mathcal{A}_{\mathrm{cyc}}$ on all 2^m possible subsequences, with a total oracle complexity of $2^m \mathrm{lit}(\mathcal{H})$.

Agnostic Sample Compression We now run the boosting algorithm $\tilde{B}$ on S'. Theorem 2 guarantees that the robust risk of $\hat{h} = \tilde{B}(S', \mathsf{O}_{\mathcal{U}}, \mathcal{A}_{\mathrm{cyc}})$ is zero, $\mathrm{R}_{\mathcal{U}}(\hat{h}; S') = 0$. Since S' is a maximal subsequence on which the robust loss can be zero, this implies that

$$\mathrm{R}_{\mathcal{U}}(\hat{h}; S) \le \min_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; S).$$

Furthermore, the predictor $\hat{h}$ can be specified using $m_0 O(\log |S'|) \leq m_0 O(\log m)$ points from S, which is due to the robust compression guarantee in the proof of Theorem 2. Now, we can apply agnostic sample compression generalization guarantees for the robust loss.

Similarly to the realizable case (see Lemma 16), uniform convergence guarantees for sample compression schemes Graepel et al. (2005) remain valid for the robust loss, by essentially the same argument; the essential argument is the same as in the proof of Lemma 16 except using Hoeffding's inequality to get concentration of the empirical robust risks for each fixed index sequence, and then a union bound over the possible index sequnces as before. We omit the details for brevity. In particular, denoting $T_m = O(\log m)$, for $m > m_0 T_m$, with probability at least $1 - \delta/2$,

$$\mathrm{R}_{\mathcal{U}}(\hat{h}; \mathcal{D}) \leq \hat{\mathrm{R}}_{\mathcal{U}}(\hat{h}; S) + \sqrt{\frac{m_0 T_m \ln(m) + \ln(2/\delta)}{2m - 2m_0 T_m}}.$$

Let $h^* = \mathrm{argmin}_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; \mathcal{D})$ (supposing the min is realized, for simplicity; else we could take an h^* with very-nearly minimal risk). By Hoeffding's inequality, with probability at least $1 - \delta/2$,

$$\hat{\mathrm{R}}_{\mathcal{U}}(h^*; S) \leq \mathrm{R}_{\mathcal{U}}(h^*; \mathcal{D}) + \sqrt{\frac{\ln(2/\delta)}{2m}}.$$

By the union bound, if $m \geq 2m_0 T_m$, with probability at least $1 - \delta$,

$$\begin{aligned}
\mathrm{R}_{\mathcal{U}}(\hat{h}; \mathcal{D}) &\leq \min_{h \in \mathcal{H}} \hat{\mathrm{R}}_{\mathcal{U}}(h; S) + \sqrt{\frac{m_0 T_m \ln(m) + \ln(2/\delta)}{m}} \\
&\leq \hat{\mathrm{R}}_{\mathcal{U}}(h^*; S) + \sqrt{\frac{m_0 T_m \ln(m) + \ln(2/\delta)}{m}} \\
&\leq \mathrm{R}_{\mathcal{U}}(h^*; \mathcal{D}) + 2\sqrt{\frac{m_0 T_m \ln(m) + \ln(2/\delta)}{m}}.
\end{aligned}$$

Since $T_m = O(\log(m))$, the above is at most ε for an appropriate choice of sample size $m = O\big(\frac{m_0}{\varepsilon^2} \log^2\big(\frac{m_0}{\varepsilon}\big) + \frac{1}{\varepsilon^2} \log\big(\frac{1}{\delta}\big)\big)$. ■

Appendix D. Lower Bound Proof for Section 3.3

Proof [of Theorem 5] Let $d = \mathrm{Tdim}(\mathcal{H})$. By definition of the threshold dimension, $\exists P = \{x_1, \ldots, x_d\} \subseteq \mathcal{X}$ that is threshold-shattered using $C = \{h_1, \ldots, h_d\} \subseteq \mathcal{H}$. Let $\mathcal{D}$ be a uniform distribution over $(x_1, +1)$ and $(x_d, -1)$. Let $\mathcal{B}$ be an arbitrary learner in the Perfect Attack Oracle model. For any $h \in C \setminus \{h_d\}$, let $\mathcal{U}_h : \mathcal{X} \to 2^{\mathcal{X}}$ be defined as:

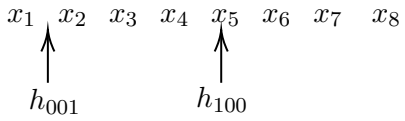
$$\begin{aligned}
\mathcal{U}_h(x_1) &= \{x \in P : h(x) = +1\}, \\
\mathcal{U}_h(x_d) &= \{x \in P : h(x) = -1\} = P \setminus \mathcal{U}_h(x_1), \\
\mathcal{U}_h(x) &= \{x_0\} \ \forall x \in \mathcal{X} \setminus \{x_1, x_d\},
\end{aligned}$$

where $x_0 \in \mathcal{X} \setminus P$.

For any such $\mathcal{U}_h$, observe that finding a predictor $\hat{h} : \mathcal{X} \to \mathcal{Y}$ that achieves zero robust loss on $\mathcal{D}$, $\mathrm{R}_{\mathcal{U}_h}(\hat{h}; \mathcal{D}) = 0$, is equivalent to figuring out which threshold $h \in C \setminus h_d$ was used to construct

$\mathcal{U}_h$, since $\mathrm{R}_{\mathcal{U}_h}(h;\mathcal{D}) = 0$ by definition of $\mathcal{U}_h$, but for any other threshold $h' \in C \setminus h_d$ where $h' \neq h$, $\mathrm{R}_{\mathcal{U}}(h';\mathcal{D}) \geq 1/2$.

We will pick h uniformly at random from $C \setminus h_d$, and we will show that in expectation over the random draw of h, learner $\mathcal{A}$ needs to make at least $\Omega(\log|C \setminus h_d|)$ oracle queries to $\mathsf{O}_{\mathcal{U}_h}$ in order to achieve robust loss zero on $\mathcal{D}$. For ease of presentation, for each $i \in [d-1]$, we will encode $h_i \in C \setminus h_d$ with the binary representation $r(i)$ of integer i, for example:



Thus, drawing h uniformly at random from $C \setminus h_d$ is equivalent to drawing a random bit-string r of length $\lceil \log_2|C \setminus h_d| \rceil$ bits. Next, we will define the behavior of the oracle $\mathsf{O}_{\mathcal{U}_h}$.

Algorithm 3: Perfect Attack Oracle $\mathsf{O}_{\mathcal{U}_h}$

Input: A predictor $f : \mathcal{X} \to \mathcal{Y}$ and a labeled example (x, y).
Output: Assert that f is robustly correct on (x, y) or return a $z \in \mathcal{U}_h(x)$ such that $f(z) \neq y$.

1 **if** $x = x_1$ **then**
2 Output the first $z \in \mathcal{U}_h(x_1)$ **(to the right of** x_1**)** such that $f(z) \neq y$. If no such z exists, assert that f is robustly correct on (x_1, y).
3 **else if** $x = x_d$ **then**
4 Output the first $z \in \mathcal{U}_h(x_d)$ **(to the left of** x_d**)** such that $f(z) \neq y$. If no such z exists, assert that f is robustly correct on (x_d, y).
5 **else**
6 Output x if $f(x) \neq y$, otherwise assert that f is robustly correct on (x, y).

Before learning starts, from the perspective of the learner $\mathcal{B}$, the version space $V_0 = \{h_1, \ldots, h_{d-1}\}$, as any of these thresholds could be the true threshold used by $\mathsf{O}_{\mathcal{U}_{h_r}}$ where r was drawn uniformly at random. On each round t, learner $\mathcal{B}$ constructs a predictor $h_t : \mathcal{X} \to \mathcal{Y}$ and asks the oracle $\mathcal{O}_{\mathcal{U}_{h_r}}$ with a query $q_t = (h_t, (x_t, y_t))$, and the oracle $\mathsf{O}_{\mathcal{U}_{h_r}}$ responds as described in Algorithm 3. Without loss of generality, we can assume that $(x_t, y_t) = (x_1, +1)$ or $(x_t, y_t) = (x_d, -1)$, since queries concerning other points $x \in \mathcal{X} \setminus \{x_1, x_d\}$ do not reveal helpful information for robustly learning distribution $\mathcal{D}$. Foreach round t, the version space V_t describes the set of thresholds that are consistent with the queries constructed by the learner so far, i.e. $\forall i \leq t, \forall h, h' \in V_t, \mathsf{O}_{\mathcal{U}_h}(q_i) = \mathsf{O}_{\mathcal{U}_{h'}}(q_i)$. So, from the perspective of the learner $\mathcal{B}$, any $h \in V_t$ could be the true threshold.

We will show that with each oracle query q_t constructed by the learner $\mathcal{B}$, in expectation over the random draw of r, the size of the newly updated version space $|V_t| \geq \frac{1}{4}|V_{t-1}|$. Formally, the expected size of the version space V_t after round t conditioned on query q_t and V_{t-1} is:

$$\mathop{\mathbb{E}}_{r_t}\left[|V_t|\,|q_t, V_{t-1}\right] = \Pr_{r_t}\left[r_t = 0\right]\mathbb{E}\left[|V_t|\,|q_t, V_{t-1}, r_t\right] + \Pr_{r_t}\left[r_t = 1\right]\mathbb{E}\left[|V_t|\,|q_t, V_{t-1}, r_t\right],$$

where r_t is the t^{th} random bit in the random bit string r. We need to consider two possible cases depending on the query $q_t = (h_t, (x_t, y_t))$. (Without loss of generality, we are assuming that $h_t \in C \setminus h_d$, as the oracle $\mathsf{O}_{\mathcal{U}_{h_r}}$ will treat it as such by Steps 2 and 4).

If $(x_t, y_t) = (x_1, +1)$, and the t^{th} bit of h_t is 0, then:

$$\mathop{\mathbb{E}}_{r_t}\left[|V_t|\,|q_t, V_{t-1}\right] \geq \Pr_{r_t}\left[r_t = 1\right]\mathbb{E}\left[|V_t|\,|q_t, V_{t-1}, r_t\right] = \frac{1}{2}\cdot\frac{1}{2}\left|V_{t-1}\right|.$$

If $(x_t, y_t) = (x_d, -1)$, and the t^{th} bit of h_t is 1, then:

$$\mathop{\mathbb{E}}_{r_t}\left[|V_t| \,|q_t, V_{t-1}\right] \geq \Pr_{r_t}\left[r_t = 0\right] \mathbb{E}\left[|V_t| \,|q_t, V_{t-1}, r_t\right] = \frac{1}{2} \cdot \frac{1}{2} \left|V_{t-1}\right|.$$

Therefore, it follows that $\mathbb{E}_{r_t}\left[|V_t| \,|q_t, V_{t-1}\right] \geq \frac{1}{4}|V_{t-1}|$. Thus, after T rounds, $\mathbb{E}_r\left[|V_T| \,|V_0\right] \geq \frac{1}{4}^T |V_0|$. This implies that there exists a fixed bit-string r^* (or equivalently, an adversary $\mathcal{U}_{h_{r^*}}$) such that for $T \leq \frac{\log|V_0|}{2}$ rounds, $|V_T| \geq 1$. This implies that learner $\mathcal{B}$ needs at least $\frac{\log|V_0|}{2}$ oracle queries to $\mathsf{O}_{\mathcal{U}_{h_{r^*}}}$ in order to robustly learn distribution $\mathcal{D}$. ■

Appendix E. Proofs for Section 4

Proof [of Theorem 9] Observe that the same lowerbound construction from Theorem 5 can be used in the setting of the online model. Specifically, by importing that construction, we get the following: there is a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, such that for any learner $\mathcal{B} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$, there is a perturbation set $\mathcal{U} : \mathcal{X} \to 2^{\mathcal{X}}$ and an adversary $\mathbb{A} : \mathcal{Y}^{\mathcal{X}} \times (\mathcal{X} \times \mathcal{Y}) \to \mathcal{X}$ (Algorithm 3) such that:

$$M_{\mathcal{U},\mathbb{A}}(\mathcal{B}, \mathcal{H}; \mathcal{D}) = \sum_{t=1}^{\infty} \mathbb{1}\left[\mathcal{B}(\{(z_i, y_i)\}_{i=1}^{t-1})(z_t) \neq y_t\right] \geq \frac{\log_2\left(\mathrm{Tdim}(\mathcal{H}) - 1\right)}{2}.$$

This is because in the setting of the Perfect Attack Oracle model, learner $\mathcal{B}$ *chooses* which example $(x, y) \in \mathrm{supp}(\mathcal{D})$ to feed into $\mathbb{A}$, and still learner $\mathcal{B}$ makes $\frac{\log_2\left(\mathrm{Tdim}(\mathcal{H})-1\right)}{2}$ mistakes before fully robustly learning distribution $\mathcal{D}$. While in this setting, the examples (x, y) that are fed into $\mathbb{A}$ are drawn iid from $\mathcal{D}$, and so its at least as hard as the other setting. ■

Appendix F. Proofs for Section 5

Algorithm 4: Robust Learner with Imperfect Attack.

Input: $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, ε, δ, black-box conservative online learner $\mathcal{A}$, black-box attacker $\mathbb{A}$.

1 Initialize $\hat{h} = \mathcal{A}(\emptyset)$.

2 **for** $1 \leq i \leq m$ **do**

3 Let $z_i = \mathbb{A}(\hat{h}, (x_i, y_i))$ be the perturbation returned by the attacker $\mathbb{A}$.

4 If $\hat{h}$ is not correct on (z_i, y_i), update $\hat{h}$ by running online learner on $\mathcal{A}$ on (z_i, y_i).

5 Break when $\hat{h}$ is correct on a consecutive sequence of perturbations of length $\frac{1}{\varepsilon}\log\left(\frac{\mathrm{lit}(\mathcal{H})}{\delta}\right)$.

Output: $\hat{h}$.

Proof [of Theorem 12] Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be an arbitrary hypothesis class, and $\mathcal{A}$ a conservative online learner for $\mathcal{H}$ with mistake bound of $\mathrm{lit}(\mathcal{H})$. Let $\mathcal{U}$ be an arbitrary adversary and $\mathbb{A}$ an arbitrary fixed (but possibly randomized) attack algorithm. Let $\mathcal{D}$ be an arbitrary distribution over $\mathcal{X} \times \mathcal{Y}$ that is robustly realizable,i.e. $\inf_{h \in \mathcal{H}} \mathrm{R}_{\mathcal{U}}(h; \mathcal{D}) = 0$.

Fix $\varepsilon, \delta \in (0, 1)$ and a sample size $m = 2\frac{\mathrm{lit}(\mathcal{H})}{\varepsilon}\log\left(\frac{\mathrm{lit}(\mathcal{H})}{\delta}\right)$. Since online learner $\mathcal{A}$ has a mistake bound of $\mathrm{lit}(\mathcal{H})$, Algorithm 4 will terminate in at most $\frac{\mathrm{lit}(\mathcal{H})}{\varepsilon}\log\left(\frac{\mathrm{lit}(\mathcal{H})}{\delta}\right)$ steps, which is an

upperbound on the number of calls to the attack algorithm $\mathbb{A}$. It remains to show that the output of Algorithm 4, the final predictor $\hat{h}$, will have low error w.r.t. future attacks from $\mathbb{A}$:

$$\mathrm{err}_{\mathbb{A}}(\hat{h}; \mathcal{D}) \triangleq \Pr_{\substack{(x,y)\sim\mathcal{D} \\ \text{randomness of } \mathbb{A}}} \left[\hat{h}(\mathbb{A}(\hat{h}, (x, y))) \neq y\right].$$

Throughout the runtime of Algorithm 4, the online learner $\mathcal{A}$ generates a sequence of at most $\mathrm{lit}(\mathcal{H}) + 1$ predictors. There's the initial predictor from Step 1, plus the $\mathrm{lit}(\mathcal{H})$ updated predictors corresponding to potential updates by online learner $\mathcal{A}$. By a union bound over these predictors, the probability that the final predictor $\hat{h}$ has error more than ε

$$\Pr_{S\sim\mathcal{D}^m}\left[\mathrm{err}_{\mathbb{A}}(\hat{h}; \mathcal{D}) > \varepsilon\right] \leq \Pr_{S\sim\mathcal{D}^m}[\exists j \in [\mathrm{lit}(\mathcal{H})+1] : \mathrm{err}_{\mathbb{A}}(h_j; \mathcal{D}) > \varepsilon] \leq (\mathrm{lit}(\mathcal{H})+1)(1-\varepsilon)^{\frac{1}{\varepsilon}\log\left(\frac{\mathrm{lit}(\mathcal{H})+1}{\delta}\right)} \leq \delta.$$

Therefore, with probability at least $1 - \delta$ over $S \sim \mathcal{D}^m$, Algorithm 4 outputs a predictor $\hat{h}$ with error $\mathrm{err}_{\mathbb{A}}(\hat{h}; \mathcal{D}) \leq \varepsilon$. ■