On Optimal Learning Under Targeted Data Poisoning

Steve Hanneke
Purdue University, USA
steve.hanneke@gmail.com

Amin Karbasi
Yale University, USA and Google Research
amin.karbasi@yale.edu

Mohammad Mahmoody
University of Virginia, USA
mohammad@virginia.edu

Idan Mehalel
Technion, Israel
idanmehalel@gmail.com

Shay Moran
Technion, Israel and Google Research
shaymoran1@gmail.com

Abstract

Consider the task of learning a hypothesis class $\mathcal{H}$ in the presence of an adversary that can replace up to an η fraction of the examples in the training set with arbitrary adversarial examples. The adversary aims to fail the learner on a particular target test point x which is *known* to the adversary but not to the learner. In this work we aim to characterize the smallest achievable error $\varepsilon = \varepsilon(\eta)$ by the learner in the presence of such an adversary in both realizable and agnostic settings. We fully achieve this in the realizable setting, proving that $\varepsilon = \Theta(\mathtt{VC}(\mathcal{H}) \cdot \eta)$, where $\mathtt{VC}(\mathcal{H})$ is the VC dimension of $\mathcal{H}$. Remarkably, we show that the upper bound can be attained by a deterministic learner. In the agnostic setting we reveal a more elaborate landscape: we devise a deterministic learner with a multiplicative regret guarantee of $\varepsilon \leq C \cdot \mathtt{OPT} + O(\mathtt{VC}(\mathcal{H}) \cdot \eta)$, where $C > 1$ is a universal numerical constant. We complement this by showing that for any deterministic learner there is an attack which worsens its error to at least $2 \cdot \mathtt{OPT}$. This implies that a multiplicative deterioration in the regret is unavoidable in this case. Finally, the algorithms we develop for achieving the optimal rates are inherently improper. Nevertheless, we show that for a variety of natural concept classes, such as linear classifiers, it is possible to retain the dependence $\varepsilon = \Theta_{\mathcal{H}}(\eta)$ by a proper algorithm in the realizable setting. Here $\Theta_{\mathcal{H}}$ conceals a polynomial dependence on $\mathtt{VC}(\mathcal{H})$.

1 Introduction

A basic goal in machine learning is to develop a predicting model from labeled examples (i.e., training data) that can reliably generalize to unseen examples (i.e., test data). In its simplest form, namely, binary classification, a learner Lrn is given a training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, usually assumed to be i.i.d. samples drawn from an unknown distribution D of labeled examples where x_i's are the domain instances (or data points) and $y_i \in \{0, 1\}$ are the labels. The aim is to produce a mapping $h = \mathsf{Lrn}(\mathcal{S})$ that predicts the labels of fresh examples $(x, y) \sim D$ as accurately as possible, i.e., to minimize the population loss $L_D(h) = \Pr_{(x,y)\sim D}[h(x) \neq y]$. This classical setting has been extensively studied in the last half a century. This accumulated work resulted in fundamental mathematical characterizations regarding the nature of learnability when the training samples are truly i.i.d without any tampering by an adversary [Shalev-Shwartz and Ben-David, 2014]. The goal of this paper is to offer a similar characterization in the presence of an adversary who can tamper with a subset of the training data.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

With the emergence of sensitive machine learning applications, it is critical to ensure the trustworthy of such predictive models in the non-ideal scenarios. In this paper, we consider *robust* learnability when the training examples can be altered by an adversary whose goal is to make sure that a target test point will be predicted incorrectly. For instance, consider an adversary who aims to fool a self-driving car to speed up once it observes a stop sign. If such an adversary can somehow influence the training sets used for training the decision rules, she has all the reasons to strategically change them with the specific goal of misleading the self-driving car. As another example, consider a loan applicant who wants to make sure that his loan will be granted. If he can somehow change the training set used by the bank, he might be able to make his application approved. Note that the training set is the lens through which a learning algorithm obtains information about the underlying learning process. Therefore, once we allow the training examples to be tampered with by an adversary, even slightly, unexpected outcomes may take place. To quantify the robustness of learning algorithms, in this paper, we show how much the outcome of a learning algorithm for a particular target test can be trusted once the training set is being altered.

(PAC) learning under instance-targeted poisoning. More formally, we consider an adversary Adv that is allowed to *replace* an η-fraction of the training sample S, resulting to a tampered training sample S' given to the learning algorithm Lrn. Note that even though the training sample S is drawn i.i.d. from a distribution D, the tampered training sample S' does not enjoy this property anymore. Such attackers are also called *poisoning* adversaries [Barreno, Nelson, Sears, Joseph, and Tygar, 2006], and variants of them are previously studied under the name of *malicious* noise [Valiant, 1985, Kearns and Li, 1993] or *nasty* noise [Bshouty, Eiron, and Kushilevitz, 2002]. More specifically, we study poisoning settings in which the adversarial perturbation of the original sample S *can also depend on the final test instance* x. Due to the adversary's knowledge of the target test point x, such poisoning attacks are sometimes referred to as *instance-targeted* poisoning attacks [Barreno, Nelson, Sears, Joseph, and Tygar, 2006]. Even without any manipulation to the training set, it is too much to ask the learning algorithm to predict correctly all the time while given only a finite number of examples to learn from. In the same vein, we can only hope to design a robust learning algorithm that is correct with high probability over the selection of $(x, y) \sim D$, especially if the adversary knows the test instance (x, y) before manipulating the training set S to S'. Gao, Karbasi, and Mahmoody [2021], building on ideas from [Levine and Feizi, 2020], proved that PAC learnability under instance-targeted poisoning attacks is achievable only when $\eta = o(1)$. In other words, when the adversary can only change a *sublinear* $o(n)$ number of n examples, then the optimal learner can achieve error $o(1)$ that goes to zero when the number of examples n goes to infinity.

1.1 Our Results

The prior work leaves several key questions open on the exact parameters of learnability under instance-targeted poisoning. Most importantly, the work of Gao, Karbasi, and Mahmoody [2021] does not quantify the error rate when the adversary's budget is $\eta = \Omega(1)$ (e.g., if the adversary can corrupt $n/100$ of the examples). Secondly, Gao, Karbasi, and Mahmoody [2021] only assume the realizable setting as it is crucial for their results that all the "sub-models" trained using the bagging technique will have error that goes to *zero*. Hence, the question of finding optimal learning rates is left open for both realizable and agnostic settings. Finally, as the developed robust algorithms are all based on "bagging" they are inherently improper learning technique.

In this work, we make progress on all the directions above and achieve optimal error rates (up to constant factors) for general η, both for the realizable and agnostic settings. We further study the proper nature of the obtained algorithms and give the first proper learning methods that are robust against instance-targeted poisoning attacks for natural hypothesis classes such as linear classifiers.

More precisely, we give a characterization of the optimal error rate of learning under instance-targeted poisoning attacks with budget $\eta \cdot n$ as follows.

Realizable setting. We show that the optimal error is $\Theta(\eta \cdot d)$ where d is the VC dimension of the hypothesis set $\mathcal{H}$. To prove this, we first present an upper bound, showing that a (deterministic) learner can guarantee the error to be at most $O(\eta d)$ under any instance-targeted poisoning attacks of budget ηn. We then also show a matching lower bound (up to a constant factor) as follows. For any

nontrivial[1] hypothesis class of VC dimension d, we show how to design a distribution D over the examples such that no matter how the learning proceeds, there always exists an adversary of budget ηn that can increase the error (under the instance-targeted attack) to $\Omega(\eta d)$. Our lower bound above holds even if the learning algorithm uses *private randomness* that is not known to the adversary[2]. Our positive result, however, is deterministic, and so can be seen as satisfying the *strong* learning guarantee, in which the adversary's perturbations to the training set is allowed to depend on the learner's randomness.

Agnostic setting. We also extend our result above to the agnostic setting in which all hypotheses $h \in \mathcal{H}$ have population loss bounded away from zero (even before the attack). In this setting, we devise a deterministic algorithm whose expected error on the test point is $O(\mathsf{OPT} + \eta \cdot d)$, where OPT is the population loss of the best hypothesis $h \in \mathcal{H}$.

A natural question that arises is whether one can achieve an additive regret guarantee of $\mathsf{OPT} + O(\eta \cdot d)$? (Note that agnostic learning is usually defined with respect to additive regret). We show that this is in fact *not possible*, at least for deterministic learners, by presenting a negative result. In particular we show that for any deterministic learner Lrn, there is an extremely simple hypothesis class (just consisting of two functions) and an input distribution such that the learner is forced to have adversarial error $\geq 2\mathsf{OPT}$. This negative result uses tools from the computational concentration of products [Talagrand, 1995] and a continuity intermediate-value argument.

Proper learning. The deterministic algorithm witnessing the above upper bound is inherently improper which might be a disadvantage in terms of interpretability or test-time computational complexity. In contrast, in (the non-adversarial) PAC setting proper algorithms are known to achieve near optimal learning rates (up to log factors). We therefore explore the cost of proper learning under instance-targeted poisoning attacks. We show that in many natural classes, such as half spaces, it is indeed possible to obtain proper learning rules that are robust to instance-targeted poisoning attacks, with guarantees which are only polynomially worse than optimal. For example, for the class of half-spaces in $\mathbb{R}^d$ we derive a deterministic proper learning rule whose error rate is at most $O(d^3\eta)$. At a technical level, we achieve this result by relying on the *projection number* of the class [Bousquet, Hanneke, Moran, and Zhivotovskiy, 2020, Kane, Livni, Moran, and Yehudayoff, 2019, Braverman, Kol, Moran, and Saxena, 2019].

1.2 Relation to Certification and Stability

Certification. Robustness to instance-targeted poisoning boils down to the following type of stability: on most of the test instances x, the prediction of the learner $y = y(x)$ remains the same even if at most η fraction of the examples in the training-set S are replaced. It is natural to require the learning rule to *certify* this stability. That is, a certifying learning rule provides a bound $k = k(x)$ along with the prediction label $y = y(x)$, where the meaning of k is that the prediction $y = y(x)$ remains the same even if at most k examples in the input sample are replaced. Note that it is always possible to provide the trivial guarantee of $k = 0$, and therefore the goal is to design robust learners that provide non-trivial certificates. Our algorithm naturally achieves that: for $\approx 1 - \varepsilon$ of the test instances x it provides a guarantee of $k \approx \eta n$.

Connection to stability. We also present a new perspective on instance-targeted poisoning attacks by showing how they can be seen as natural forms of algorithmic stability [Bousquet and Elisseeff, 2002, Rakhlin, Mukherjee, and Poggio, 2005]. In particular, we show that one can study the adversarial robustness (around the *true* label) to instance-targeted poisoning by decoupling the (pure) stability aspect (which does not depend on the true labels) from the (non-adversarial) risk. We refer to the former as the *prediction stability*. Roughly speaking, prediction stability requires that the model's prediction on x does not change even if the adversary changes the training set withing its budget ηn. Note that here we do not care whether the model's output on x is the correct label or not, and hence is a pure measure of stability of the predictions.

[1] A non-trivial class $\mathcal{H}$ is one for which there are $x_1, x_2 \in \mathcal{X}$ and $h_1, h_2 \in \mathcal{H}$ so that $h_1(x_1) = h_2(x_1)$ and $h_1(x_2) \neq h_2(x_2)$. In particular, any class containing at least 3 hypotheses is non-trivial.

[2] This model is referred to as the "weak" learning model (under instance-targeted poisoning attacks) in the work of Gao, Karbasi, and Mahmoody [2021].

It might be helpful to compare prediction stability with the algorithmic stability of [Bousquet and Elisseeff, 2002, Rakhlin, Mukherjee, and Poggio, 2005]. The later requires that for a typical sample S of size n, and for every *fixed* $i \in [n]$, the prediction of the model trained on S and tested on a random test-point x is likely not changed if one substitutes the i-th example in S with a *fresh* random example. Prediction stability strengthens this condition in two ways: (1) the choice of what coordinate in S to change can adversarially depend on the test instance x, (2) the adversary is allowed to change *more* than one examples (i.e., up to $\eta \cdot n$).

1.3 Related Work

Poisoning attacks have a history in theoretical learning and are studied under various noise models [Valiant, 1985, Kearns and Li, 1993, Sloan, 1995, Bshouty, Eiron, and Kushilevitz, 2002]. However, these works focus on the *non-targeted* setting in which the adversary does *not* know the target instance. More recently, starting with [Diakonikolas, Kamath, Kane, Li, Moitra, and Stewart, 2016, Lai, Rao, and Vempala, 2016] (also see the survey [Diakonikolas and Kane, 2019]), the *computational* aspects of efficient learning under (non-targeted) poisoning is studied. In contrast, our work focuses on instance-targeted poisoning, and we study the learning rates *information theoretically* regardless of learner's computing power. The work of Steinhardt, Koh, and Liang [2017] further studied the *certification* of the overall (non-targeted) error. More recently, such (non-targeted) poisoning attacks are combined with *test-time* attacks and are studied under the name of *backdoor* attacks [Gu, Dolan-Gavitt, and Garg, 2017, Ji, Zhang, and Wang, 2017].

Besides instance-targeted attacks (which are the focus of this paper), other notions of targeted attacks were studied in the literature: for example, in *model-targeted* attacks, the adversary's goal is to make the learner predict according to a specific model. Recent works on this model include [Farhadkhani, Guerraoui, Hoang, and Villemaud, 2022, Suya, Mahloujifar, Suri, Evans, and Tian, 2021].

Most relevant to our setting are the recent works of Gao, Karbasi, and Mahmoody [2021], Blum, Hanneke, Qian, and Shao [2021] where the general problem of learning (and more quantitative variant of learning error rate) under *instance-targeted* poisoning was formally defined and studied. In particular, Blum, Hanneke, Qian, and Shao [2021] studied learnability under instance-targeted poisoning where the adversary can add an *unbounded* number of so-called clean-label examples to the training set. A clean-label example (x, y) has the property that y is the *correct* label of x, while x could be an arbitrary instance that is *not* sampled from the same distribution that generates other instances in the training set. Gao, Karbasi, and Mahmoody [2021] also showed that when the adversary's corruption is only an $o(1)$ fraction of the training set, PAC learning is possible (if it is possible without the attack). In a concurrent work, Balcan, Blum, Hanneke, and Sharma [2022] study the problem of certifying the *correct* prediction even under instance-targeted data poisoning. In this work, however, our methods can be used to obtain certification of the stability of the model (as also done in [Gao, Karbasi, and Mahmoody, 2021]) around their prediction (even though the prediction might *not* be true always), while controlling the overall error to be provably small (again under the instance-targeted attack).

Rosenfeld, Winston, Ravikumar, and Kolter [2020] empirically demonstrated that randomized smoothing [Cohen, Rosenfeld, and Kolter, 2019] can provide robustness against label-flipping attacks, in which the adversary is limited to merely flipping the label of a subset of the training set. They also showed that randomized smoothing can be used to handle *replacing* attacks (the model also studied in this paper), in which the adversary substitutes a part of the training set with a new set of same size. Subsequently, Levine and Feizi [2020] used deterministic methods that further allowed attacks that can add examples to or remove them from the training set. Chen, Li, Wu, Sheng, and Li [2020], Weber, Xu, Karlas, Zhang, and Li [2020], Jia, Cao, and Gong [2020] further developed the technique of randomized bagging/sub-sampling for the goal of resisting instance-targeted poisoning attacks.

Finally, we comment that other theoretical works have also studied instance-targeted poisoning attacks [Mahloujifar and Mahmoody, 2017, Etesami, Mahloujifar, and Mahmoody, 2020]. These works show how to *amplify* error for specific test instances, say from 0.01 error to 0.5, through instance-targeted poisoning. In particular, these works do not talk about the *fraction* of the test population that is vulnerable to targeted poisoning. The work of Shafahi, Huang, Najibi, Suciu, Studer, Dumitras, and Goldstein [2018] studied the power of such attacks empirically.

2 Preliminaries

Notation and basic learning theory definitions. We consider the setting of binary classification. Let $\mathcal{X}$ denote the input domain and $\mathcal{Y} = \{0, 1\}$ denote the label-set. A pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is called an *example*. A sequence $S = (x_1, y_1), \ldots, (x_n, y_n) \in (\mathcal{X} \times \mathcal{Y})^n$ of n examples is a *sample* of size n. The i'th example in S is denoted by S_i.

A function $h\colon \mathcal{X} \to \mathcal{Y}$ is called an hypothesis or a concept. A set of hypotheses $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is called an *hypothesis class*, or a *concept class*. We denote the VC-dimension of a concept class $\mathcal{H}$ by $d = d(\mathcal{H})$.

For a set Z, let $Z^* = \cup_n Z^n$ denote the set of all finite sequences with elements from Z. A *learning rule* or *learning algorithm* or *learner* $\mathsf{Lrn}\colon (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{X}^{\mathcal{Y}}$ is a deterministic[3] mapping which takes an input sample $S \in (\mathcal{X} \times \mathcal{Y})^*$ and maps it to a hypothesis $\mathsf{Lrn}(S) = h \in \mathcal{X}^{\mathcal{Y}}$. If it is guaranteed that $\mathsf{Lrn}(S) \in \mathcal{H}$ for all input samples S then Lrn is said to be *proper*; otherwise, it is *improper*.

Let D be a distribution over examples, and let h be an hypothesis. The *population loss* of h with respect to D is defined by $L_D(h) = \Pr_{(x,y)\sim D}[h(x) \neq y] = \mathbb{E}_{(x,y)\sim D}[1[h(x) \neq y]]$. A distribution D is said to be *realizable* by $\mathcal{H}$ if $\inf_{h\in\mathcal{H}} L_D(h) = 0$. Similarly, for a sample S, let $L_S(h) = \frac{1}{|S|}\sum_{i=1}^{n} 1[h(x_i) \neq y_i]$ denote the *empirical error* of h with respect to S, and call a sample realizable by a class $\mathcal{H}$ if there exists $h \in \mathcal{H}$ such that $L_S(h) = 0$. The expected loss (also called risk) of a learning algorithm Lrn w.r.t a distribution D and sample size n is defined by

$$\varepsilon_n(\mathsf{Lrn}|D) := \Pr_{S\sim D^n,(x,y)\sim D}\left[\mathsf{Lrn}(S)(x) \neq y\right].$$

The function $n \mapsto \varepsilon_n(\mathsf{Lrn}|D)$ is called the *learning curve*, or *learning rate* of Lrn w.r.t D.

For a real number r, let $\lfloor r \rceil$ denote the nearest integer to r. In case of ties, when $r = k + 1/2$ for some $k \in \mathbb{Z}$, then define $\lfloor r \rceil = k + 1$. For any finite multiset $\mathcal{H}' \subset \mathcal{H}$, denote by $\mathsf{Maj}(\mathcal{H}')$ the function defined for all $x \in \mathcal{X}$ by

$$\mathsf{Maj}(\mathcal{H}')(x) = \left\lfloor \frac{1}{|\mathcal{H}'|} \sum_{h'\in\mathcal{H}'} h'(x) \right\rceil.$$

Adversarial risk and prediction stability. Before we introduce the definition of Adversarial risk, we define *Hamming distance* between samples, which is a natural way to quantify distance between samples of equal size.

Definition 2.1 (Hamming distance between samples)**.** Fix $n \in \mathbb{N}$ and let $S, S' \in (\mathcal{X} \times \mathcal{Y})^n$. We define the *Hamming distance* between S and S', denoted by $\mathsf{d_H}(S, S')$ as follows:

$$\mathsf{d_H}(S, S') = \sum_{i=1}^{n} 1[S_i \neq S'_i].$$

Note that the Hamming distance is defined only for samples of equal sizes. If $\mathsf{d_H}(S, S') \leq \eta \cdot n$, we say that S, S' are η-close. For any sample S, let

$$B_\eta(S) := \left\{S' : \mathsf{d_H}(S, S') \leq \eta \cdot n\right\}.$$

Definition 2.2 (η-adversarial risk)**.** Let $\eta \in (0, 1)$ be the adversary's budget, let Lrn be a learning rule, and let D be a distribution over examples. The *η-adversarial risk* of Lrn w.r.t D and sample size n is defined by

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D, \eta) := \Pr_{S\sim D^n,(x,y)\sim D}\left[\exists S' \in B_\eta(S) : \mathsf{Lrn}(S')(x) \neq y\right].$$

Thus, robust learning with respect to instance-targeted data poisoning with budget η boils down to minimizing the adversarial risk. Indeed, given an input sample S and a test example (x, y), an adversary with budget γ can force a mistake on x if and only if there exists $S' \in B_\eta(S)$ such that $\mathsf{Lrn}(S')(x) \neq y$.

[3] In Appendix B, we extend the definition in a way that captures also a family of randomized learners.

Randomness. In Definition 2.2 above we define adversarial risk for the setting in which both the learner Lrn and the model $h = \mathsf{Lrn}(S)$ are *deterministic*. When either Lrn or h is allowed to use randomness, then the notion of adversarial risk as defined in Definition 2.2 can be extended in several ways, depending on whether the adversary can see the randomness of the learner or not. Some of these variations are discussed in the work of Gao, Karbasi, and Mahmoody [2021]. We remark however that our results in the realizable setting apply to all variations. This is simply because our upper bounds are achieved by deterministic learners, whereas our lower bound uses the weakest type of an adversary (which does not depend on the randomness of the learner). In contrast, our lower bound in the agnostic setting applies only to deterministic learners.

Explicit bounds. We do not try to optimize the constants hidden in the $O(\cdot), \Omega(\cdot)$ notation in the derived bounds. The reason is because on the one hand, this way the proofs are simpler and more accessible, and on the other hand, we do not know how to get tight (or nearly tight) lower and upper bounds on the constants. Obtaining tight bounds is a natural direction for future research; we elaborate on this in Section 5. Nevertheless, the complete proofs (which are given in the appendix) include explicit numerical bounds on the constants.

Decoupling adversarial risk into stability and risk. It is convenient and illustrative to decouple robust learnability to two properties: small expected loss and prediction stability. The latter means that the prediction of the learning algorithm on a random test point is stable under replacing a bounded amount of examples from the training set:

Definition 2.3 (Prediction stability)**.** Let $n \in \mathbb{N}$, $\sigma, \eta \in (0,1)$. Let Lrn be a learning rule and D be a distribution over examples. We say that the learning rule Lrn is (n, σ, η)*-prediction stable with respect to* D if the following holds

$$\lambda_n(\mathsf{Lrn}|D, \eta) := \Pr_{S \sim D^n, x \sim D_x} [\exists S' \in B_\eta(S) : \mathsf{Lrn}(S')(x) \neq \mathsf{Lrn}(S)(x)] \leq \sigma.$$

where D_x is the marginal distribution induced by D on the domain $\mathcal{X}$.

Of course, prediction stability alone does not guarantee robust learning. Indeed, useless learning rule that always outputs the all 0's classifier has maximal stability. At the very least, the learning rule should learn the class in the classical sense (in the absence of an adversary). The following observation asserts that prediction-stable learning rules with small loss are robust learners:

Observation 2.4 (Prediction stability + small error = robust learning)**.** *Let* Lrn *be a learner and* D *a distribution over examples. Then,*

$$\max\{\varepsilon_n(\mathsf{Lrn}|D), \lambda_n(\mathsf{Lrn}|D, \eta)\} \leq \varepsilon_n(\mathsf{Lrn}|D, \eta) \leq \lambda_n(\mathsf{Lrn}|D, \eta) + \varepsilon_n(\mathsf{Lrn}|D).$$

In other words, if Lrn is (n, σ, η)-prediction stable with respect to D whose expected population loss is $\varepsilon_n(\mathsf{Lrn}|D) \leq \varepsilon$. Then Lrn learns D with an adversarial expected loss $\sigma + \epsilon$. Conversely, if $\varepsilon_n(\mathsf{Lrn}|D, \eta) \leq \varepsilon$ then Lrn is (n, ε, η)-prediction stable with respect to D and its expected population loss is also $\varepsilon_n(\mathsf{Lrn}|D) \leq \varepsilon$. We leave the (simple) proof of Observation 2.4 to the reader.

3 Realizable Setting

Theorems 3.1 and 3.3 below characterize the optimal adversarial risk in the realizable setting.

Theorem 3.1 (Realizable case – positive result)**.** *There exists a constant* $c_1 > 0$ *so that the following holds. Let* $\mathcal{H}$ *be a hypothesis class with VC dimension* d *and let* $\eta \in (0,1)$*. Then there exists a learner* Lrn *having* η*-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D, \eta) \leq c_1 \eta d$$

for any distribution D *realizable by* $\mathcal{H}$ *and for any sample size* $n \geq 1/\eta$*.*

We prove Theorem 3.1 in Appendix A.

Note that the requirement that the sample size is $n \geq 1/\eta$ is necessary since otherwise $\eta \cdot n < 1$, which means that the adversary cannot modify the input sample, and so this case reduces to classical learning without an adversary.

SPV: STABLE PARTITION AND VOTE

Input: Stability parameter $\eta \in (0,1)$, a learning algorithm Lrn and an input sample $S \sim D^n$ where $n \geq 1/\eta$.
Output: A classifier $h\colon \mathcal{X} \to \mathcal{Y}$.

1. Partition S into $\lceil 7\eta n \rceil$ consecutive subsamples such that all first $t = \lfloor 7\eta n \rfloor$ subsamples are of size at least $\frac{1}{7\eta}$. Denote the i'th subsample by $S^{(i)}$.
2. For all $i \in [t]$, run the learning algorithm Lrn on $S^{(i)}$ to obtain a hypothesis $h_i = \mathsf{Lrn}(S^{(i)})$.
3. Return the hypothesis h defined as follows for all $x \in \mathcal{X}$:

$$h(x) = \mathsf{Maj}\left(\{h_1, \ldots, h_t\}\right)(x).$$

Figure 1: SPV - A meta algorithm implementing a stable version of the input learning algorithm Lrn.

Theorem 3.1 is proven using the STABLE PARTITION AND VOTE (or SPV, for short) meta-algorithm, described in Figure 1. The meta-algorithm is based on the idea of partitioning and then voting used in [Gao, Karbasi, and Mahmoody, 2021], but with a more refined and precise analysis. The partition and vote technique works as follows. First, partition the input sample to subsamples of a carefully chosen size. Then, train a given learner (which is called the *input learner* of SPV) on each subsample, and finally let the trained learners vote to determine the output label. The size of each subsample trades-off, in a way, expected loss and prediction-stability: if it is too small, the given learner will perform poorly on each subsample. On the other hand, if it is relatively large then the number of learners that participate in the majority vote is small and the adversary can poison a large fraction of these learners and flip the overall majority vote. We elaborate on this when proving Theorem 3.1. Notice that the time complexity of SPV is proportional to the time complexity of the input learner Lrn.

To state the complementing impossibility result, we need the following definition of *non-trivial concept classes* [Bshouty, Eiron, and Kushilevitz, 2002].

Definition 3.2 (Non-trivial concept classes)**.** Let $\mathcal{H}$ be a concept class over a domain $\mathcal{X}$. We say that $\mathcal{H}$ is *non-trivial*, if there are $x_1, x_2 \in \mathcal{X}$ and $h_1, h_2 \in \mathcal{H}$ so that $h_1(x_1) = h_2(x_1)$ and $h_1(x_2) \neq h_2(x_2)$.

Theorem 3.3 (Realizable case – impossibility result)**.** *There exists a constant $c_2 > 0$ so that the following holds. Let $\mathcal{H}$ be a non-trivial hypothesis class with VC dimension d and let $\eta \in (0,1)$. Then, there exists a distribution D realizable by $\mathcal{H}$, so that every learner* Lrn *has η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D, \eta) \geq \min\{c_2 \eta d, 1/100\}$$

for any sample size $n \geq 1/\eta$.

We note that this impossibility result applies also to a variety of randomized learners; we elaborate on this in Appendix B, where we also prove Theorem 3.3.

The above lower bound demonstrates how vulnerability to instance-targeted attacks depends greatly on the hypothesis class we want to learn, and specifically on its VC-dimension.

3.1 Certification

Besides prediction-stability, another useful property our SPV meta-algorithm has is the ability to efficiently calculate and output a *certificate* for the stability of its predictions. Formally, given an input sample S, a certificate is a function $\eta_S : \mathcal{X} \to [0,1]$, outputted by a learner in addition to its output hypothesis h_S such that the following is satisfied: $h_S(x) = h_{S'}(x)$ for every point x and for every input sample S' which is $\eta_S(x)$-close to S. If one ignores computational considerations, outputting optimal certificates is always possible:

Definition 3.4 (Optimal Certificate)**.** Let Lrn be any learning rule, and let S be an input sample. Define the *optimal certificate* $\eta^\star(\cdot) = \eta^\star(\cdot|S)$ of Lrn on input sample S as follows. The optimal

PSPV: PROPER STABLE PARTITION AND VOTE

Input: Stability parameter $\eta \in (0,1)$, a proper learning algorithm Lrn_p and an input sample $S \sim D^n$ where $n \geq 1/\eta$.
Output: A classifier $h \in \mathcal{H}$.

1. Partition S into $\lceil 5k_p\eta n \rceil$ consecutive subsamples such that all first $t = \lfloor 5k_p\eta n \rfloor$ subsamples are of size at least $\frac{1}{5k_p\eta}$. Denote the i'th subsample by $S^{(i)}$.
2. For all $i \in [t]$, train Lrn_p on $S^{(i)}$ to obtain a hypothesis $h_i = \mathsf{Lrn}_p(S^{(i)})$.
3. Return $h \in \mathcal{H}$ such that
$$h(x) = \mathsf{Maj}\left(\{h_1, \ldots, h_t\}\right)(x)$$
holds for all $x \in \mathcal{X}_{\{h_1,\ldots,h_t\},2k_p}$.

Figure 2: PSPV - A meta-algorithm that implements a stable version of the input proper learning algorithm Lrn_p and maintains properness.

certificate $\eta^\star(x|S)$ is equal to $\frac{k}{n}$ where k is the largest integer for which $\mathsf{Lrn}(S')(x) = \mathsf{Lrn}(S)(x)$ for every sample S' with hamming distance at most k from S.

In other words, if S is a sample that was corrupted by an adversary with budget η such that $\eta \leq \eta^\star(x|S)$ then the output label $\mathsf{Lrn}(S)(x)$ is equal to the label that would have been outputted if the learner was trained with the uncorrupted sample.

The issue with the optimal certificate $\eta^\star(x)$ is that it can be impossible to compute as it requires to iterate over the potentially infinite space of all samples S' of hamming distance at most $n \cdot \eta(x)$ from the input sample S. In contrast, our SPV learner can *efficiently* calculate a non-trivial lower bound on $\eta^\star$ which therefore also serves as a certificate. The key property which enables this is the fact that its output hypothesis is the majority vote of base learners, each trained on a *disjoint* subsample. This is summarized in the following proposition:

Proposition 3.5. *Consider a learner whose output hypothesis is given by a majority vote of t learners $L_1, \ldots, L_t$ that are trained on t disjoint subsamples $S_1, \ldots, S_t$ of the input sample S. Define*
$$\eta(x|S) = \frac{1}{n} \cdot \Big(\frac{\sum_{i\in[t]} 1[h_i(x) = y] - \sum_{i\in[t]} 1[y_i \neq y]}{2} - 1 \Big),$$
where h_i is the output hypothesis of L_i, y is the output label of the majority vote of the L_i's, and n is the size of the input sample S. Then, $\eta(x|S) \leq \eta^\star(x|S)$.

Proof. Notice that $n \cdot \eta(x|S) + 1$ is equal to the minimal number of h_i's whose prediction on x must be flipped in order to enforce that
$$|\{i : h_i(x) = y\}| \leq |\{i : h_i(x) \neq y\}|.$$
Therefore, at least one example in each S_i such that $h_i(x) = y$ must be replaced in order to change the prediction of $\mathsf{Lrn}(S)$ on x. In particular, if only $n \cdot \eta(x|S)$ examples are replaced than the prediction of $\mathsf{Lrn}(S)$ on x remains the same. This implies that $\eta^\star(x|S) \geq \eta(x|S)$ as stated. □

In light of Proposition 3.5, our SPV learner can efficiently compute and output a certificate $\eta(x)$ which is proportional to η (where η is the stability parameter given to SPV), with probability proportional to the expected loss of the input learner given to SPV when executed on a sample of size $\left\lceil \frac{1}{7\eta} \right\rceil$.

3.2 A Proper Variant of SPV

We now present a proper version of SPV for classes $\mathcal{H}$ with a finite *projection number*, described in Figure 2. The projection number of a concept class $\mathcal{H}$ is denoted by $k_p = k_p(\mathcal{H})$ (we present its definition after the statement of Theorem 3.6 below). In particular, for the class of halfspaces it yields a robust learner with the following guarantee:

Theorem 3.6. *There exists a constant $c > 0$ so that the following holds. Let $\mathcal{H}$ be the class of halfspaces over $\mathbb{R}^d$ for some $d \geq 1$, and let $\eta \in (0,1)$. Then, there exists a proper learner* Lrn *having η-adversarial risk*

$$\varepsilon_n(\mathsf{Lrn}|D,\eta) \leq c\eta d^3$$

for any distribution D realizable by $\mathcal{H}$ and for any sample size $n \geq 1/\eta$.

The proof of Theorem 3.6 is deferred to Appendix C.

To derive Theorem 3.6, we reinforce the SPV algorithm with a technique introduced by Kane, Livni, Moran, and Yehudayoff [2019] and further developed by Bousquet, Hanneke, Moran, and Zhivotovskiy [2020]. This technique allows in certain cases to *project* a majority vote of hypotheses from the class $\mathcal{H}$ back to $\mathcal{H}$. Its applicability hinges on a combinatorial parameter called the *projection number*. The PSPV learner explicitly uses the projection number, so for completeness we give its definition below. The interested reader may look into the work of Bousquet, Hanneke, Moran, and Zhivotovskiy [2020] for an insightful discussion on the role of the projection number in proper learning.

Definition 3.7 (Projection Number)**.** Let $\mathcal{H}$ be a concept class. For any $\ell \geq 2$ and for any multiset $\mathcal{H}' \subset \mathcal{H}$ define the set $\mathcal{X}_{\mathcal{H}',\ell}$ to be the set of all $x \in \mathcal{X}$, for which the number of hypotheses in $\mathcal{H}'$ that disagree with $\mathsf{Maj}(\mathcal{H}')(x)$ is less than $|\mathcal{H}'|/\ell$. The Projection Number of the class $\mathcal{H}$, denoted $k_p = k_p(\mathcal{H})$, is defined to be the smallest ℓ so that for any finite multiset $\mathcal{H}' \subset \mathcal{H}$, there exist $h \in \mathcal{H}$ such that $h(x) = \mathsf{Maj}(\mathcal{H}')(x)$ for all $x \in \mathcal{X}_{\mathcal{H}',\ell}$. If no such ℓ exists then $k_p = \infty$.

4 Agnostic Setting

In this section, we extend the results on robust learnability to the agnostic case. First, by a simple generalization of the positive result for the realizable case, we provide a robust semi-agnostic learner. That is, our learner has adversarial risk depending linearly on $\mathsf{OPT} = \mathsf{OPT}(\mathcal{H}, D) := \min_{h\in\mathcal{H}} L_D(h)$. While semi-agnostic learning is considered not ideal in many cases, we complement our positive result by showing that semi-agnostic learning is unavoidable when the goal is to design a robust and deterministic (as ours) learner for the agnostic setting.

4.1 A Semi-agnostic Learner

Formally, a semi agnostic learner is defined as follows. Let $c \in \mathbb{R}$. A learning rule Lrn is a *c-semi agnostic learner* if the following holds. Let $\mathcal{H}$ be a concept class and let D be a distribution over examples. Then there exists an *excess error rate* $\varepsilon^{\mathsf{Agn}} : \mathbb{N} \to [0,1]$ such that $\varepsilon_n(\mathsf{Lrn}|D) \leq c\mathsf{OPT} + \varepsilon^{\mathsf{Agn}}(n)$ where $\mathsf{OPT} = \inf_{h\in\mathcal{H}} L_D(h)$.

Before stating our positive result in this setting, we first discuss how achieving adversarial risk $O(d(\mathsf{OPT} + \eta)$ is possible by reduction to the realizable setting.

Reduction to the realizable setting. Suppose a learner is given a training set S' of size n that comes with ηn replacements made by the adversary on the original set S. Moreover, suppose that S is sampled from a distribution D such that the best $h \in \mathcal{H}$ has OPT error on D. This means that, roughly OPT fraction of S does *not* match to h. Therefore, one can see S' as first sampled from D (without noise) followed by $\approx (\eta + \mathsf{OPT}) \cdot n$ replacement corruptions. This way, one can employ a learner that can tolerate $\eta' = \eta + \mathsf{OPT}$ fraction of adversarial corruptions in the realizable setting and obtain total adversarial risk $O(d(\mathsf{OPT} + \eta)$.

The above discussion raises a natural question: can a learner achieve adversarial risk $O(\mathsf{OPT} + d\eta)$ or even (ideally) $\mathsf{OPT} + O(d\eta)$? The latter is the typical type of risk bound in agnostic settings, where there is no multiplicative dependence on OPT in the risk.

The following theorem, which we prove in Appendix D states the positive result.

Theorem 4.1 (Positive result for the agnostic case)**.** *There exist constants c_1, c_2 so that the following holds. Let $\mathcal{H}$ be a hypothesis class with VC dimension d and let $\eta \in (0,1)$. Then, there exists a learner* Lrn *having η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \leq c_2 \cdot \mathsf{OPT} + c_1 \cdot d \cdot \eta$$

for any distribution D over examples and for any sample size $n \geq 1/\eta$.

As in the realizable case upper bound, the above upper bound is proved by using the SPV meta-learner. The main difference is that to prove this result we use a different input learner Lrn given to SPV than the one we use in the realizable case.

4.2 Ruling Out Agnostic Learning

Note that Theorem 4.1 only proves the existence of a *semi*-agnostic learner under instance-targeted poisoning. A more desirable goal would be to obtain (standard) *agnostic* learners whose error under the attack is $\mathsf{OPT} + \psi$ where ψ is a vanishing (additive) error term when $\eta \to 0$. Here we will prove that at least when it comes to *deterministic* learners, such a goal is out of reach, and the best we can hope for is $2\mathsf{OPT}$ plus additive terms that depend on η and the VC dimension. This explains why we can only achieve a semi-agnostic learner.

The following theorem, which we prove in Appendix E shows that in Theorem 4.1, the constant c_1 needs to be at least 2, and so the standard way agnostic learners bound their regret is not possible for instance-targeted poisoning.

Theorem 4.2 (Impossibility of agnostic learning)**.** *Let $\eta' \in (0,1), n \in \mathbb{N}$. For any hypothesis class $\mathcal{H}$ that has at least two hypotheses and for any deterministic learner, there is a distribution D over (two) examples and $\eta = \eta' + \widetilde{O}(1/\sqrt{n})$ such that* Lrn *has η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \geq 2\mathsf{OPT} + \Omega(\eta') - O(1/n).$$

5 Conclusion and Open Questions

In this work, we studied the optimal rate of learning for binary classification problems under instance-targeted poisoning. We showed that in the realizable setting the error rate can be characterized up to a constant factor and is proportional both to adversary's budget and the VC dimension of the class. In the agnostic setting, we proved a perhaps surprising lower bound that standard agnostic learning (with additive regret compared to the optimal error in the no-attack setting) is impossible for deterministic learners, and also complemented this with a positive result using a *semi-agnostic* learner. We also showed how to make our learners proper in a variety of interesting settings.

Our work leaves a few interesting directions for future research.

- **Finding the exact constant in the realizable case.** Our results in the realizable case characterize the optimal adversarial risk up to a constant multiplicative factor in the sense that there exist constants c_1, c_2 so that achieving η-adversarial risk of $c_1\eta d$ is possible for any hypothesis class with VC-dimension d, whereas obtaining η-adversarial risk of $c_2\eta d$ can't be achieved for any hypothesis class with VC-dimension d. However, there is a large gap between c_1, c_2. Can we close or shrink this gap?
- **Finding the correct multiplicative factor in the agnostic case.** Our results show that in the agnostic case, there must be a constant $C \geq 2$ so that the best adversarial risk attainable is $C \cdot \mathsf{OPT}$. What is the value of C?
- **Characterizing proper robust learning.** In the proper and realizable case, our stable learner for linear classifiers depends on d^3, while our lower bound depends linearly on d, as in the general improper case. It remains open to identify the correct dependence on d.
- **Characterizing the role of randomness.** Our impossibility result for the agnostic learning (Theorem 4.2) only applies to deterministic learners. It remains open to either effectively use randomness during the learning (known or unknown to the adversary) and obtain an agnostic learner, or to extend the negative result to cover such randomized learners as well.

Acknowledgments

Amin Karbasi acknowledges funding in direct support of this work from NSF (IIS-1845032), ONR (N00014- 19-1-2406), and the AI Institute for Learning-Enabled Optimization at Scale (TILOS). Mohammad Mahmoody is supported by NSF grants CCF-1910681 and CNS1936799. Shay Moran is a Robert J. Shillman Fellow; he acknowledges support by ERC grant 802599, by ISF grant 1225/20, by BSF grant 2018385, by an Azrieli Faculty Fellowship, by Israel PBC-VATAT, and by the Technion Center for Machine Learning and Intelligent Systems (MLIS).

References

Maria-Florina Balcan, Avrim Blum, Steve Hanneke, and Dravyansh Sharma. Robustly-reliable learners under poisoning attacks. *arXiv preprint arXiv:2203.04160*, 2022.

Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25. ACM, 2006.

Avrim Blum, Steve Hanneke, Jian Qian, and Han Shao. obust learning under clean-label attack. In *Conference on Learning Theory*, 2021.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Olivier Bousquet, Steve Hanneke, Shay Moran, and Nikita Zhivotovskiy. Proper learning, helly number, and an optimal svm bound. In *Conference on Learning Theory*, pages 582–609. PMLR, 2020.

Mark Braverman, Gillat Kol, Shay Moran, and Raghuvansh R Saxena. Convex set disjointness, distributed learning of halfspaces, and lp feasibility. *arXiv preprint arXiv:1909.03547*, 2019.

Nader H Bshouty, Nadav Eiron, and Eyal Kushilevitz. Pac learning with nasty noise. *Theoretical Computer Science*, 288(2):255–275, 2002.

Ruoxin Chen, Jie Li, Chentao Wu, Bin Sheng, and Ping Li. A framework of randomized selection based certified defenses against data poisoning attacks, 2020.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

Ilias Diakonikolas and Daniel M. Kane. Recent advances in algorithmic high-dimensional robust statistics, 2019.

Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 655–664. IEEE, 2016.

Omid Etesami, Saeed Mahloujifar, and Mohammad Mahmoody. Computational concentration of measure: Optimal bounds, reductions, and more. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 345–363. SIAM, 2020.

Sadegh Farhadkhani, Rachid Guerraoui, Lê-Nguyên Hoang, and Oscar Villemaud. An equivalence between data poisoning and byzantine gradient attacks. In *International Conference on Machine Learning*, pages 6284–6323. PMLR, 2022.

Ji Gao, Amin Karbasi, and Mohammad Mahmoody. Learning and certification under instance-targeted poisoning. In *Uncertainty in Artificial Intelligence*, pages 2135–2145. PMLR, 2021.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

David Haussler, Nick Littlestone, and Manfred K Warmuth. Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.

Yujie Ji, Xinyang Zhang, and Ting Wang. Backdoor attacks against learning systems. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2017.

Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. *arXiv preprint arXiv:2008.04495*, 2020.

Daniel Kane, Roi Livni, Shay Moran, and Amir Yehudayoff. On communication complexity of classification problems. In *Conference on Learning Theory*, pages 1903–1943. PMLR, 2019.

Michael Kearns and Ming Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 665–674. IEEE, 2016.

Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defenses against general poisoning attacks. In *International Conference on Learning Representations*, 2020.

Philip M Long. The complexity of learning according to two models of a drifting environment. *Machine Learning*, 37(3):337–354, 1999.

Saeed Mahloujifar and Mohammad Mahmoody. Blockwise p-tampering attacks on cryptographic primitives, extractors, and learners. In *Theory of Cryptography Conference*, pages 245–279. Springer, 2017.

Alexander Rakhlin, Sayan Mukherjee, and Tomaso Poggio. Stability results in learning theory. *Analysis and Applications*, 3(04):397–417, 2005.

Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Robert H. Sloan. Four Types of Noise in Data for PAC Learning. *Information Processing Letters*, 54 (3):157–162, 1995.

Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3520–3532, 2017.

Fnu Suya, Saeed Mahloujifar, Anshuman Suri, David Evans, and Yuan Tian. Model-targeted poisoning attacks with provable convergence. In *International Conference on Machine Learning*, pages 10000–10010. PMLR, 2021.

Michel Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81(1):73–205, 1995.

Leslie G Valiant. Learning disjunction of conjunctions. In *IJCAI*, pages 560–566, 1985.

Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition. 1974.

Maurice Weber, Xiaojun Xu, Bojan Karlas, Ce Zhang, and Bo Li. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proof of Theorem 3.1 (Realizable Case – Positive Result)

Theorem (Restatement of Theorem 3.1)**.** *There exists a constant $c_1 > 0$ so that the following holds. Let $\mathcal{H}$ be a hypothesis class with VC dimension d and let $\eta \in (0,1)$. Then there exists a learner* Lrn *having η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \leq c_1 \eta d$$

for any distribution D realizable by $\mathcal{H}$ and for any sample size $n \geq 1/\eta$.

To prove Theorem 3.1, we will use the STABLE PARTITION AND VOTE (or SPV for short) meta learner described in Figure 1 with the One-inclusion graph algorithm of Haussler, Littlestone, and Warmuth [1994] as the input learner. First, we prove a more general result on the performance of our SPV meta learner. We denote the algorithm obtained by executing SPV with a learner Lrn as the input algorithm by SPV(Lrn).

Lemma A.1 (General performance of SPV)**.** *Let $\mathcal{H}$ be a concept class, D be a distribution over examples, and* Lrn *be a learning rule. Let also $\eta \in (0,1)$ be the stability parameter given to* SPV *and let $n \geq 1/\eta$ be the sample size. Then* SPV(Lrn) *has η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{SPV}(\mathsf{Lrn})|D,\eta) \leq 6\varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D).$$

Recall that $\varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D)$ is the expected population loss of Lrn *when trained on a sample of size $\lceil 1/(7\eta)\rceil$ from D (in the standard, non adversarial, setting).*

Proof. Let $S \sim D^n$ be the input sample, and $(x,y) \sim D$ be the test example. Note that for all $i \in [t]$ (where $t = \lfloor 7\eta n \rfloor$ is the number of subsamples of size at least $\frac{1}{7\eta}$ in the partition made by SPV) it holds that $\mathbb{E}\big[1[h_i(x) \neq y]\big] \leq \varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D)$. By applying linearity of expectation we get

$$\mathbb{E}\left[\frac{1}{t}\sum_{i=1}^{t} 1[h_i(x) \neq y]\right] \leq \varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D).$$

By Markov's inequality:

$$\Pr\left[\frac{1}{t}\sum_{i=1}^{t} 1[h_i(x) \neq y] \geq 1/6\right] \leq 6\varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D).$$

Let $S' \in B_\eta(S)$. Let $h' = \mathsf{SPV}(\mathsf{Lrn})(S')$, and for all $i \in [t]$ let h'_i be the hypothesis obtained by training Lrn on $S'^{(i)}$. Note that, since S and S' are η-close by, and since $n \geq 1/\eta$ it holds that

$$\frac{1}{t}\sum_{i=1}^{t} 1\left[S^{(i)} \neq S'^{(i)}\right] \leq \frac{\eta n}{\lfloor 7\eta n \rfloor} \leq 1/6.$$

Hence it is implied that $\frac{1}{t}\sum_{i=1}^{t} 1\left[h_i(x) \neq h'_i(x)\right] \leq 1/6$. Thus, the event that $\frac{1}{t}\sum_{i=1}^{t} 1[h'_i(x) \neq y] \geq 1/3$ implies (or, is contained in) the event that $\frac{1}{t}\sum_{i=1}^{t} 1[h_i(x) \neq y] \geq 1/6$, hence,

$$\Pr\left[\frac{1}{t}\sum_{i=1}^{t} 1[h'_i(x) \neq y] \geq 1/3\right] \leq 6\varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D).$$

Since $h'(x)$ is a majority vote of $\{h'_1(x), \ldots, h'_t(x)\}$, the above implies that

$$\Pr[h'(x) \neq y] \leq 6\varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D).$$

Since S' is an arbitrary sample in $B_\eta(S)$, the above implies that SPV(Lrn) has the stated η-adversarial risk. □

To prove Theorem 3.1, we will need an optimal learner as an input learner for SPV.

Theorem A.2 (Haussler, Littlestone, and Warmuth [1994])**.** *Let $\mathcal{H}$ be a concept class with VC-dimension d, and let D be a distribution realizable by $\mathcal{H}$. Let also $n \in \mathbb{N}$, and let* Lrn *be the One-inclusion graph algorithm. Then $\varepsilon_n(\mathsf{Lrn}|D) \leq \frac{d}{n+1}$.*

Theorem 3.1 can now be immediately inferred as a direct application of Lemma A.1 and Theorem A.2.

Corollary A.3 (Realizable case – positive result)**.** *Let $\mathcal{H}$ be a concept class with VC-dimension d, let D be a distribution realizable by $\mathcal{H}$, and let* Lrn *be the One-inclusion graph algorithm. Let also $\eta \in (0,1)$ be the stability parameter given to* SPV *and let $n \geq 1/\eta$ be the sample size. Then* $\mathsf{SPV}(\mathsf{Lrn})$ *has η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{SPV}(\mathsf{Lrn})|D,\eta) \leq 42\eta d.$$

Proof. By Theorem A.2, plug in $\varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D) \leq \frac{d}{\lceil 1/(7\eta)\rceil+1} \leq 7\eta d$ to Lemma A.1 and the result follows. □

B Proof of Theorem 3.3 (Realizable Case – Impossibility Result)

Randomized Learning Rules. The impossibility result in Theorem 3.3 extends to randomized learning rules. But in order for the statement in Theorem 3.3 to be meaningful, we need to define adversarial risk with respect to randomized learners. As common in the literature on learning theory (see, e.g. the book of Shalev-Shwartz and Ben-David [2014]) we model randomized learners as deterministic learning rules with *continuous* predictions $p \in [0,1]$, and loss function $\ell(p,y) = |p-y|$. Indeed, the loss of a deterministic learner predicting a value $p \in [0,1]$ under the loss function $|y-p|$ is equal to the expected $0/1$-loss of a randomized learner predicting 1 with probability p. In the course of discussing the impossibility result, a *learning algorithm* $\mathsf{Lrn}\colon (\mathcal{X} \times \{0,1\})^* \to [0,1]^{\mathcal{X}}$ is a deterministic mapping which takes an input sample $S \in (\mathcal{X} \times \{0,1\})^*$ and maps it to a hypothesis $f \in [0,1]^{\mathcal{X}}$. We re-define η-adversarial risk with this view of randomized learners as *randomized η-adversarial risk*.

Definition B.1 (Randomized η-Adversarial Risk)**.** Let $\eta \in (0,1)$ be the adversaries' budget, let Lrn be a learning rule, and let D be a distribution over examples. The *randomized η-adversarial risk* of Lrn w.r.t D and sample size n is defined by

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) := \mathbb{E}_{S\sim D^n,(x,y)\sim D}\left[\sup_{S'\in B_\eta(S)} |\mathsf{Lrn}(S')(x) - y|\right].$$

The above definition of adversarial risk captures the case of an adversary that knows the expected prediction of the learner (that is, its test-time randomness), but not the learner's "internal" randomness (computation-time randomness). Indeed, the supremum is taken only with respect to the expected prediction, and not with respect to a specific execution of the algorithm determined by its internal randomness. Note that deterministic learners are a special case ($\{0,1\}$-valued outputs), in which case this definition collapses to the previous Definition 2.2. To avoid further notation, note that we overloaded the notation $\varepsilon_n^{\mathsf{Adv}}$ from Definition 2.2 in the above more general definition.

We are now ready to prove the impossibility result.

Theorem (Restatement of Theorem 3.3)**.** *There exists a constant $c_2 > 0$ so that the following holds. Let $\mathcal{H}$ be a non-trivial hypothesis class with VC dimension d and let $\eta \in (0,1)$. Then, there exists a distribution D realizable by $\mathcal{H}$, so that every learner* Lrn *has η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \geq \min\{c_2\eta d, 1/100\}$$

for any sample size $n \geq 1/\eta$.

Proof. Let $\mathcal{H}$ be a non-trivial concept class; in particular this means that its VC-dimension d satisfies $d \geq 1$. Let $\eta \in (0,1)$ be the adversaries' budget and let Lrn be an arbitrary learner. We need to show that there exists a distribution D realizable by $\mathcal{H}$ so that $\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \geq \min\{\eta d/32, 1/100\}$.

It suffices to consider the case when $\eta d/32 \leq 1/100$ and prove that $\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \geq \eta d/32$. Indeed, in the complementing case we have $\eta d/32 > 1/100$ and we need to show that $\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \geq 1/100$. Notice that $\eta d/32 > 1/100$ is equivalent to $\eta > \frac{32}{100d}$, and thus it suffices to show that even if the adversary's budget η is reduced to $\eta = \frac{32}{100d}$ then $\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \geq 1/100$. The latter indeed follows from the case when $\eta d/32 \leq 1/100$, because $\frac{32}{100d} \cdot d/32 = 1/100$.

We thus assume that $\eta d/32 \leq 1/100$ and set out to prove that $\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \geq \eta d/32$. We first consider the case when the VC-dimension of $\mathcal{H}$ is $d \geq 2$ and later handle the case when $d = 1$.

The VC dimensions is $d \geq 2$. Let $V = \{v_1, \ldots, v_d\} \subset \mathcal{X}$ be shattered by $\mathcal{H}$. Define a distribution $D_{\mathcal{X}}$ over V as follows. Set $D_{\mathcal{X}}(v_i) = \eta/2$ for all $2 \leq i \leq d$, and set $D_{\mathcal{X}}(v_1) = 1 - \eta(d-1)/2$. Notice that $D_{\mathcal{X}}$ is well defined since $d \geq 2$ and $\eta \leq 2/(d-1)$ (the latter is implied by the assumption that $\eta d/32 \leq 1/100$). For any labeling function $\ell \in \mathcal{Y}^V$, let D_ℓ denote the distribution over examples defined by $D_\ell(v_i, \ell(v_i)) = D_{\mathcal{X}}(v_i)$ for all $i \in [d]$. Note that D_ℓ is realizable, since V is shattered. It suffices to show that if the label vector $\ell \sim \mathcal{Y}^V$ is drawn uniformly at random then

$$\mathbb{E}_{\ell \sim \mathcal{Y}^V} \mathbb{E}_{S \sim D_\ell^n, (x,y) \sim D_\ell} \left[\sup_{S' \in B_\eta(S)} |\mathsf{Lrn}(S')(x) - y| \right] \geq \eta(d-1)/16. \tag{1}$$

Indeed, the above implies that there exists $\ell \in \mathcal{Y}^V$ such that

$$\begin{aligned} \mathbb{E}_{S \sim D_\ell^n, (x,y) \sim D_\ell} \left[\sup_{S' \in B_\eta(S)} |\mathsf{Lrn}(S')(x) - y| \right] &\geq \eta(d-1)/16 \\ &\geq \eta d/32. \qquad (d \geq 2) \end{aligned}$$

We establish Equation 1 in two steps:

1. For a sample S let S^u be the unlabeled input sample underlying it. We say that an unlabeled sample S^u and an instance x are *hard* if $x \neq v_1$ and x appears at most ηn times in S^u. In the first step we show that $\Pr_{\ell, S, (x,y)}[S^u, x \text{ are hard}] \geq \eta(d-1)/4$.
2. Let E_2 denote the event of all label vectors ℓ, input samples S, and test examples (x, y) such that $\sup_{S' \in B_\eta(S)} |\mathsf{Lrn}(S')(x) - y| \geq 1/2$. In the second step we show that $\Pr[E_2 | S^u, x \text{ are hard}] \geq 1/2$.

Indeed, once we prove both steps we have:

$$\begin{aligned} \mathbb{E}_{l \sim \mathcal{Y}^V} \mathbb{E}_{S \sim D_\ell^n, (x,y) \sim D_\ell} \left[\sup_{S' \in B_\eta(S)} |\mathsf{Lrn}(S')(x) - y| \right] &\geq \frac{1}{2} \cdot \Pr[E_2] \\ &\geq \frac{1}{2} \cdot \Pr[S^u, x \text{ are hard}] \cdot \Pr[E_2 | S^u, x \text{ are hard}] \\ &\geq \frac{1}{2} \cdot \frac{\eta(d-1)}{4} \cdot \frac{1}{2} = \eta(d-1)/16, \end{aligned}$$

as desired.

Let us prove step 1. Notice that S^u and x are distributed according to the marginal distribution $D_{\mathcal{X}}^{n+1}$. Thus, $x \neq v_1$ with probability $\eta(d-1)/2$, and given that $x \neq v_1$ the expected number of appearances of x in S^u is $\eta n/2$. Therefore, by Markov's inequality, the probability that S^u and x are hard given that $x \neq v_1$ is at least $\frac{\eta n/2}{\eta n} = 1/2$. Thus, the overall probability that S^u, x are hard is at least $\eta(d-1)/4$.

We now prove step 2. Let S^u, x be hard. It suffices to show that

$$\mathbb{E}_{\ell(x_1), \ldots, \ell(x_n), y} \left[\sup_{S' \in B_\eta(S)} |\mathsf{Lrn}(S')(x) - y| \;\middle|\; S^u, x \right] \geq \frac{1}{2},$$

where $\ell(x_i)$ is the label of the i'th instance in S^u and y is the test label. Crucially, notice that the test-label y is independent of S^u, x, and all other labels $\ell(x_i)$ for $x_i \in S^u$ such that $x_i \neq x$. Thus, even conditioned on S^u, x and all labels of $x_i \neq x$, the test-label y is distributed uniformly in $\mathcal{Y} = \{0, 1\}$.

Define samples S_0', S_1' to be the same as S' with the exception that every appearance of x in S_0' is labeled with 0 in S_0' and with 1 in S_1'. Note that both $S_0', S_1' \in B_\eta(S)$, because S^u, x are hard. We claim that, with probability at least half over the drawing of the $\ell(x_i)$'s and y we have

$$|\mathsf{Lrn}(S_0')(x) - \ell(y)| \geq 1/2 \quad \text{or} \quad |\mathsf{Lrn}(S_1')(x) - \ell(y)| \geq 1/2.$$

Having this in hand, and given that $\hat{S}$ is hard, we are done: both $S'_0, S'_1 \in B_\eta(S)$, and Item 2 follows.

It thus remains to show that indeed $|\mathsf{Lrn}(S'_0)(x) - y| \geq 1/2$ or $|\mathsf{Lrn}(S'_1)(x) - \ell(y)| \geq 1/2$ with probability at least $1/2$ over the drawing of the $\ell(x_i)$'s and y. This is achieved by a simple case analysis:

- if both $\mathsf{Lrn}(S'_0)(x), \mathsf{Lrn}(S'_1)(x) \leq 1/2$ then with probability $1/2$ we have $y = 1$ and the claim follows. The case $\mathsf{Lrn}(S'_0)(x), \mathsf{Lrn}(S'_1)(x) > 1/2$ is treated similarly.
- If $\mathsf{Lrn}(S'_0)(x) \leq 1/2, \mathsf{Lrn}(S'_1)(x) \geq 1/2$ then $|\mathsf{Lrn}(S'_0)(x) - y| \geq 1/2$ or $|\mathsf{Lrn}(S'_1)(x) - y| \geq 1/2$ with probability 1 and the claim follows. The case $\mathsf{Lrn}(S'_0)(x) > 1/2, \mathsf{Lrn}(S'_1)(x) < 1/2$ is treated similarly.

This finishes the proof of Theorem 3.3 when the VC-dimension d is at least 2.

The VC-dimension is $d = 1$. In this case, we can not define the distribution $D_{\mathcal{X}}$ as before because $d < 2$. However, the fact that $\mathcal{H}$ is non-trivial allows to modify the definition as follows. Let $x_1, x_2 \in \mathcal{X}$ and $h_1, h_2 \in \mathcal{H}$ so that $h_1(x_1) = h_2(x_1)$ and $h_1(x_2) \neq h_2(x_2)$, guaranteed by the fact that $\mathcal{H}$ is non-trivial. Set $V = \{x_1, x_2\}$, and define the distribution $D_{\mathcal{X}}$ by $D_{\mathcal{X}}(x_1) = 1-\eta/2, D_{\mathcal{X}}(x_2) = \eta/2$ as in the case $d \geq 2$. Also, define the random labeling function ℓ to agree with h_1 on with probability half and with h_2 with probability half. The rest of the proof is the same. □

C Proof of Theorem 3.6 (Realizable and Proper Case – Positive Result)

Theorem (Restatement of Theorem 3.6)**.** *There exists a constant $c > 0$ so that the following holds. Let $\mathcal{H}$ be the class of halfspaces over $\mathbb{R}^d$ for some $d \geq 1$, and let $\eta \in (0, 1)$. Then, there exists a proper learner* Lrn *having η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D, \eta) \leq c\eta d^3$$

for any distribution D realizable by $\mathcal{H}$ and for any sample size $n \geq 1/\eta$.

To derive Theorem 3.6, we reinforce the SPV algorithm with a technique introduced by Kane, Livni, Moran, and Yehudayoff [2019] and further developed by Bousquet, Hanneke, Moran, and Zhivotovskiy [2020]. This technique allows in certain cases to *project* a majority vote of hypotheses from the class $\mathcal{H}$ back to $\mathcal{H}$. Its applicability hinges on a combinatorial parameter called the *projection number*:

Definition C.1 (Projection Number)**.** Let $\mathcal{H}$ be a concept class. For any $\ell \geq 2$ and for any multiset $\mathcal{H}' \subset \mathcal{H}$ define the set $\mathcal{X}_{\mathcal{H}',\ell}$ to be the set of all $x \in \mathcal{X}$, for which the number of hypotheses in $\mathcal{H}'$ that disagree with $\mathsf{Maj}(\mathcal{H}')(x)$ is less than $|\mathcal{H}'|/\ell$. The Projection Number of the class $\mathcal{H}$, denoted $k_p = k_p(\mathcal{H})$, is defined to be the smallest ℓ so that for any finite multiset $\mathcal{H}' \subset \mathcal{H}$, there exist $h \in \mathcal{H}$ such that $h(x) = \mathsf{Maj}(\mathcal{H}')(x)$ for all $x \in \mathcal{X}_{\mathcal{H}',\ell}$. If no such ℓ exists then $k_p = \infty$.

First, let us analyze the general performance of PSPV.

Lemma C.2 (General performance of PSPV)**.** *Let $\mathcal{H}$ be a concept class with a finite projection number $k_p < \infty$. Let D be a distribution over examples, and let Lrn_p be a proper learning rule. Let also $\eta \in (0, 1)$ be the stability parameter given to* PSPV *and let $n \geq 1/\eta$ be the sample size. Then* $\mathsf{PSPV}(\mathsf{Lrn}_p)$ *is a proper learning rule having η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{PSPV}(\mathsf{Lrn}_p)|D, \eta) \leq 4k_p \varepsilon_{\lceil 1/(5k_p\eta)\rceil}(\mathsf{Lrn}_p|D).$$

Proof. The proof follows the same lines as the proof of Lemma A.1. Let $S \sim D^n$ be the input sample, and $(x, y) \sim D$ be the test example. Note that for all $i \in [t]$ (where $t = \lfloor 5k_p\eta n\rfloor$ is the number of subsamples of size at least $\frac{1}{5k_p\eta}$ in the partition made by PSPV) it holds that $\mathbb{E}\big[1[h_i(x) \neq y]\big] \leq \varepsilon_{\lceil 1/(5k_p\eta)\rceil}(\mathsf{Lrn}_p|D)$. By applying linearity of expectation we get

$$\mathbb{E}\left[\frac{1}{t}\sum_{i=1}^{t} 1[h_i(x) \neq y]\right] \leq \varepsilon_{\lceil 1/(5k_p\eta)\rceil}(\mathsf{Lrn}_p|D).$$

By Markov's inequality:

$$\Pr\left[\frac{1}{t}\sum_{i=1}^{t} 1[h_i(x) \neq y] \geq \frac{1}{4k_p}\right] \leq 4k_p \varepsilon_{\lceil 1/(5k_p\eta)\rceil}(\mathsf{Lrn}_p|D).$$

Let $S' \in B_\eta(S)$. Let $h' = \mathsf{PSPV}(\mathsf{Lrn}_p)(S')$, and for all $i \in [t]$ let h'_i be the hypothesis obtained by training Lrn_p on $S'^{(i)}$. Note that, since S and S' are η-close by, and since $n \geq 1/\eta$ it holds that

$$\frac{1}{t}\sum_{i=1}^{t} 1\left[S^{(i)} \neq S'^{(i)}\right] \leq \frac{\eta n}{\lfloor 5k_p\eta n\rfloor} \leq \frac{1}{4k_p}.$$

Hence it is implied that $\frac{1}{t}\sum_{i=1}^{t} 1\left[h_i(x) \neq h'_i(x)\right] \leq \frac{1}{4k_p}$. Thus, the event that $\frac{1}{t}\sum_{i=1}^{t} 1[h'_i(x) \neq y] \geq \frac{1}{2k_p}$ implies (or, is contained in) the event that $\sum_{i=1}^{t} 1[h_i(x) \neq y] \geq \frac{1}{4k_p}$, hence:

$$\Pr\left[\frac{1}{t}\sum_{i=1}^{t} 1[h'_i(x) \neq y] \geq \frac{1}{2k_p}\right] \leq 4k_p \varepsilon_{\lceil 1/(5k_p\eta)\rceil}(\mathsf{Lrn}_p|D).$$

Note that by definition of projection number it holds that the hypothesis $h' \in \mathcal{H}$ returned by the algorithm exists. Hence, by definition of $\mathcal{X}_{\{h'_1,\ldots,h'_t\},2k_p}$ the above implies that

$$\Pr[h'(x) \neq y] \leq 4k_p \varepsilon_{\lceil 1/(5k_p\eta)\rceil}(\mathsf{Lrn}_p|D).$$

Since S' is an arbitrary sample in $B_\eta(S)$, the above implies that $\mathsf{PSPV}(\mathsf{Lrn}_p)$ has the stated η-adversarial risk. □

To prove Theorem 3.6 we will use the following result regarding the projection number of halfspaces.

Theorem C.3 (Kane, Livni, Moran, and Yehudayoff [2019], Braverman, Kol, Moran, and Saxena [2019], Bousquet, Hanneke, Moran, and Zhivotovskiy [2020])**.** *Let $\mathcal{H}$ be the class of halfspaces over $\mathbb{R}^m$. Then $k_p(\mathcal{H}) = d(\mathcal{H}) = m+1$.*

We will use the SVM learner as an input learner for PSPV.

Theorem C.4 (Vapnik and Chervonenkis [1974])**.** *Let $m \geq 1$ and let $\mathcal{H}$ be the class of halfspaces over $\mathbb{R}^m$. Let D be a distribution realizable by $\mathcal{H}$. Let also $n \in \mathbb{N}$, and let Lrn_p be the SVM algorithm. Then $\varepsilon_n(\mathsf{Lrn}_p|D) \leq \frac{m+1}{n+1}$.*

Theorem 3.6 now follows as an immediate application of Theorem C.3, Theorem C.4 and Lemma C.2.

Corollary C.5 (Realizable and proper case – positive result)**.** *Let $m \geq 1$, let $\mathcal{H}$ be the class of halfspaces over $\mathbb{R}^m$, and let $d = m+1$ be the VC-dimension of $\mathcal{H}$. Let D be a distribution realizable by $\mathcal{H}$, and let Lrn_p be the SVM learner. Let also $\eta \in (0,1)$ be the stability parameter given to PSPV and let $n \geq 1/\eta$ be the sample size. Then $\mathsf{PSPV}(\mathsf{Lrn}_p)$ has η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{PSPV}(\mathsf{Lrn}_p)|D,\eta) \leq 20\eta d^3.$$

Proof. By Theorem C.3, if $\mathcal{H}$ is the class of halfspaces over $\mathbb{R}^m$ then its projection number is $k_p = d = m+1$. Also, by Theorem C.4, we have that $\varepsilon_{\lceil 1/(5d\eta)\rceil}(\mathsf{Lrn}_p|D) \leq 5\eta d^2$. Plug both results to Lemma C.2, and the result follows. □

D Proof of Theorem 4.1 (Agnostic Case – Positive Result)

Theorem (Restatement of Theorem 4.1)**.** *There exist constants c_1, c_2 so that the following holds. Let $\mathcal{H}$ be a hypothesis class with VC dimension d and let $\eta \in (0,1)$. Then, there exists a learner Lrn having η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \leq c_2 \cdot \mathsf{OPT} + c_1 \cdot d \cdot \eta$$

for any distribution D over examples and for any sample size $n \geq 1/\eta$.

To derive Theorem 4.1, we use an agnostic variation of the One-inclusion graph learner.

Theorem D.1 (Corollary of Lemma 16 in [Long, 1999])**.** *There exists a constant C such that the following holds. Let $\mathcal{H}$ be a concept class with VC-dimension d and let* Lrn *be the agnostic variation of the One-inclusion graph algorithm implied by Lemma 16 in [Long, 1999]. Let also n be the sample size. Then, for any distribution D over examples (not necessarily such that is realizable by $\mathcal{H}$), it holds that $\varepsilon_n(\mathsf{Lrn}|D) \leq C(\mathsf{OPT} + d/n)$.*

Theorem 4.1 is implied by the following immediate corollary of Theorem 4.1 and Lemma A.1.

Corollary D.2 (Agnostic case – positive result)**.** *There exists a constant C such that the following holds. Let $\mathcal{H}$ be a concept class with VC dimension d, let $\eta \in (0,1)$ be the stability parameter given to* SPV*, and let D be a (not necessarily realizable) distribution over examples. Let also $n \geq 1/\eta$ be the sample size. Then* $\mathsf{SPV}(\mathsf{Lrn})$ *has η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{SPV}(\mathsf{Lrn})|D,\eta) \leq 6C\mathsf{OPT} + 42C\eta d,$$

where Lrn *is the agnostic variant of the One-inclusion graph algorithm mentioned in Theorem D.1.*

Proof. By Theorem D.1, there exists a constant C such that $\varepsilon_{\lceil 1/(7\eta)\rceil}(\mathsf{Lrn}|D) \leq C\mathsf{OPT} + 7C\eta d$. Plug this into Lemma A.1 and the result follows. □

E Proof of Theorem 4.2 (Agnostic Case – Impossibility Result)

Theorem (Restatement of Theorem 4.2)**.** *Let $\eta' \in (0,1), n \in \mathbb{N}$. For any hypothesis class $\mathcal{H}$ that has at least two hypotheses and. for any deterministic learner, there is a distribution D over (two) examples and $\eta = \eta' + \widetilde{O}(1/\sqrt{n})$ such that* Lrn *has η-adversarial risk*

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D,\eta) \geq 2\mathsf{OPT} + \Omega(\eta') - O(1/n).$$

Let $h_1, h_2 \in \mathcal{H}$ be two distinct hypotheses and let $x \in \mathcal{X}$ such that $h_1(x) \neq h_2(x)$. In this proof we consider distributions D supported only on $\{(x,0),(x,1)\}$. Notice that such a distribution is determined by the probability $p = \Pr_{(x,y)\sim D}[y=1]$ and hence can be thought of as a coin with bias p. Thus, the task of agnostic learning such distributions with respect to instance-targeted data poisoning boils down to predicting a random p-coin toss given an input sample of n p-coin tosses out of which at most $\eta \cdot n$ tosses are flipped by an adversary who *knows* the result of the coin toss that needs to be predicted. We summarize this in the following game:

Definition E.1 (The coin game)**.** The coin game is parameterized by (n,η) where $n \in \mathbb{N}, \eta \in (0,1)$, and the game is played between an adversary Adv and a learner Lrn as follows.

1. Adv picks $p \in [0,1]$.
2. $c_1,\ldots,c_{n+1} \sim X_p^{n+1}$, where X_p is a binary random variable satisfying $\Pr[X_p = 1] = p$.
3. Adv changes $\overline{c} = (c_1,\ldots,c_n)$ into $\overline{c}' = (c_1',\ldots,c_n')$ where $\mathsf{d_H}(\overline{c},\overline{c}') \leq \eta \cdot n$.
4. Lrn gets to see $\overline{c}' = (c_1',\ldots,c_n')$ and outputs a bit $c \in \{0,1\}$.
5. Lrn wins if $c = c_{n+1}$, and Adv wins otherwise.

In this game, we define $\mathsf{OPT}_p = \min\{p, 1-p\}$ to be the optimal error of the learner if it had known p, and we define $\mathsf{ERR} = \Pr[c \neq c_{n+1}]$ (over all the randomness involved) to be the *error* of the game (i.e., when the learner does not win). We also refer to $\mathsf{ERR} - \mathsf{OPT}_p$ as the regret.[4]

Theorem E.2. *For any $\eta' \in [0,1/2]$ and any deterministic learner* Lrn *that participates in the coin game of Definition E.1, there is an adversary* Adv *with a fixed choice of p (determining $\mathsf{OPT} = \mathsf{OPT}_p$) and $\eta = \eta' + \widetilde{O}(1/\sqrt{n})$ such that when we run the game of Definition E.1 with parameters (n,η), it holds that $\mathsf{ERR} - \mathsf{OPT} \geq 1/2 + \eta' - O(1/n)$.*

[4]Note that OPT_p is a random variable in general, if the adversary is randomized. But if the adversary uses a deterministic strategy for the fixed p, then OPT_p is a constant.

Remark 1 (On deterministic adversaries)**.** *In Theorem E.2 we show the existence of an adversary with a fixed choice of p. This adversary is in fact randomized. Here we remark that, for every fixed (even randomized) learner* Lrn *and a fixed choice of p, there is always a deterministic adversary that achieves the maximum regret (for such* Lrn, p*). The reason is that if by using randomness r_{Adv} the adversary achieves expected regret $R(r_{\mathsf{Adv}})$ over the randomness of the learner, then its overall regret will be $\mathbb{E}_{r_{\mathsf{Adv}}}[R(r_{\mathsf{Adv}})]$. Therefore, if $r^{(p)}_{\mathsf{Adv}}$ is the randomness (for fixed p) that maximizes $R(r_{\mathsf{Adv}})$, the adversary can simply fix its randomness to $r^{(p)}_{\mathsf{Adv}}$ without decreasing its gain. This means that without loss of generality, the adversary of Theorem E.2 is deterministic. In addition, since the adversary sends the first message p, the overall optimal strategy* Adv *(who picks p potentially in a randomized way) can* also *fix p to what maximizes $R(r^{(p)}_{\mathsf{Adv}})$, which makes* Adv *fully deterministic.*

Deriving Theorem 4.2. We first show how to derive Theorem 4.2 from Theorem E.2.

Proof of Theorem 4.2. First assume $\eta' \leq 1/2$, and at the end we explain how to deal with $\eta' > 1/2$. By Theorem E.2, there is an adversary (with a fixed choice of p) in the coin game of Definition E.1 such that $\mathsf{ERR} - \mathsf{OPT} \geq 1/2 + \eta' - O(1/n)$ when we use (n, η) as game parameters. Since $\mathsf{ERR} \leq 1$, we have $\mathsf{OPT} \leq 1/2 - \eta' + O(1/n)$, and so

$$\mathsf{ERR} - \mathsf{OPT} \geq 1/2 + \eta' - O(1/n) \geq \mathsf{OPT} + 2\eta' - O(1/n).$$

This implies that $\mathsf{ERR} \geq 2\mathsf{OPT} + 2\eta' - O(1/n)$. Note that OPT is indeed the minimal error that the learner can achieve by outputting any of the constant coins $0, 1$, which in turn refers to outputting either of h_0, h_1 from the hypothesis class. In addition, ERR is equal to the adversarial risk for parameters n, η and the distribution D_p for this particular attack. This means that

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D, \eta) \geq 2\mathsf{OPT} + 2\eta' - O(1/n),$$

which implies Theorem 4.2. Now, if $\eta' > 1/2$, we first artificially decrease adversary's budget η' to $\eta'' = 1/2$, which leads to

$$\varepsilon_n^{\mathsf{Adv}}(\mathsf{Lrn}|D, \eta) \geq 2\mathsf{OPT} + 2\eta'' - O(1/n),$$

but we also know that $\eta'' = \Omega(\eta')$, which again proves Theorem 4.2. □

Before proving Theorem E.2 we recall two useful tools.

Lemma E.3 (Proposition 2.1.1 in Talagrand [1995])**.** *Let $\mu = \mu_1 \times \ldots \mu_n$ be a product measure and $f \colon \mu \mapsto \{0, 1\}$ a boolean function where $\Pr[f(\mu) = 1] = 1/2$. Then, for all $b \in [n]$,*

$$\Pr_{x \sim \mu}[\exists x', \mathsf{d_H}(x, x') \leq b \wedge f(x') = 1] \geq 1 - 2e^{-b^2/n}.$$

In other words, with probability at least $1 - 2e^{-b^2/n}$ over the sampling of $x \sim \mu$, one can change up to b of the coordinates of x and obtain x' (i.e., $\mathsf{d_H}(x, x') \leq b$) such that $f(x') = 1$.

Lemma E.4 (Modifying coins)**.** *Suppose $0 \leq p, p' \leq 1$, and let $q = |p - p'|$. Then there is an adversary who can change $q \cdot n$ coins, in expectation, of a sample $\overline{c} \sim X_p^n$ into $\overline{c}'$ (i.e., $\mathbb{E}[\mathsf{d_H}(\overline{c}', \overline{c})] = q \cdot n$) such that $\overline{c}' \sim X_{p'}^n$ (Namely, the tampered sequence looks exactly like it is sampled from $X_{p'}^n$, while in reality it is being first sampled from X_p^n and then modified by the adversary in $q \cdot n$ points in expectation). Moreover, the probability that the adversary changes more than $qn + \sqrt{(n \ln n)/2}$ of the coordinates is at most $1/n$.*

Proof. Without loss of generality, let $p' - p = q \geq 0$. Then the adversary will change each of the coins with independent probability q as follows. If a coin $c_i = 1$, the adversary will not change it, which will happen with probability p. If $c_i = 0$, which will happen with probability $1 - p$, the adversary will change this to 1 with probability $q/(1 - p)$ over its own randomness. Note that $q = p' - p \leq (1 - p)$, and so $q/(1 - p) \in [0, 1]$ can be interpreted as a probability. The probability that $c_i' = 1$ is now exactly $p + q = p'$, while the expected number of changed coins is $q \cdot n$. Finally, since the adversary's changes of the coin outcomes are done *independently* for each coin, the bound on the number of changes made by the adversary is implied by the Hoeffding-Chernoff bound. □

We now prove Theorem E.2 using the two tools above.

Proof of Theorem E.2. Fix the deterministic learning algorithm Lrn. This means that for every given input vector $\overline{c} = (c_1, \ldots, c_n)$, we have $\mathsf{Lrn}(\overline{c}) \in \{0, 1\}$. Now define $\alpha(p) = \Pr_{\overline{c} \sim X_p^n}[\mathsf{Lrn}(\overline{c}) = 1]$.

We do a case study as follows.

- If $\alpha(0) \neq 0$, it means that $\alpha(0) = 1$ (i.e., the deterministic learner outputs 1 over the all zero vector). In this case, $\mathsf{OPT} = 0$ and $\mathsf{ERR} = 1$, which implies $\mathsf{ERR} - \mathsf{OPT} \geq 1/2 + \eta'$.
- If $\alpha(1) \neq 1$, it implies $\mathsf{ERR} - \mathsf{OPT} \geq 1/2 + \eta'$ similarly.
- If none of the above cases happens, we can assume $\alpha(b) = b$ for both $b \in \{0, 1\}$. Because the learner is deterministic, $\mathsf{Lrn}(\overline{c}) = 1$ if $\overline{c} \in \mathcal{S}$ for a fixed set $\mathcal{S} \subseteq \{0, 1\}^n$. Moreover, for all $\overline{c} \in \{0, 1\}^n$, it holds that $\Pr[X_p^n = \overline{c}] = p^d(1-p)^{n-d}$, where d is the number of non-zero coordinates of $\overline{c}$. This implies that $\alpha(p)$ is a polynomial of degree at most n over p, which is a continuous function. Therefore, there exists $q \in (0, 1)$ such that $\alpha(q) = 1/2$. Without loss of generality, assume that $q \leq 1/2$. Then, the adversary picks $p = \max\{0, q - \eta'\}$, which guarantees $\mathsf{OPT} = p \leq 1/2 - \eta'$ (due to the assumptions $\eta', q \leq 1/2$). Then, the adversary uses Lemma E.4 to shift the coin's distribution back to q. For this change, the adversary makes at most $\eta' \cdot n + \sqrt{(n \ln n)/2}$ changes with probability $1 - 1/n$. We then apply the algorithm of Lemma E.3 to make further $\sqrt{n \ln(2n)}$ changes to the coins to make sure that the output of the learner is the wrong outcome (different from c_{n+1}) with probability $1 - 1/n$. In total, the adversary can make at most $\eta' \cdot n + \sqrt{(n \ln n)/2} + \sqrt{n \ln(2n)} \in \eta' \cdot n + \widetilde{O}(\sqrt{n})$ changes to the coin flips outcomes, while the learner's output bit is wrong with probability $1 - 1/n - 1/n = 1 - O(1/n)$. Since $\mathsf{OPT} \leq 1/2 - \eta'$ and $\mathsf{ERR} \geq 1 - O(1/n)$, we get

$$\mathsf{ERR} - \mathsf{OPT} \geq 1/2 + \eta' - O(1/n),$$

  which finishes the proof. □