# Learning and Certification under Instance-targeted Poisoning

**Ji Gao**[1] **Amin Karbasi**[2] **Mohammad Mahmoody**[3]

[1]University of Virginia
[2]Yale University
[3]University of Virginia

## Abstract

In this paper, we study PAC learnability and certification under instance-targeted poisoning attacks, where the adversary may change a fraction of the training set with the goal of fooling the learner at a specific target instance. Our first contribution is to formalize the problem in various settings, and explicitly discussing subtle aspects such as learner's randomness and whether (or not) adversary's attack can depend on it. We show that when the budget of the adversary scales sublinearly with the sample complexity, PAC learnability and certification are achievable. In contrast, when the adversary's budget grows linearly with the sample complexity, the adversary can potentially drive up the expected 0-1 loss to one. We also study *distribution-specific* PAC learning in the same attack model and show that *proper* learning with certification is possible for learning half spaces under natural distributions. Finally, we empirically study the robustness of $K$ nearest neighbour, logistic regression, multi-layer perceptron, and convolutional neural network on real data sets against targeted-poisoning attacks. Our experimental results show that many models, especially state-of-the-art neural networks, are indeed vulnerable to these strong attacks. Interestingly, we observe that methods with high standard accuracy might be more vulnerable to instance-targeted poisoning attacks.

## 1 INTRODUCTION

Learning to predict from empirical examples is a fundamental problem in machine learning. In its classic form, the problem involves a benign setting where the empirical and test examples are sampled from the same distribution $D$. More formally, a learner, denoted by $\mathsf{Lrn}$, is given a training set $\mathcal{S}$, consists of i.i.d. samples $(x, y)$ from distribution $D$, where $x$ is a data point and $y$ is its label. Then, the learner returns a model/hypothesis $h$ where it will be ultimately tested on a fresh sample from the same distribution $D$.

More recently, the above-mentioned classic setting has been revisited by allowing adversarial manipulations that tamper with the process, while still aiming to make correct predictions. In general, adversarial tampering can take place in both training or testing of models. Our interest in this work is on a form of training-time attacks, known as poisoning or causative attacks [Barreno et al., 2006, Papernot et al., 2016, Diakonikolas and Kane, 2019, Goldblum et al., 2020]. In particular, poisoning adversaries may partially change the training set $\mathcal{S}$ into another training set $\mathcal{S}'$ in such a way that the "quality" of the returned hypothesis $h'$ by the learning algorithm $\mathsf{Lrn}$, that is trained on $\mathcal{S}'$ instead of $\mathcal{S}$, degrades significantly. Depending on the context, the way we measure the quality of the poisoning attack may change. For instance, the quality of $h'$ may refer to the expected error of $h'$ when test data points are sampled from the distribution $D$. It could also refer to the error on a particular test point $x$, known to the adversary but unknown to the learning algorithm $\mathsf{Lrn}$. The latter scenario, which is the main focus of this work, is known as (instance) *targeted poisoning* [Barreno et al., 2006]. In this setting, as the name suggests, an adversary could craft its strategy based on the knowledge of a target instance $x$. Given a training set of $\mathcal{S}$ of size $m$, we assume that an adversary can change up to $b(m)$ data points, and we refer to $b(m)$ as adversary's "budget". Other examples of natural (weaker) attacks may include flipping binary labels, or adding/removing data points (see Section 2).

Given a poisoning attack, the predictions of a learning algorithm may or may not change. To this end, Steinhardt et al. [2017] initiated the study of *certification* against poisoning attacks, studying the conditions under which a learning algorithm can certifiably obtain an expected low risk. To extend these results to the instance-targeted positing scenario, Rosenfeld et al. [2020] recently addressed the *instance targeted* (a.k.a., pointwise) certification with the goal of provid-

*Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence* (UAI 2021), PMLR 161:2135–2145.

ing certification guarantees about the prediction of *specific* instances when the adversary can poison the training data. While the instance-targeted certification has sparked a new line of research [Levine and Feizi, 2021, Chen et al., 2020, Weber et al., 2020, Jia et al., 2020] with interesting insights, the existing works do not address the fundamental question of when, and under what conditions, learnability and certification are achievable under the instance-targeted poisoning attack. In this work, we take an initial step along this line and layout the precise conditions for such guarantees.

**Problem setup.** Let $\mathcal{H}$ consists of a hypothesis class of classifiers $h : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ denotes the instances domain and $\mathcal{Y}$ the labels domain. We would like to study the learnability of $\mathcal{H}$ under instance-targeted poisoning attacks. But before discussing the problem in that setting, we recall the notion of PAC learning *without* attacks.

Informally speaking, $\mathcal{H}$ is "Probably Approximately Correct" learnable (PAC learnable for short) if there is a learning algorithm $\mathsf{Lrn}$ such that for every distribution $D$ over $\mathcal{X} \times \mathcal{Y}$, if $D$ can be learned with $\mathcal{H}$ (i.e., the so-called realizability assumption holds) then with high probability over sampling any sufficiently large set $\mathcal{S} \sim D^m$, $\mathsf{Lrn}$ maps $\mathcal{S}$ to a hypothesis $h \in \mathcal{H}$ with "arbitrarily small" risk under the distribution $D$. $\mathsf{Lrn}$ is called *improper* if it is allowed to output functions outside $\mathcal{H}$, and it is a *distribution-specific* learner, if it is only required to work when the marginal distribution $D_{\mathcal{X}}$ on the instance domain $\mathcal{X}$ is fixed e.g., to be isotropic Gaussian. (See Section 2 and Definition 2.4 for formal definitions.)

Suppose that before the example $(x, y) \sim D$ is tested, an adversary who is aware of $(x, y)$ (and hence, is *targeting* the instance $x$) can craft a poisoned set $\mathcal{S}'$ from $\mathcal{S}$ by *arbitrarily changing* up to $b$ of the training examples in $\mathcal{S}$. Now, the learning algorithm encounters $\mathcal{S}'$ as the training set and the hypothesis it returns is, say, $h' \in \mathcal{H}$ in the proper learning setting. Now, the predicted label of $x$, i.e., $y' = h'(x)$, may no longer be equal to the correct label $y$.

**Main questions.** In this paper, we would like to study under what conditions on the class complexity $\mathcal{H}$, budget $b$, and different (weak/strong) forms of instance-targeted poisoning attacks, one can achieve (proper/improper) PAC learning. In particular, the learner's goal is to still be correct, with high probability, on *most* test instances, despite the existence of the attack. A stronger goal than robustness is to also *certify* the predictions $h(x) = y$ with a lower bound $k$ on how much an instance-targeted poisoning adversary needs to change the training set $\mathcal{S}$ to eventually flip the decision on $x$ into $y' \neq y$. In this work, we also keep an eye on when robust learners can be enhanced to provide such guarantees, leading to *certifiably robust* learners.

We should highlight that all the aforementioned methods [Rosenfeld et al., 2020, Levine and Feizi, 2021, Chen et al., 2020, Weber et al., 2020, Jia et al., 2020] mainly considered practical methods that allow predictions for individual instances under specific conditional assumptions about the model's performance at the decision time that can be only verified empirically, but it is not clear (provably) if such conditions would actually happen during the prediction moment. In this work, we avoid such assumptions and address the question of under what conditions on the *problem's setting*, the learnability is possible provably.

Our contributions are as follows.

*Formalism.* We provide a precise and general formalism for the notions of certification and PAC learnability under instance-targeted attacks. These formalisms are based on a careful treatment of the notions of *risk* and *robustness* defined particularly for learners under instance-targeted poisoning attacks. The definitions carefully consider various attack settings, e.g., based on whether the adversary's perturbation can depend on learner's randomness or not, and also distinguish between various forms of certification (to hold for *all* training sets, or just *most* training sets.)

*Distribution-independent setting.* We then study the problem of robust learning and certification under instance-targeted poisoning attacks in the distribution-independent setting. Here, the learner shall produce "good" models for *any* distribution over the examples, as long as the distribution can be learned by at least one hypothesis $h \in \mathcal{H}$ (i.e., the realizable setting). We separate our studies here based on the subtle distinction between two cases: Adversaries who can base their perturbation also for a *fixed* randomness of the learner (the default attack setting), and those whose perturbation would be retrained using *fresh* randomness (called weak adversaries). In the first setting, We show that as long as the hypothesis class $\mathcal{H}$ is (properly or improperly) PAC learnable under the 0-1 loss and the strong adversary's budget is $b = o(m)$, where $m$ is the number of samples in the training set, then the hypothesis class $\mathcal{H}$ is always *improperly* PAC learnable under the instance-targeted attack with certification (Theorem 3.3). This result is inspired by the recent work of Levine and Feizi [2021] and comes with certification. We then show that the limitation on $b(m) = o(m)$ is inherent in general, as when $\mathcal{H}$ is the set of homogeneous hyperplanes, if $b(m) = \Omega(m)$, then robust PAC learning against instance-targeted poisoning is impossible in a strong sense (Theorem 3.5). $m$. We then show that if the adversary is "weak" and is *not* aware of learner's randomness, if the hypothesis class $\mathcal{H}$ is properly PAC learnable and the weak adversary's budget is $b = o(m)$, then $\mathcal{H}$ is also properly PAC learnable under instance-targeted attacks (Theorem 3.2). This result, however, does *not* come with certification guarantees.

*Distribution-specific learning.* We then study robust learning under instance-targeted poisoning when the instance distribution is fixed. We show that when the projection of the marginal distribution $D_{\mathcal{X}}$ is the uniform distribution over the unit sphere (e.g., $d$-dimensional isotropic Gaussian), the hypothesis class consists of homogeneous half-spaces, and

the strong adversary's budget is $b = c/\sqrt{d}$, then proper PAC learnability under instant-targeted attack is possible iff $c = o(m)$ (see Theorems 3.7 and 3.8). Note that if we allow $d$ to grow with $m$ to capture the "high dimension" setting, then the mentioned result becomes incomparable to our above-mentioned results for the distribution-independent setting). To prove this result we use tools from measure concentration over the unit sphere in high dimension.

*Experiments.* We empirically study the robustness of $K$ nearest neighbour, logistic regression, multi-layer perceptron, and convolutional neural network on real data sets. We observe that methods with high standard accuracy (such as convolutional neural network) are indeed more vulnerable to instance-targeted poisoning attacks. This observation might be explained by the fact that more complex models fit the training data better and thus the adversary can more easily confuse them at a specific test instance. A possible interpretation is that models that somehow "memorize" their data could be more vulnerable to targeted poisoning. In addition, we study whether dropout on the inputs and also $L2$-regularization on the output can help the model to defend against instance-targeted poisoning attacks. We observe that adding these regularization to the learner does not help in defending against such attacks.

### 1.1 RELATED WORK

The concurrent work of Blum et al. [2021] also studies instance-targeted PAC learning. In particular, they formalize and prove positive and negative results about PAC learnability under instance-targeted poisoning attacks, in which the adversary can add an unbounded number of clean-label examples to the training set. In comparison, we formalize the problem for any prediction task, and we study both robust learning and certification. Our main positive and negative results are, however, proved for classification tasks and for adversaries who can *change* a limited number of examples in the training set. Other theoretical works have also studied instance-targeted poisoning *attacks* (rather than learnability under such attacks) using clean labels [Mahloujifar and Mahmoody, 2017, Mahloujifar et al., 2018, 2019b, Mahloujifar and Mahmoody, 2019, Mahloujifar et al., 2019a, Diochnos et al., 2019, Etesami et al., 2020]. The work of Shafahi et al. [2018] studied such (targeted clean-label) attacks empirically, and showed that neural nets can be very vulnerable to them. Finally, Koh and Liang [2017] also studied clean label attacks empirically, but for non-targeted setting.

More broadly, some classical works in machine learning can also be interpreted as (non-targeted) data poisoning [Valiant, 1985, Kearns and Li, 1993, Sloan, 1995, Bshouty et al., 2002]. In fact, the work of Bshouty et al. [2002] studies the same question as in this paper, but for the *non-targeted setting*. However, making learners robust against such attacks can easily lead to *intractable* learning methods that do *not* run in polynomial time. Recently, starting with the seminal results of Diakonikolas et al. [2016], Lai et al. [2016] (and many follow up works, e.g., Diakonikolas et al. [2019a,b], see [Diakonikolas and Kane, 2019]), it was shown that in some natural settings one can go beyond the intractability barriers and obtain polynomial-time methods to resist non-targeted poisoning. In contrast, our work focuses on targeted poisoning. We shall also comment that, while our focus in this work is on instance-targeted attacks for prediction tasks, it is not clear how to even define such (targeted) attacks for robust parameter estimation (e.g., learning Gaussians).

Regarding *certification*, Steinhardt et al. [2017] were the first who studied certification of the *overall risk* under the poisoning attack. However, the more relevant to our paper is the work by Rosenfeld et al. [2020] who introduced the instance-targeted poisoning attack and applied randomized smoothing for certification in this setting. Empirically, they showed how smoothing can provide robustness against label-flipping adversaries. Subsequently, Levine and Feizi [2021] introduced Deep Partition Aggregation (DPA), a novel technique that uses deterministic bagging in order to develop robust predictions against general (arbitrary addition/removal) instance-targeted poisoning. In the same spirit, Chen et al. [2020], Weber et al. [2020], Jia et al. [2020] developed *randomized* bagging/sub-sampling and empirically studied the intrinsic robustness of their methods. predictions.

Finally, we note that while our focus is on *training-time-only* attacks, poisoning attacks can be performed in conjunction with test time attacks, leading to backdoor attacks [Gu et al., 2017, Ji et al., 2017, Chen et al., 2018, Wang et al., 2019, Turner et al., 2019, Diochnos et al., 2019].

## 2 DEFINITIONS

**Basic definitions and notation.** We let $\mathbb{N} = \{0, 1, \ldots\}$ denote the set of integers, $\mathcal{X}$ the input/instance space, and $\mathcal{Y}$ the space of labels. By $\mathcal{Y}^{\mathcal{X}}$ we denote the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. By $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ we denote the set of hypotheses. We use $D$ to denote a distribution over $\mathcal{X} \times \mathcal{Y}$. By $e \sim D$ we state that $e$ is distributed/sampled according to distribution $D$. For a set $\mathcal{S}$, the notation $e \sim \mathcal{S}$ means that $e$ is uniformly sampled from $\mathcal{S}$. By $D^m$ we denote a product distribution over $m$ i.i.d. samples from $D$. By $D_{\mathcal{X}}$ we denote the projection of $D$ over its first coordinate (i.e., the marginal distribution over $\mathcal{X}$). For a function $h \in \mathcal{Y}^{\mathcal{X}}$ and an example $e = (x, y) \in \mathcal{X} \times \mathcal{Y}$, we use $\ell(h, e)$ to denote the loss of predicting $h(x) \in \mathcal{Y}$ while the correct label for $x$ is $y$. Loss will always be non-negative, and when it is in $[0, 1]$, we call it bounded. For classification problems, unless stated differently, we use the 0-1 loss, i.e., $\ell(h, e) = \mathbb{1}[h(x) = y]$. We use $\mathcal{S} \in (\mathcal{X} \times \mathcal{Y})^*$ to denote a training "set", even though more formally it is in fact a sequence. We use $\mathsf{Lrn}$ to denote a learning algorithm that (perhaps randomly) maps a training set $\mathcal{S} \sim D^m$ of any size $m$ to some $h \in \mathcal{Y}^{\mathcal{X}}$. We

call a leaner $\mathsf{Lrn}$ *proper* (with respect to hypothesis class $\mathcal{H}$) if it always outputs some $h \in \mathcal{H}$. $\mathsf{Lrn}(\mathcal{S})(x)$ denotes the prediction on $x$ by the hypothesis returned by $\mathsf{Lrn}(\mathcal{S})$. When $\mathsf{Lrn}$ is randomized, by $y \sim \mathsf{Lrn}(\mathcal{S})(x)$ we state that $y$ is the prediction when the randomness of $\mathsf{Lrn}$ is chosen uniformly. For a randomized $\mathsf{Lrn}$ and the random seed $r$ (of the appropriate length), $\mathsf{Lrn}_r$ denotes the deterministic learner with the hardwired randomness $r$. For a hypothesis $h \in \mathcal{H}$, a loss function $\ell$, and a distribution $D$ over $\mathcal{X} \times \mathcal{Y}$, the population (a.k.a. true) risk of $h$ over $D$ (with respect to the loss $\ell$) is defined as $\mathrm{Risk}(h, D) = \mathbb{E}_{e \sim D}[\ell(h, e)]$, and the empirical risk of $h$ over $\mathcal{S}$ is defined as $\mathrm{Risk}(h, \mathcal{S}) = \mathbb{E}_{e \sim \mathcal{S}}[\ell(h, e)]$. For a hypothesis class $\mathcal{H}$, we say that the realizability assumption holds for a distribution $D$ if there exists an $h \in \mathcal{H}$ such that $\mathrm{Risk}(h, D) = 0$. To add clarity to the text, We use a diamond "$\Diamond$" to denote the end of a technical definition. For a hypothesis class $\mathcal{H}$, we call a data set $\mathcal{S} \sim D^m$ $\varepsilon$-*representative* if $\forall h \in \mathcal{H}, |\mathrm{Risk}(h, D) - \mathrm{Risk}(h, \mathcal{S})| \leq \varepsilon$. A hypothesis class has the *uniform convergence* property, if there is a function $m = m^{\mathcal{H}}_{\mathsf{UC}}(\varepsilon, \delta)$ such that for any distribution $D$, with probability $1 - \delta$ over $\mathcal{S} \sim D^m$, it holds that $\mathcal{S}$ is $\varepsilon$-representative.

**Notation for the poisoning setting.** For simplicity, we work with deterministic strategies, even though our results could be extended directly to randomized adversarial strategies as well. We use $\mathsf{A}$ to denote an adversary who changes the training set $\mathcal{S}$ into $\mathcal{S}' = \mathsf{A}(\mathcal{S})$. This mapping can depend on (the knowledge of) the learning algorithm $\mathsf{Lrn}$ or any other information such as a targeted example $e$ as well as the randomness of $\mathsf{Lrn}$. By $\mathcal{A}$ we refer to a *set* (or *class*) of adversarial mappings and by $\mathsf{A} \in \mathcal{A}$ we denote that the adversary $\mathsf{A}$ belongs to this class. (See below for examples of such classes.) Our adversaries always will have a budget $b \in \mathbb{N}$ that controls how much they can change the training set $\mathcal{S}$ into $\mathcal{S}'$ under some (perhaps asymmetric) distance metric. To explicitly show the budget, we denote the adversary as $\mathsf{A}_b$ and their corresponding classes as $\mathcal{A}_b$. Finally, we let $\mathcal{A}_b(\mathcal{S}) = \{\mathcal{S}' \mid \mathsf{A}_b \in \mathcal{A}_b(\mathcal{S})\}$ be the set of all "adversarial perturbations" of $\mathcal{S}$ when we go over all possible attacks of budget $b$ from the adversary class $\mathcal{A}$.

**Adversary classes.** Here we define the main adversary classes that we use in this work. For more noise models see the work of Sloan [1995].

- $\mathcal{R}ep_b$ **($b$-replacing).** The adversary can replace up to $b$ of the examples in $\mathcal{S}$ (with arbitrary examples) and then put the whole sequence $\mathcal{S}'$ in an arbitrary order. More formally, the adversary is limited to (1) $|\mathcal{S}| = |\mathcal{S}'|$, and (2) by changing the order of the elements in $\mathcal{S}$, one can make the Hamming distance between $\mathcal{S}', \mathcal{S}$ at most $b$. This is essentially the targeted version of the "nasty noise" model introduced by Bshouty et al. [2002].
- $\mathcal{F}lip_b$ **($b$-label flipping).** The adversary can change the label of up to $b$ examples in $\mathcal{S}$ and reorder the final set.
- $\mathcal{A}dd_b$ **($b$-adding).** The adversary adds up to $b$ examples to $\mathcal{S}$ and put them in arbitrary order. Namely, the multi-set $\mathcal{S}'$ has size at most $|\mathcal{S}| + b$ and it holds that $\mathcal{S} \subseteq \mathcal{S}'$.
- $\mathcal{R}em_b$ **($b$-removing).** The adversary removes up to $b$ examples from $\mathcal{S}$ and puts the rest in an arbitrary order. Namely, as multi-sets $|\mathcal{S}'| \geq |\mathcal{S}| - b$ and $\mathcal{S}' \subseteq \mathcal{S}$.[1]

We now define the notions of risk, robustness, certification, and learnability under targeted poisoning attacks for prediction tasks with a focus on classification. We emphasize that in the definitions below, the notions of targeted-poisoning risk and robustness are defined with respect to a *learner* rather than a hypothesis. The reason is that, very often (and in many natural settings) when the data set is changed by the adversary, the learner needs to return a new hypothesis, reflecting the change in the training data,

**Definition 2.1** (Instance-targeted poisoning risk)**.** Let $\mathsf{Lrn}$ be a possibly randomized learner, $\mathcal{A}_b$ be a class of attacks of budget $b$. For a training set $\mathcal{S} \in (\mathcal{X} \times \mathcal{Y})^m$, an example $e = (x, y) \in \mathcal{X} \times \mathcal{Y}$, and randomness $r$, the *targeted poisoning loss* (under attacks $\mathcal{A}_b$) is defined as[2]

$$\ell_{\mathcal{A}_b}(\mathcal{S}, r, e) = \sup_{\mathcal{S}' \in \mathcal{A}_b(\mathcal{S})} \ell(\mathsf{Lrn}_r(\mathcal{S}'), e). \tag{1}$$

For a distribution $D$ over $\mathcal{X} \times \mathcal{Y}$, the *targeted poisoning risk* is defined as

$$\mathrm{Risk}_{\mathcal{A}_b}(\mathcal{S}, r, D) = \mathop{\mathbb{E}}_{e \sim D}[\ell_{\mathcal{A}_b}(\mathcal{S}, r, e)].$$

For a bounded loss function with values in $[0, 1]$ (e.g., the 0-1 loss), we define the *correctness* of the learner for the distribution $D$ under targeted poisoning attacks of $\mathcal{A}_b$ as

$$\mathrm{Cor}_{\mathcal{A}_b}(\mathcal{S}, D) = 1 - \mathrm{Risk}_{\mathcal{A}_b}(\mathcal{S}, D).$$

The above formulation implicitly allows the adversary to depend (and hence "know") on the randomness $r$ of the learning algorithm. We also define *weak targeted-poisoning* loss and risk by using *fresh* learning randomness $r$ unknown to the adversary, when doing the retraining:

$$\ell^{\mathrm{wk}}_{\mathcal{A}_b}(\mathcal{S}, e) = \sup_{\mathcal{S}' \in \mathcal{A}_b(\mathcal{S})} \mathop{\mathbb{E}}_{r}[\ell(\mathsf{Lrn}_r(\mathcal{S}'), e)],$$

$$\mathrm{Risk}^{\mathrm{wk}}_{\mathcal{A}_b}(\mathcal{S}, D) = \mathop{\mathbb{E}}_{e \sim D}[\ell^{\mathrm{wk}}_{\mathcal{A}_b}(\mathcal{S}, e)].$$

In particular, having a small weak targeted-poisoning risk under the 0-1 loss means that for most of the points $e \sim D$ the decisions are correct, and the prediction on $e$ would not change under any $e$-targeted poisoning attacks with high probability over a randomized retraining. $\Diamond$

[1] Note that $b$-replacing attacks are essentially as powerful as any adversary who can add or remove up to $b$ examples arbitrarily, with the only limitation that they preserve the training set size. Our results extend to attacks with up to $b$ additions or removals, however we focus on $b$-replacing attacks for simplicity of presentation.

[2] Note that Equation 1 is equivalent to $\ell_{\mathcal{A}_b}(\mathcal{S}, r, e) = \sup_{\mathsf{A} \in \mathcal{A}_b} \ell(\mathsf{Lrn}_r(\mathsf{A}(\mathcal{S}, r, e)), e)$, because we are choosing the attack over $\mathcal{S}$ after fixing $r, e$.

We now define robustness of predictions, which is more natural for classification tasks, but we state it more generally.

**Definition 2.2** (Robustness under instance-targeted poisoning)**.** Consider the same setting as that of Definition 2.1, and let $\tau > 0$ be a threshold to model when the loss is "large enough". For a data set[3] $\mathcal{S}$ and learner's randomness $r$, we call an example $e = (x, y)$ to be $\tau$-*vulnerable* to a targeted poisoning (of attacks in $\mathcal{A}_b$), if the $e$-targeted adversarial loss is at least $\tau$, namely, $\ell_{\mathcal{A}_b}(\mathcal{S}, r, e) \geq \tau$. For the same $(\mathcal{S}, r, e, \tau)$ we define the *targeted poisoning robustness* (under attacks in $\mathcal{A}$) as the smallest budget $b$ such that $e$ is $\tau$-vulnerable to a targeted poisoning, i.e.,

$$\mathrm{Rob}^{\tau}_{\mathcal{A}}(\mathcal{S}, r, e) = \inf \left\{ b \mid \ell_{\mathcal{A}_b}(\mathcal{S}, r, e) \geq \tau \right\}.$$

If no such $b$ exists, we let $\mathrm{Rob}^{\tau}(\mathcal{S}, r, e) = \infty$.[4] When working with the 0-1 loss (e.g., for classification), we will use $\tau = 1$ and simply write $\mathrm{Rob}_{\mathcal{A}}(\cdot)$ instead. Also note that in this case, $\ell(\mathsf{Lrn}_r(\mathcal{S}'), e) \geq 1$ is simply equivalent to $\mathsf{Lrn}_r(\mathcal{S}')(x) \neq y$. In particular, if $e = (x, y)$ is an example and $\mathsf{Lrn}_r(\mathcal{S})$ is already wrong in its prediction of the label for $x$, then the robustness will be $\mathrm{Rob}_{\mathcal{A}}(\mathcal{S}, r, e) = 0$, as no poisoning will be needed to make the prediction wrong. For a distribution $D$ we define the *expected targeted-poisoning robustness* as $\mathrm{Rob}^{\tau}_{\mathcal{A}}(\mathcal{S}, r, D) = \mathbb{E}_{e \sim D}[\mathrm{Rob}^{\tau}_{\mathcal{A}}(\mathcal{S}, r, e)]$. $\Diamond$

We now formalize when a learner provides certifying guarantees for the produced predictions. For simplicity, we state the definition for the case of 0-1 loss, but it can be generalized to other loss functions by employing a threshold parameter $\tau$ as it was done in Definition 2.2.

**Definition 2.3** (Certifying predictors and learners)**.** A *certifying predictor* (as a generalization of a hypothesis function) is a function $h \colon \mathcal{X} \to \mathcal{Y} \times \mathbb{N}$, where the second output is interpreted as a claim about the robustness of the prediction. When $h(x) = (y, b)$, we define $h_{\mathrm{pred}}(x) = y$ and $h_{\mathrm{cert}}(x) = b$. If $h_{\mathrm{cert}}(x) = b$, the interpretation is that the prediction $y$ shall not change when the adversary performs a $b$-budget poisoning perturbation (defined by the attack model) over the training set used to train $h$.[5] Now, suppose $\mathcal{A}_b$ is an adversary class with budget $b = b(m)$ (where $m$ is the sample complexity) and $\mathcal{A} = \cup_i \mathcal{A}_i$. Also suppose $\mathsf{Lrn}$ is a learning algorithm such that $\mathsf{Lrn}_r(\mathcal{S})$ always outputs a certifying predictor for any data set $\mathcal{S} \in (\mathcal{X} \times \mathcal{Y})^{\star}$. We call $\mathsf{Lrn}$ a *certifying learner* (under the attacks in $\mathcal{A}$) for a specific data set $\mathcal{S} \in (\mathcal{X} \times \mathcal{Y})^{\star}$ and randomness $r$, if the following holds. For all $x \sim D$, if $\mathsf{Lrn}_r(\mathcal{S})(x) = (y, b)$ and if we let $e = (x, y)$,[6] then $\mathrm{Rob}_{\mathcal{A}}(\mathcal{S}, r, e) \geq b$. In other words, to change the prediction $y$ on $x$ (regardless of $y$ being a correct prediction or not), any adversary needs a budget at least $b$. We call $\mathsf{Lrn}$ a *universal* certifying learner if it is a certifying learning for all data sets $\mathcal{S}$. For an adversary class $\mathcal{A} = \cup_{b \in \mathbb{N}} \mathcal{A}_b$, and a certifying learner $\mathsf{Lrn}$ for $(\mathcal{S}, r)$, we define the $b$-certified correctness of $\mathsf{Lrn}$ over $(\mathcal{S}, r, D)$ as the probability of outputting correct predictions while certifying them with robustness at least $b$. Namely,

$$\mathrm{CCor}_{\mathcal{A}_b}(\mathcal{S}, r, D) = \Pr_{(x,y) \sim D} [(y' = y) \wedge (b' \geq b) \text{ where } (y', b') = \mathsf{Lrn}_r(\mathcal{S})(x)] . \Diamond$$

The following definition extends the standard PAC learning framework of Valiant [1984] by allowing targeted-poisoning attacks and asking the leaner now to have small targeted-poisoning risk. This definition is strictly more general than PAC learning, as the trivial attack that does not change the training set, Definition 2.4 below reduces to the standard definition of PAC learning.

**Definition 2.4** (Learnability under instance-targeted poisoning)**.** Let the function $b \colon \mathbb{N} \to \mathbb{N}$ model adversary's budget as a function of sample complexity $m$. A hypothesis class $\mathcal{H}$ is *PAC learnable under targeted poisoning attacks in* $\mathcal{A}_b$, if there is a proper learning algorithm $\mathsf{Lrn}$ such that for every $\varepsilon, \delta \in (0, 1)$ there is an integer $m$ where the following holds. For every distribution $D$ over $\mathcal{X} \times \mathcal{Y}$, if the realizability condition holds[7] (i.e., $\exists h \in \mathcal{H}, \mathrm{Risk}(h, D) = 0$), then with probability $1 - \delta$ over the sampling of $\mathcal{S} \sim D^m$ and $\mathsf{Lrn}$'s randomness $r$, it holds that $\mathrm{Risk}_{\mathcal{A}_b}(\mathcal{S}, r, D) \leq \varepsilon$.

- **Improper learning.** We say that $\mathcal{H}$ is *improperly* PAC learnable under targeted $\mathcal{A}_b$-poisoning attacks, if the same conditions as above hold but using an improper learner that might output functions outside $\mathcal{H}$.[8]
- **Distribution-specific learning.** Suppose $\mathcal{D}$ is the set of all distributions $D$ over $\mathcal{X} \times \mathcal{Y}$ such that the marginal distribution of $D$ over its first coordinate (in $\mathcal{X}$) is a fixed distribution $D_{\mathcal{X}}$ (e.g., isotropic Gaussian in dimension $d$). If all the conditions above (resp. for the improper cases) are only required to hold for distributions $D \in \mathcal{D}$, then we say that the hypothesis class $\mathcal{H}$ is PAC learnable (resp. improperly PAC learnable) under instance distribution $D_{\mathcal{X}}$ and targeted $\mathcal{A}_b$-poisoning.

A hypothesis class is *weakly* (improperly and/or distribution-specific) PAC learnable under targeted $\mathcal{A}_b$-poisoning, if with probability $1 - \delta$ over the sampling of $\mathcal{S} \sim D^m$, it holds that

[3] Even though, in natural attack scenarios the set $\mathcal{S}$ is sampled from $D^m$, Definitions 2.1 and 2.2 are more general in the sense that $\mathcal{S}$ is an arbitrary set.

[4] If the adversary's budget allows it to flip all the labels, in natural settings (e.g., when the hypothesis class contains the complement functions and the learner is a PAC learner), no robustness will be infinite for such attacks.

[5] When using a general loss function, $b$ would be interpreted as the attack budget that is needed to increase the loss over the example $e(x, y)$ (where $y$ is the prediction) to $\tau$.

[6] Note that $y$ might not be the right label

[7] Note that realizability holds while no attack is launched.

[8] We note, however, that whenever the proper or improper condition is not stated, the default is to be proper.

$\mathrm{Risk}^{\mathrm{wk}}_{\mathcal{A}_b}(\mathcal{S}, D) \leq \varepsilon$. A hypothesis class is *certifiably* (improperly and/or distribution-specific) PAC learnable under targeted $\mathcal{A}_b$-poisoning, if we modify the $(\varepsilon, \delta)$ learnability condition as follows. With probability $1-\delta$ over $\mathcal{S} \sim D^m$ and randomness $r$, it holds that (1) $\mathsf{Lrn}$ is a certifying learner for $(\mathcal{S}, r)$, and (2) $\mathrm{CCor}_{\mathcal{A}_b}(\mathcal{S}, r, D) \geq 1-\varepsilon$. A hypothesis class is *universally* certifiably PAC learnable, if it is certifiably PAC learnable using a universal certifying learner $\mathsf{Lrn}$. We call the sample complexity of any learner of the forms above *polynomial*, if the sample complexity $m$ is at most $\mathrm{poly}(1/\varepsilon, 1/\delta) = (1/(\varepsilon\delta))^{O(1)}$. We call the learner *polynomial time*, if it runs in time $\mathrm{poly}(1/\varepsilon, 1/\delta)$, which implies the sample complexity is polynomial as well. $\diamondsuit$

**Remark 2.5** (On defining agnostic learning under instance–targeted poisoning)**.** Definition 2.4 focuses on the realizable setting. However, one can generalize this to the agnostic (non-realizable) case by requiring the following to hold with probability $1-\delta$ over $\mathcal{S} \sim D^m$ and randomness $r$

$$\mathrm{Risk}_{\mathcal{A}_b}(\mathcal{S}, r, D) \leq \varepsilon + \inf_{h \in \mathcal{H}} \mathrm{Risk}(h, r, D).$$

Note that in this definition the learner wants to achieve *adversarial* risk that is $\varepsilon$-close to the risk under *no attack*. One might wonder if there is an alternative definition in which the learner aims to "$\varepsilon$-compete" with the best *adversarial* risk. However, recall that targeted-poisoning adversarial risk is *not* a property of the hypothesis, and it is rather a property of the learner. This leads to the following arguably unnatural criteria that needs to hold with probability $1-\delta$ over $S \sim D^m$ and $r$. (For clarity the learner is explicitly denoted as super-index for $\mathrm{Risk}_{\mathcal{A}_b}$.)

$$\mathrm{Risk}^{\mathsf{Lrn}}_{\mathcal{A}_b}(\mathcal{S}, r, D) \leq \varepsilon + \inf_{L} \mathrm{Risk}^{L}_{\mathcal{A}_b}(\mathcal{S}, r, D)$$

The reason that the above does not trivially hold is that $\mathsf{Lrn}$ needs to satisfy this for *all* distributions $D$ (and most $\mathcal{S}$) simultaneously, while the learner $L$ in the right hand side can depend on $D$ and $\mathcal{S}$.

# 3 OUR RESULTS

We now study the question of learnability under instance-targeted poisoning. We first discuss our positive and negative results in the context of distribution-independent learning. We then turn to the setting of distribution-dependent setting. At the end, we prove some generic relations between risk and robustness, showing how to derive one from the other.

Due to space limitations, all proofs are moved the full version of this paper [Gao et al., 2021].

## 3.1 DISTRIBUTION-INDEPENDENT LEARNING

We start by showing results on distribution-independent learning. We first show that in the realizable setting, for any hypothesis class $\mathcal{H}$ that is PAC-learnable, $\mathcal{H}$ is also PAC learnable under instance-targeted poisoning attacks that can replace up to $b(m) = o(m)$ (e.g., $b(m) = \sqrt{m}$) number of examples arbitrarily. To state the bound of sample complexity of robust learners, we first define the $\lambda(\cdot)$ function based an adversary's budget $b(m)$.

**Definition 3.1** (The $\lambda(\cdot)$ function)**.** Suppose $b(m) = o(m)$. Then for any real number $x$, $\lambda(x)$ returns the minimum $m$ where $m'/b(m') \geq x$ for any $m' > m$. Formally,

$$\lambda(x) = \inf_{m \in \mathcal{N}} \left\{ \forall m' \geq m, \frac{m'}{b(m')} \geq x \right\}.$$

Note that because $b(m) = o(m)$, we have $m/b(m) = \omega_m(1)$, so $\lambda(x)$ is well-defined. $\diamondsuit$

**Theorem 3.2** (Proper learning under weak instance-targeted poisoning)**.** *Let $\mathcal{H}$ be the PAC learnable class of hypotheses. Then, for adversary budget $b(m) = o(m)$, the same class $\mathcal{H}$ is also PAC learnable using randomized learners under* weak *$b$-replacing targeted-poisoning attacks. The proper/improper nature of learning remains the same. Specifically, let $m_{\mathsf{Lrn}}(\varepsilon, \delta)$ be the sample complexity of a PAC learner* $\mathsf{Lrn}$ *for $\mathcal{H}$. Then, there is a learner* $\mathsf{WR}$ *that PAC learns $\mathcal{H}$ under weak $b$-replacing attacks with sample complexity at most*

$$m_{\mathsf{WR}}(\varepsilon, \delta) = \lambda\left(\max\left\{m^2_{\mathsf{Lrn}}\left(\varepsilon, \frac{\delta}{2}\right), \frac{4}{\delta^2}\right\}\right).$$

*Moreover, if $b(m) \leq O(m^{1-\Omega(1)})$, then whenever $\mathcal{H}$ is learnable with a polynomial sample complexity and/or a polynomial-time learner* $\mathsf{Lrn}$*, the robust variant* $\mathsf{WR}$ *will have the same features as well.*

The above theorem shows that targeted-poisoning-robust proper learning is possible for PAC learnable classes using *private* randomness for the learner if $b(m) = o(m)$. Thus, it is natural to ask the following question: can we achieve the stronger (default) notion of robustness as in Definition 2.4 in which the adversarial perturbation can also depend on the (fixed) randomness $r$ of the learner? Also, can this be a learning with certifications? Our next theorem answers these questions positively, yet that comes at the cost of improper learning. Interestingly, the improper nature of the learner used in Theorem (3.3) could be reminiscent of the same phenomenon in *test-time* attacks (a.k.a., adversarial example) where, as it was shown by Montasser et al. [2019], improper learning came to rescue as well.

**Theorem 3.3** (Improper learning and certification under targeted poisoning)**.** *Let $\mathcal{H}$ be (perhaps improperly) PAC learnable. If $b$-replacing attacks have their budget limited to $b(m) = o(m)$, then $\mathcal{H}$ is improperly certifiably PAC learnable under $b$-replacing targeted poisoning attacks. Specifically, let $m_{\mathsf{Lrn}}(\varepsilon, \delta)$ be the sample complexity of a PAC*

*learner for $\mathcal{H}$. Then there is a learner* Rob *that universally certifiably PAC learns $\mathcal{H}$ under $b$-replacing attacks with sample complexity at most*

$$m_{\mathsf{Rob}}(\varepsilon,\delta) = 576\lambda\left(\max\left\{m^2_{\mathsf{Lrn}}\left(\frac{\varepsilon}{12},\frac{\varepsilon}{12}\right),\frac{1}{4\varepsilon^2},\frac{\log\left(\frac{\delta}{2}\right)^2}{\left(\frac{2\sqrt{3}\varepsilon}{3}\right)^4},\frac{\log_2\left(\frac{2}{\delta}\right)}{576}\right\}\right).$$

*Moreover, if $b(m) \leq O(m^{1-\Omega(1)})$ and $\mathcal{H}$ is learnable using a learner with a polynomial sample complexity and/or time, the robust variant* Rob *will have the same features as well.*

We then show that limiting adversary's budget to $b(m) = o(m)$ is essentially necessary for obtaining positive results in the distribution-independent PAC learning setting, as some hypothesis classes with finite-VC dimension are not learnable under targeted poisoning attacks when $b(m) = \Omega(m)$ in a very strong sense: any PAC learner (without attack) would end up having essentially a risk arbitrary close to 1 under attack for any $b(m) = \Omega(m)$ budget given to a $b$-replacing adversary.

We use homogeneous halfspace classifiers, defined in Definition 3.4 below, as an example of hypothesis classes with finite VC dimension. Then in Theorem 3.5, we show that the hypothesis class of halfspaces are not distribution-independently robust learnable against $\Omega(m)$-label flipping instance-targeted attacks.

**Definition 3.4** (Homogeneous halfspace classifiers)**.** A (homogeneous) halfspace classifier $h_\omega : \mathbb{R}^d \to \{0,1\}$ is defined as $h_\omega(x) = \mathrm{Sign}(\omega \cdot x)$, where $\omega$ is a $d$-dimensional vector. We then call $\mathcal{H}_{\mathsf{half}}$ the class of halfspace classifiers $\mathcal{H}_{\mathsf{half}} = \{h_\omega(x) \colon \omega \in \mathbb{R}^d\}$. For simplicity, we may use $\omega$ to refer to both the model parameter and the classifier. $\diamondsuit$

**Theorem 3.5** (Limits of distribution-independent learnability of halfspaces)**.** *Consider the halfspaces hypothesis set $\mathcal{H} = \mathcal{H}_{\mathsf{half}}$ and we aim to learn any distribution over the unit sphere using $\mathcal{H}$. Let the adversary class be $b$-replacing with $b(m) = \beta \cdot m$ for any (even very small) constant $\beta$. For any (even improper) learner* Lrn *one of the following two conditions holds. Either* Lrn *is* not *a PAC learner for the hypothesis class of half spaces (even without attacks) or there exists a distribution $D$ such that $\mathrm{Risk}_{\mathcal{F}lip_b}(\mathcal{S}, D) \geq 1-\sqrt{\sigma}$ with probability $1-\sqrt{\sigma}$ over the selection of $\mathcal{S}$ of sufficiently large $m \geq m_{\mathsf{Lrn}}(\beta \cdot \sigma/6, \sigma/2)$, where $m_{\mathsf{Lrn}}$ is the sample complexity of PAC learner* Lrn.

The idea behind the example of Theorem 3.5 can be found in Figure 1. In particular, consider the uniform distribution over the two circles. In the original (clean) distribution, the points in the top circle are labeled red, and the points in the bottom circle are labeled blue. The adversary can fool the learner on *every* targeted instance, by changing the labels on the targeted point and the *opposite* points on the *other* circle (as shown in the picture, where the two arcs from each circle are flipped). Any learner who tries to learn this distribution would be forced to make a mistake on the targeted point. The actual proof is more subtle and requires randomizing the arcs around the target point.

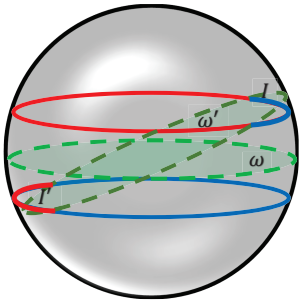



Figure 1: Example for proving Theorem 3.5. The red circle has label 1, and the blue circle has label $-1$. $\omega$ is the ground-truth halfspace with 0 risk, and $\omega'$ is the halfspace that has 0 risk after adversary make replacements.

Note that it was already proved by Bshouty et al. [2002] that, if the adversary can corrupt $b = \Omega(m)$ of the examples, even with *non-targeted* adversary, robust PAC learning is impossible. However, in that case, there is a learning algorithm with error $O(b/m)$. So if, e.g., $b = m/1000$, then non-targeted learning is possible for practical purposes. On the other hand, Theorem 3.5 shows that any PAC learning algorithm in the *no attack* setting, would have essentially risk 1 under *targeted* poisoning.

**Remark 3.6** (Other loss functions)**.** Most of our initial results in this work are proved for the 0-1 loss as the default for classification. Yet, the written proof of Theorem 3.2 holds for any loss function. Theorem 3.3 can also likely be extended to other "natural" losses, but using a more complicated "decision combiner" than the majority. In particular, the learner can now output a label for which "most" sub-models will have "small" risk (parameters most/small shall be chosen carefully). The existence of such a label can probably be proved by a similar argument to the written proof of the 0-1 loss. However, this operation is not poly time.

### 3.2 DISTRIBUTION-SPECIFIC LEARNING

Our previous results are for distribution-independent learning. This still leaves open to study distribution-specific learning. That is, when the input distribution is fixed, one might able to prove stronger results.

We then study the learnability of halfspaces under instance-targeted poisoning on *the uniform distribution over the unit sphere*. Note that one can map all the examples in the $d$-dimensional space to the surface of the unit sphere, and

their relative position to a homogeneous halfspace remains the same. Hence, one can limit both $\omega$ and instance $x \in \mathbb{R}^d \setminus 0^d$ to be unit vectors in $\mathbb{S}^{d-1}$. Therefore, distributions $D_{\mathcal{X}}$ on the unit sphere surface can represent any distribution in the $d$-dimensional space. For example, a $d$-dimensional isotropic Gaussian distribution can be equivalently mapped to the uniform distribution over the unit sphere as far as classification with homogeneous halfspaces is concerned. We note that when the attack is *non-targeted*, it was already shown by Bshouty et al. [2002] that whenever $b(m) = o(m)$, then robust PAC learning is possible (if it is possible in the no-attack setting). Therefore, our results below can be seen as extending the results of [Bshouty et al., 2002] to the *instance-targeted* poisoning attacks.

**Theorem 3.7** (Learnability of halfspaces under the uniform distribution)**.** *In the realizable setting, let $D$ be uniform on the $d$ dimensional unit sphere $\mathbb{S}^{d-1}$ and let adversary's budget for $\mathcal{R}ep_{b(m)}$ be $b(m) = cm/\sqrt{d}$. Then for the halfspace hypothesis set $\mathcal{H}_{\mathsf{half}}$, there exists a deterministic proper certifying learner $\mathsf{CLrn}$ such that the following*

$$\Pr_{\mathcal{S}\leftarrow D^m}\left[\mathrm{CCor}_{\mathcal{R}ep_{b(m)}}(\mathcal{S}, D) \geq 1 - 2\sqrt{2\pi}\cdot c - \sqrt{2\pi d}\cdot\varepsilon\right]$$

*is at least $1-\delta$ for sufficiently large sample complexity $m \geq m^{\mathcal{H}}_{\mathsf{UC}}(\varepsilon, \delta)$, where $m^{\mathcal{H}}_{\mathsf{UC}}$ is the sample complexity of uniform convergence on $\mathcal{H}_{\mathsf{half}}$. So the problem is properly and certifiably PAC learnable under $b$-replacing instance-targeted poisoning attacks.*

For example, when $c = 1/502$, $\varepsilon = c/(100\sqrt{d})$ and $\delta = 0.01$, Theorem 3.7 implies that

$$\Pr_{\mathcal{S}\leftarrow D^m}\left[\mathrm{CCor}_{\mathcal{R}ep_{b(m)}}(\mathcal{S}, D) \geq 99\%\right] \geq 99\%.$$

We also show that the above theorem is essentially optimal, as long as we use proper learning. Namely, for any fixed dimension $d$, with budget $b = O(m/\sqrt{d})$, a $b$-replacing adversary can guarantee success of fooling the majority of examples. Note that for constant $d$, when $m \to \infty$, this is just a constant fraction of data being poisoned, yet this constant fraction can be made arbitrary small when $d \to \infty$.

**Theorem 3.8** (Limits of robustness of PAC learners under the uniform distribution)**.** *In the realizable setting, let $D$ be uniform over the $d$ dimensional unit sphere $\mathbb{S}^{d-1}$. For the halfspace hypothesis set $\mathcal{H}_{\mathsf{half}}$, if $b(m) \geq cm/\sqrt{d}$ for $b$-label flipping attacks $\mathcal{F}lip_b$, for any proper learner $\mathsf{Lrn}$ one of the following two conditions holds. Either $\mathsf{Lrn}$ is* not *a PAC learner for the hypothesis class of half spaces (even without attacks), or for sufficiently large $m \geq m_{\mathsf{Lrn}}(3c/(10\sqrt{d}), \delta)$, with probability $1 - \sqrt{\delta + 2e^{-c^2/18}}$ over the selection of $\mathcal{S}$ we have*

$$\mathrm{Risk}_{\mathcal{R}ep_b}(\mathcal{S}, D) \geq 1 - \sqrt{\delta + 2e^{-c^2/18}},$$

*where $m_{\mathsf{Lrn}}$ is the sample complexity of the learner $\mathsf{Lrn}$.*

For example, when $c = 20$ and $\delta = 0.00009$, we have $\mathrm{Risk}_{\mathcal{F}lip_b}(\mathcal{S}, D) \geq 99\%$.

# 4 EXPERIMENTS

In this section, we study the power of instance-targeted poisoning on the MNIST dataset [LeCun et al., 1998]. We first analyze the robustness of $K$-Nearest Neighbor model, where the robustness can be efficiently calculated empirically. We then empirically study the accuracy under targeted poisoning for multiple other different learners. Previous empirical analysis on instance-targeted poisoning (e.g., Shafahi et al. [2018]) mostly focus on clean-label attacks. In this work, we use attacks of any labels, which lead to stronger attacks compared to clean-label attacks. We also study multiple models in our experiment, while previous work mostly focus on neural networks, and we then compare the performance of different models under the same attack.

$K$-Nearest Neighbor ($K$-NN) is non-parameterized model that memorizes every training example in the dataset. This special structure of $K$-NN allows us to empirically evaluate the robustness to poisoning attacks. The $K$-NN model in this section uses the majority vote defined below.

**Definition 4.1** ($K$-NN learner)**.** For training dataset $\mathcal{S}$ and example $e = (x, y)$, let $\mathcal{N}(x)$ denote the set of $K$ closest examples from $\mathcal{S}$ $e$. Then the prediction of the $K$-NN is

$$h_{\mathsf{KNN}}(x) = \operatorname*{argmax}_{j\in\mathcal{Y}} \sum_{(x_i,y_i)\in\mathcal{N}(x)} \mathbb{1}[y_i = j]. \qquad \Diamond$$

From our definition of poisoning attack and robustness, we can measure the robustness empirically by the following lemma. Similar ideas can also be found in [Jia et al., 2020].

**Lemma 4.2** (Instance-targeted Poisoning Robustness of the $K$-NN learner)**.** *Let $\mathsf{margin}(h_{\mathsf{KNN}}, e)$ be defined as $0$ if $h_{\mathsf{KNN}}(x) \neq y$ and be defined as*

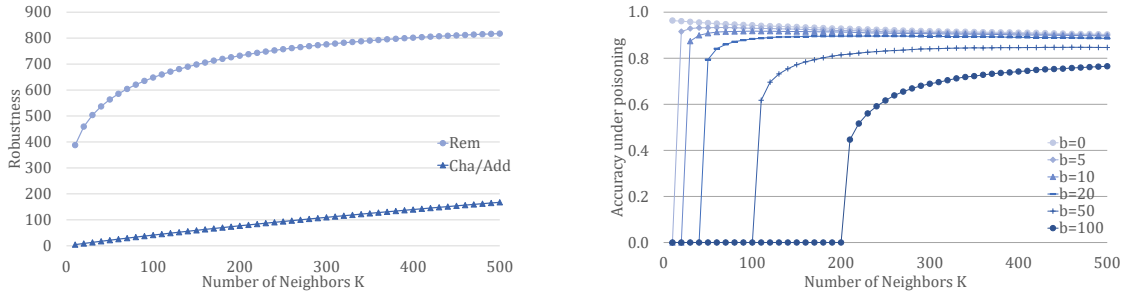
$$\sum_{(x_i,y_i)\in\mathcal{N}(x)} \mathbb{1}[y_i = y] - \max_{j\in\mathcal{Y}, j\neq y} \sum_{(x_i,y_i)\in\mathcal{N}(x)} \mathbb{1}[y_i = j]$$

*otherwise. We then have*

$$\mathrm{Rob}_{\mathcal{R}ep_b}(\mathsf{Lrn}_{\mathsf{KNN}}, \mathcal{S}, e) = \left\lceil \frac{\mathsf{margin}(\mathsf{Lrn}_{\mathsf{KNN}}(\mathcal{S}), e)}{2} \right\rceil.$$

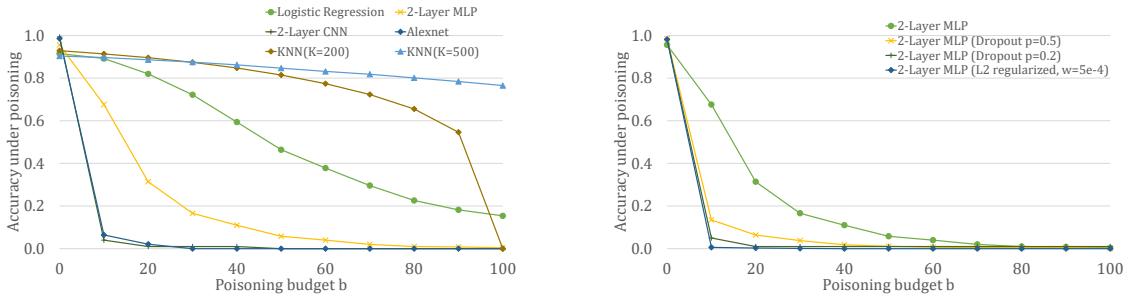
Using Lemma 4.2, one can compute the robustness of the $K$-NN model empirically by calculating the margin for every $e$ in the distribution. We then use the popular digit classification dataset MNIST to measure the robustness.

In the experiment, we use the whole training dataset to train (60, 000 examples), and evaluate the robustness on the testing dataset (10, 000 examples). We calculate the robustness




(a) (b)

Figure 2: Experiment of $K$-Nearest Neighbors on the MNIST dataset. (a) The trend of Robustness $\mathrm{Rob}(\mathsf{Lrn}_{\mathsf{knn}}, \mathcal{S}_{\mathsf{MNIST}}, \mathcal{D})$ on attacks $\mathcal{R}ep$, $\mathcal{A}dd$, and $\mathcal{R}em$, with the increase of number of neighbors $K$. (b) Accuracy of $K$-NN model under $\mathcal{R}ep_b$ with different poisoning budget $b$.




(a) (b)

Figure 3: Accuracy of different learners under $\mathcal{A}dd_b$ instance-targeted poisoning on the MNIST dataset. (a) Compare different learners. (b) Compare dropout and regularization mechanics on Neural Networks.

under $\mathcal{R}ep_b$, $\mathcal{R}em_b$, and $\mathcal{A}dd_b$ attacks. We measure the result with different number of neighbors $K$ present the result in Figure 2a. We also measure the accuracy under poisoning of $\mathcal{R}ep_b$ and report it in Figure 2b. The results in Figure 2 indicates the following message. (1) From Figure 2a, when the number of neighbors $K$ increases, the robustness also increases as expected. The robustness of $K$-NN to $\mathcal{R}ep$ and $\mathcal{A}dd$ increases almost linearly with $K$. (2) The robustness to $\mathcal{R}em$ is much larger than to $\mathcal{R}ep$ and $\mathcal{A}dd$. $\mathcal{R}em$ is a more difficult attack in this scenario. (3) From Figure 2b, when the number of neighbors $K$ increases, the models' accuracy without poisoning slightly decreases. (4) From Figure 2b, $K$-NN keeps around 80% accuracy to $b = 100$ instance-targeted poisoning when $K$ becomes large.

For general learners, measuring their robustness provably under attacks is harder because there is no clear efficient attack that is provably optimal. In this case, we perform a heuristic attack to study the power of $\mathcal{A}dd_b$. The general idea is that for an example $e = (x, y)$, we poison the dataset by adding $b$ copies of $(x, y')$ into the dataset with the second best label $y'$ in $h(x)$, where $b$ is the Adversary's budget. We then report the accuracy under poisoning with different budget $b$ on classifiers including Logistic regression, 2-layer Multi-layer Perceptron (MLP), 2-layer Convolutional Neural Network (CNN), AlexNet and also $K$-NN in Figure 3a. We get the following conclusion: (1) Models that have low risk without poisoning, such as MLP, CNN and AlexNet, typically have low empirical error, which makes it less robust under poisoning. (2) $K$-NN with large $K$ have high accuracy under poisoning compared to other models by sacrificing its clean-label prediction accuracy.

Finally, in Figure 3b we report on our findins about two regularization mechanics, dropout and $L2$-regularization, on the Neural Network learner and whether adding them can provide better robustness against instance-targeted poisoning $\mathcal{A}dd_b$. We use a 2-layer Multi-layer Perceptron (MLP) as the base learner and adds dropout/regularization to the learner. From the figure, we get the following messages: (1) Dropout and regularization help to improve the accuracy without the attacks (when $b = 0$). (2) These mechanics don't help the accuracy with the $\mathcal{A}dd_b$ attacks. The accuracy under attack is worse than the vanilla Neural Network. We conclude that these simple mechanics cannot help the neural net to defend against instance-targeted poisoning.


## Acknowledgements

Mohammad Mahmoody and Ji Gao were supported by NSF CCF-1910681 and CNS-1936799. Amin Karbasi was supported by NSF IIS-1845032 and ONR N00014-19-1-2406.

## References


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.

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

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

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

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.

Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606. PMLR, 2019a.

Ilias Diakonikolas, Daniel M Kane, and Pasin Manurangsi. Nearly tight bounds for robust proper learning of halfspaces with a margin. *arXiv preprint arXiv:1908.11335*, 2019b.

Dimitrios I Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Lower bounds for adversarially robust pac learning. *arXiv preprint arXiv:1906.05815*, 2019.

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.

Ji Gao, Amin Karbasi, and Mohammad Mahmoody. Learning and certification under instance-targeted poisoning, 2021.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Data security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020.

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

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.

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

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017. URL `http://proceedings.mlr.press/v70/koh17a.html`.

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.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defenses against general poisoning attacks. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YUGG2tFuPM`.

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.

Saeed Mahloujifar and Mohammad Mahmoody. Can adversarially robust learning leveragecomputational hardness? In *Algorithmic Learning Theory*, pages 581–609. PMLR, 2019.

Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. Learning under $p$-tampering attacks. In *Algorithmic Learning Theory*, pages 572–596. PMLR, 2018.

Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4536–4543, 2019a.

Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. Universal multi-party poisoning attacks. In *International Conference on Machine Learing (ICML)*, 2019b.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory*, pages 2512–2530. PMLR, 2019.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

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. *arXiv preprint arXiv:1804.00792*, 2018.

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.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

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

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

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

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