Robust and Private Learning of Halfspaces

Badih Ghazi **Ravi Kumar** **Pasin Manurangsi** **Thao Nguyen**
Google Research
Mountain View, CA
{badihghazi, ravi.k53}@gmail.com, {pasin, thaotn}@google.com

Abstract

In this work, we study the trade-off between differential privacy and adversarial robustness under L_2-perturbations in the context of learning halfspaces. We prove nearly tight bounds on the sample complexity of robust private learning of halfspaces for a large regime of parameters. A highlight of our results is that robust *and* private learning is harder than robust *or* private learning alone. We complement our theoretical analysis with experimental results on the MNIST and USPS datasets, for a learning algorithm that is both differentially private and adversarially robust.

1 Introduction

In this work, we study the interplay between two topics at the core of AI ethics and safety: privacy and robustness.

As modern machine learning models are trained on potentially sensitive data, there has been a tremendous interest in privacy-preserving training methods. *Differential privacy (DP)* (Dwork et al., 2006b,a) has emerged as the gold standard for rigorously tracking the privacy leakage of algorithms in general (see, e.g., Dwork and Roth, 2014; Vadhan, 2017, and the references therein), and machine learning models in particular (e.g., Abadi et al., 2016), resulting in several practical deployments in recent years (e.g., Erlingsson et al., 2014; Shankland, 2014; Greenberg, 2016; Apple Differential Privacy Team, 2017; Ding et al., 2017; Abowd, 2018).

Another vulnerability of machine learning models that has also been widely studied recently is with respect to adversarial manipulations of their inputs at test time, with the intention of causing classification errors (e.g., Dalvi et al., 2004; Biggio et al., 2013; Szegedy et al., 2014; Goodfellow et al., 2015; Papernot et al., 2016). Numerous methods have been proposed with the goal of training models that are robust to such adversarial attacks (e.g., Madry et al., 2018; Gowal et al., 2018, 2019; Schott et al., 2019), which in turn has led to new attacks being devised in order to fool these models (Athalye et al., 2018; Carlini and Wagner, 2018; Sharma and Chen, 2017). See (Kolter and Madry, 2018) for a recent tutorial on this topic.

Some recent work has suggested incorporating mechanisms from DP into neural network training to enhance adversarial robustness (Lecuyer et al., 2019; Phan et al., 2020, 2019). Given this existing interplay between DP and robustness, we seek to answer the following natural question:

Is achieving privacy and adversarial robustness harder than achieving either criterion alone?

Recent empirical work has provided mixed response to the question (Song et al., 2019b,a; Hayes, 2020), reporting the success rate of membership inference attacks as a heuristic measure of privacy. Instead, using theoretical analysis and the strict guarantees offered by DP, we formally investigate this question in the classic setting of halfspace learning, and arrive at a near-complete picture.

Background. In order to present our results, we start by recalling some notions from robust learning in the PAC model. Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a (Boolean) hypothesis class on an instance space $\mathcal{X} \subseteq \mathbb{R}^d$. A *perturbation* is defined by a function $\mathbb{P} : \mathcal{X} \to 2^{\mathcal{X}}$, where $\mathbb{P}(x) \subseteq \mathcal{X}$ denotes the set of allowable perturbed instances starting from an instance x. The *robust risk* of a hypothesis h with respect to a distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ and perturbation $\mathbb{P}$ is defined as $\mathcal{R}_{\mathbb{P}}(h, \mathcal{D}) = \Pr_{(\mathbf{x},y)\sim\mathcal{D}}[\exists z \in \mathbb{P}(\mathbf{x}), h(\mathbf{z}) \neq y]$. A distribution $\mathcal{D}$ is said to be *realizable* (with respect to $\mathcal{C}$ and $\mathbb{P}$) iff there exists $h^* \in \mathcal{C}$ such that $\mathcal{R}_{\mathbb{P}}(h^*, \mathcal{D}) = 0$. In the adversarially

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

robust PAC learning problem, the learner is given i.i.d. samples from a realizable distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$, and the goal is to output a hypothesis $h : \mathcal{X} \to \{\pm 1\}$ such that with probability $1-\xi$ it holds that $\mathcal{R}_{\mathbb{P}}(h, \mathcal{D}) \leq \alpha$. We refer to ξ as the *failure probability* and α as the *accuracy parameter*. A learner is said to be *proper* if the output hypothesis h belongs to $\mathcal{C}$; otherwise, it is said to be *improper*.

We focus our study on the concept class of *halfspaces*, i.e., $\mathcal{C}_{\text{halfspaces}} := \{h_{\mathbf{w}} \mid \mathbf{w} \in \mathbb{R}^d\}$ where $h_{\mathbf{w}}(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle)$, in this model with respect to L_2 perturbations, i.e., $\mathbb{P}_\gamma(\mathbf{x}) := \{\mathbf{z} \in \mathcal{X} : \|\mathbf{z} - \mathbf{x}\|_2 \leq \gamma\}$ for *margin* parameter $\gamma > 0$. We assume throughout that the domain of our functions is bounded in the d-dimensional Euclidean unit ball $\mathbb{B}^d := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 \leq 1\}$. We also write $\mathcal{R}_\gamma$ as a shorthand for $\mathcal{R}_{\mathbb{P}_\gamma}$. An algorithm is said to be a (γ, γ')-*robust learner* if, for any realizable distribution $\mathcal{D}$ with respect to $\mathcal{C}_{\text{halfspaces}}$ and $\mathbb{P}_\gamma$, using a certain number of samples, it outputs a hypothesis h such that w.p. $1 - \xi$, we have $\mathcal{R}_{\gamma'}(h, \mathcal{D}) \leq \alpha$, where $\alpha, \xi > 0$ are sufficiently smaller than some positive constant. We are especially interested in the case[1] where γ' is close to γ; for simplicity, we use $\gamma' = 0.9\gamma$ as a representative setting throughout. In robust learning, the main quantities of interest are the *sample complexity*, i.e., the minimum number of samples needed to learn, and the *running time* of the learning algorithm.

We use the standard terminology of DP. Recall that two datasets $\mathbf{X}$ and $\mathbf{X}'$ are *neighbors* if $\mathbf{X}'$ results from adding or removing a single data point from $\mathbf{X}$.

Definition 1 (Differential Privacy (DP) (Dwork et al., 2006b,a))**.** *Let $\epsilon, \delta \in \mathbb{R}_{\geq 0}$. A randomized algorithm $\mathbb{A}$ taking as input a dataset is said to be* (ϵ, δ)-differentially private *(denoted by (ϵ, δ)-DP) if for any two neighboring datasets $\mathbf{X}$ and $\mathbf{X}'$, and for any subset S of outputs of $\mathbb{A}$, it is the case that $\Pr[\mathbb{A}(\mathbf{X}) \in S] \leq e^\epsilon \cdot \Pr[\mathbb{A}(\mathbf{X}') \in S] + \delta$. If $\delta = 0$, $\mathbb{A}$ is said to be* ϵ-differentially private *(denoted by ϵ-DP).*

As usual, ϵ should be thought of as a small constant, whereas δ should be negligible in the dataset size. We refer to the case where $\delta = 0$ as *pure-DP*, and the case where $\delta > 0$ as *approximate-DP*.

Our Results. We assume that $\epsilon \leq O(1)$ unless otherwise stated, and that $\gamma, \alpha, \xi > 0$ are sufficiently smaller than some positive constant. We will not state these assumptions explicitly here for simplicity; interested readers may refer to (the first lines of) the proofs for the exact upper bounds that are imposed.

[1] It is necessary to have $\gamma' < \gamma$; when $\gamma = \gamma'$, proper (γ, γ')-robust learning is as hard as general proper learning of halfspace (see, e.g., Diakonikolas et al., 2020), which is impossible with any finite number of samples under DP (Bun et al., 2015).

We first prove that robust learning with pure-DP requires $\Omega(d)$ samples.

Theorem 2. *Any ϵ-DP $(\gamma, 0.9\gamma)$-robust (possibly improper) learner has sample complexity $\Omega(d/\epsilon)$.*

In the private but non-robust setting (i.e., for an ϵ-DP $(\gamma, 0)$-robust learner[2]), Nguyen et al. (2020) showed that $O(1/\gamma^2)$ samples suffice. Together with earlier known results showing that $(\gamma, 0.9\gamma)$-robust learning (without privacy) only requires $O(1/\gamma^2)$ samples (e.g., Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2002), our result gives a separation between robust private learning and private learning alone or robust learning alone, whenever $d \gg 1/\gamma^2$.

For the case of approximate-DP, we establish a lower bound of $\Omega(\min\{\sqrt{d}/\gamma, d\})$, which holds only against *proper* learners. As for Theorem 2 in the context of pure-DP, this result implies a similar separation in the approximate-DP proper learning setting.

Theorem 3. *Let $\epsilon < 1$. Any $(\epsilon, o(1/n))$-DP $(\gamma, 0.9\gamma)$-robust proper learner has sample complexity $n = \Omega(\min\{\sqrt{d}/\gamma, d\})$.*

Our proof technique can also be used to improve the lower bound for DP $(\gamma, 0)$-robust learning. Specifically, Nguyen et al. (2020) show an $\Omega(1/\gamma^2)$ lower bound for *proper* $(\gamma, 0)$-robust learners with *pure*-DP. We extend it to hold even for *improper* $(\gamma, 0)$-robust learners with *approximate*-DP:

Theorem 4. *For any $\epsilon > 0$, there exists $\delta > 0$ such that any (ϵ, δ)-DP $(\gamma, 0)$-robust (possibly improper) learner has sample complexity $\Omega\left(\frac{1}{\epsilon\gamma^2}\right)$. Moreover, this holds even when $d = O(1/\gamma^2)$.*

Finally, we provide algorithms with nearly matching upper bounds. For pure-DP, we prove the following, which matches the lower bound in Theorem 2 to within a constant factor when $d \geq 1/\gamma^2$.

Theorem 5. *There is an ϵ-DP $(\gamma, 0.9\gamma)$-robust learner with sample complexity[3] $O_\alpha\left(\frac{1}{\epsilon}\max\{d, \frac{1}{\gamma^2}\}\right)$.*

For approximate-DP, it is already possible[4] to achieve a sample complexity[3] of $n = \tilde{O}_\alpha(\sqrt{d}/\gamma)$ (Nguyen et al., 2020; Bassily et al., 2014) but the running time is $\Omega(n^2 d)$.[5] We give a faster algorithm with running time $O_\alpha(nd/\gamma)$.

[2] This means that the output hypothesis only needs to have a small classification error, but may have a large robust risk.

[3] Here $O_\alpha(\cdot)$ hides a factor of $\text{poly}(1/\alpha)$, and $\tilde{O}(\cdot)$ hides a factor of $\text{poly}\log(1/(\alpha\gamma\delta))$.

[4] This can be achieved by running the DP ERM algorithm of (Bassily et al., 2014) with the hinge loss; see (Nguyen et al., 2020) for the analysis.

[5] Specifically, Nguyen et al. (2020) uses the DP empirical

	Bounds	Robust		Non-robust	
		Proper	Improper	Proper	Improper
Non-private	Tight	$\Theta(1/\gamma^2)$			
Pure-DP	Upper	$O(d)$ (Theorem 5)		$\tilde{O}(1/\gamma^2)$	
	Lower	$\Omega(d)$ (Theorem 2)		$\Omega(1/\gamma^2)$	$\Omega(1/\gamma^2)$ (Theorem 4)
Approximate-DP	Upper	$\tilde{O}(\sqrt{d}/\gamma)$ †		$\tilde{O}(1/\gamma^2)$	
	Lower	$\Omega(\sqrt{d}/\gamma)$ (Theorem 3)	$\Omega(1/\gamma^2)$ (Theorem 4)	$\Omega(1/\gamma^2)$ (Theorem 4)	

Table 1: Trade-offs between privacy and robustness. The *robust* column corresponds to $(\gamma, 0.9\gamma)$-robust learners, whereas the *non-robust* column corresponds to $(\gamma, 0)$-robust learners. For simplicity of presentation, we assume $\alpha, \epsilon, \xi \in (0,1)$ are sufficiently small constants, $d \geq 1/\gamma^2$, and for approximate-DP lower bounds, that $\delta = o(1/n)$. While approximate-DP upper bounds (marked with†) can already be derived from previous work, we give a faster algorithm (Theorem 6). For DP, known results are from (Nguyen et al., 2020); for the non-private case, the results follow, e.g., from (Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2002).

Theorem 6. *There is an (ϵ, δ)-DP $(\gamma, 0.9\gamma)$-robust learner with sample complexity $n = \tilde{O}_\alpha\left(\frac{1}{\epsilon} \cdot \max\left\{\frac{\sqrt{d}}{\gamma}, \frac{1}{\gamma^2}\right\}\right)$ and running time $\tilde{O}_\alpha(nd/\gamma)$.*

Our theoretical results and those from prior works are summarized in Table 1. Notice that the non-private robust setting and the non-robust private setting each requires only $O(1/\gamma^2)$ samples, whereas our results show that the private and robust setting requires either $\Omega(d)$ samples (for pure-DP) or $\Omega(\sqrt{d}/\gamma)$ samples (for approximate-DP). This separation positively answers the question central to our study.

We complement our theoretical results by empirically evaluating our algorithm (from Theorem 6) on the MNIST (LeCun et al., 2010) and USPS (Hull, 1994) datasets. Our results show that it is possible to achieve both robustness and privacy guarantees while maintaining reasonable performance. We further provide evidence that models trained via our algorithm are more resilient to adversarial noise compared to neural networks trained via DP-SGD (Abadi et al., 2016).

Organization. In the two following sections, we describe in detail the ideas behind each of our proofs. We then present our experimental results in Section 4. Finally, we discuss additional related work and several open questions in Sections 5 and 6 respectively. Due to space constraints, all missing proofs and additional experiments are deferred to the Supplementary Material.

2 Sample Complexity Lower Bounds

In this section, we explain the high-level ideas behind each of our sample complexity lower bounds. Our pure-DP lower bound is based on a packing framework and our approximate-DP lower bounds are based on fingerprinting codes.

2.1 Pure-DP Lower Bound (Theorem 2)

We use the *packing framework*, a DP lower bound proof technique that originated in (Hardt and Talwar, 2010). Roughly speaking, to apply this framework, we have to construct many input distributions for which the sets of valid outputs for each distribution are disjoint (hence the name "packing"). In our context, this means that we would like to construct distributions $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(K)}$ such that the sets $G^{(i)}$ of hypotheses with small robust risk on $\mathcal{D}^{(i)}$ are disjoint. Once we have done this, the packing framework immediately gives us a lower bound of $\Omega(\log K/\epsilon)$ on the sample complexity; below we describe a construction for $K = 2^{\Omega(d)}$ distributions, which yields the desired $\Omega(d/\epsilon)$ lower bound in Theorem 2.

Our construction proceeds by picking unit vectors $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(K)}$ that are nearly orthogonal, i.e., $\left|\left\langle \mathbf{w}^{(i)}, \mathbf{w}^{(j)} \right\rangle\right| < 0.01$ for all $i \neq j$. It is not hard to see (and well-known) that such vectors exist for $K = 2^{\Omega(d)}$. We then let $\mathcal{D}^{(i)}$ be the uniform distribution on $(1.01\gamma \cdot \mathbf{w}^{(i)}, +1), (-1.01\gamma \cdot \mathbf{w}^{(i)}, -1)$.

Now, let $G^{(i)}$ denote the set of hypotheses h for which $\mathcal{R}_{0.9\gamma}(h, \mathcal{D}^{(i)}) < 0.5$. Since our distribution $\mathcal{D}^{(i)}$ is uniform on two elements, we must have that $\mathcal{R}_{0.9\gamma}(h, \mathcal{D}^{(i)}) = 0$ for all $h \in G^{(i)}$. To see that $G^{(i)}$ and $G^{(j)}$ are disjoint for any $i \neq j$, notice that, since $\left|\left\langle \mathbf{w}^{(i)}, \mathbf{w}^{(j)} \right\rangle\right| < 0.01$, the point $1.01\gamma \cdot \mathbf{w}^{(i)}$ is within distance 1.8γ from the point $-1.01\gamma \cdot \mathbf{w}^{(j)}$. This means that any hypothesis h cannot correctly classify both $(1.01\gamma \cdot \mathbf{w}^{(i)}, 1)$ and $(-1.01\gamma \cdot \mathbf{w}^{(i)}, -1)$ with margin at least 0.9γ, which implies that $G^{(i)} \cap G^{(j)} = \emptyset$. This completes our proof sketch.

We end by remarking that the previous work of Nguyen et al. (2020) also uses a packing argument; the main differences between our construction and theirs are in the choice of the distributions $\mathcal{D}^{(i)}$ and the proof of disjointness of the $G^{(i)}$'s. Our construction of $\mathcal{D}^{(i)}$ is in fact simpler, since our proof of disjointness can

risk minimization algorithm of (Bassily et al., 2014) with the hinge loss; however, the latter requires $\Omega(n^2)$ iterations and each iteration requires $\Omega(d)$ time.

rely on the robustness guarantee. These differences are inherent, since our lower bound holds even against *improper* learners, i.e., when the output hypothesis may not be a halfspace, whereas the lower bound of Nguyen et al. (2020) only holds against the particular case of *proper* learners.

2.2 Approximate-DP Lower Bound (Theorem 3)

We reduce from a lower bound from the line of works (Bun et al., 2018; Dwork et al., 2015; Steinke and Ullman, 2016, 2017) inspired by *fingerprinting codes*. Specifically, these works consider the so-called *attribute mean* problem, where we are given a set of vectors drawn from some (hidden) distribution $\mathcal{D}$ on $\{\pm 1\}^d$ and the goal is to compute the mean. It is known that getting an estimate to within 0.1 of the true mean in each coordinate requires $\Omega(\sqrt{d})$ samples. In fact, Steinke and Ullman (2017) show that even outputting a vector with a "non-trivial" dot product[6] with the mean already requires $\Omega(\sqrt{d})$ samples. This *almost* implies our desired lower bound: the only remaining step is to turn $\mathcal{D}$ to a distribution that is realizable with margin γ^*. We do this by conditioning $\mathcal{D}$ on only points $\mathbf{x}$ with a sufficiently large dot product with the true mean, and then adding both $(\mathbf{x}, +1)$ and $(-\mathbf{x}, -1)$ to our distribution. This reduction gives an $\Omega(\sqrt{d})$ lower bound on the sample complexity of $(\gamma^*, 0.9\gamma^*)$-robust proper learners, for some absolute constant margin parameter $\gamma^* > 0$.

To get an improved bound for a smaller margin γ, we "embed" $\Omega(1/\gamma^2)$ hard instances above in each of $O(\gamma^2 d)$ dimensions. More specifically, let $T = \gamma^*/\gamma$; for each $i \in [T^2]$ we create a distribution $\mathcal{D}^{(i)}$ that is the hard distribution from the previous paragraph in $d' := d/T$ dimensions embedded onto coordinates $d'(i-1)+1, \ldots, d'i$. We then let the distribution $\mathcal{D}'$ be the (uniform) mixture of $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(T^2)}$. Since each $\mathcal{D}^{(i)}$ is realizable with margin γ^* via some halfspace $\mathbf{w}^{(i)}$, we may take $\mathbf{w}^* := \frac{1}{T}\sum_{i\in[T^2]} \frac{\mathbf{w}^{(i)}}{\|\mathbf{w}^{(i)}\|}$ to realize the distribution $\mathcal{D}'$ with margin $\frac{1}{T}\cdot\gamma^* = \gamma$ as desired.

Now, to find any $\mathbf{w}$ with small $\mathcal{R}_{0.9\gamma}(h_{\mathbf{w}}, \mathcal{D}')$, we roughly have to solve (most of) the T^2 instances $\mathcal{D}^{(i)}$'s. Recall that solving each of these instances requires $\Omega(\sqrt{d'})$ samples. Thus, the combined instance requires $\Omega(T^2 \cdot \sqrt{d'}) = \Omega(1/\gamma^2 \cdot \sqrt{\gamma^2 d}) = \Omega(\sqrt{d}/\gamma)$ samples.

[6] Specifically, this holds when the output vector has ℓ_2-norm at most $\sqrt{d}$ and the dot product is at least ζd for any constant $\zeta > 0$.

2.3 Non-Robust DP Learning Lower Bound (Theorem 4)

This lower bound once again uses the "embedding" technique described above. Here we start with a hard one-dimensional instance, which is simply the uniform distribution on $(x, -1), (x, +1)$ where x is either $+1$ or -1. When $\delta > 0$ is sufficiently small (depending on ϵ), it is simple to show that any (ϵ, δ)-DP $(1, 0)$-learner for this instance requires $\Omega(1/\epsilon)$ samples. Similar to the previous proof overview, we embed $1/\gamma^2$ such instances into $1/\gamma^2$ dimensions. Since the one-dimensional instance requires $\Omega(1/\epsilon)$ samples, the combined instance requires $\Omega(1/(\epsilon\gamma^2))$ samples, thereby yielding Theorem 4.

3 Sample-Efficient Algorithms

In this section, we present our algorithms for robust and private learning. Our pure-DP algorithm is based on an improved analysis of the exponential mechanism. Our approximate-DP algorithm is based on a private, batched version of the perceptron algorithm.

In the following discussions, we assume that $\mathbf{w}^*$ is an (unknown) optimal halfspace with respect to the input distribution $\mathcal{D}$ and L_2-perturbations with margin parameter γ, i.e., that $\mathbf{w}^*$ satisfies $\mathcal{R}_\gamma(h_{\mathbf{w}^*}, \mathcal{D}) = 0$. We may assume without loss of generality that $\|\mathbf{w}^*\| = 1$.

3.1 Pure-DP Algorithm (Theorem 5)

Theorem 5 is shown via the exponential mechanism (EM) (McSherry and Talwar, 2007). Our guarantee is an improvement over the "straightforward" analysis of EM on a (0.1γ)-net of the unit sphere in $\mathbb{R}^d$, which gives an upper bound of $O_\alpha(d\log(1/\gamma)/\epsilon)$ (Nguyen et al., 2020). On the other hand, when $d \geq 1/\gamma^2$, our sample complexity is $O_\alpha(d/\epsilon)$. The intuition behind our improvement is that, if we take a random unit vector $\mathbf{w}$ such that $\langle \mathbf{w}, \mathbf{w}^* \rangle \geq 0.99$, then it already gives a small robust risk (in expectation) when $d \gg 1/\gamma^2$ because the component of $\mathbf{w}$ orthogonal to $\mathbf{w}^*$ is a random $(d-1)$-dimensional vector of norm less than one, meaning that in expectation it only affects the margin by $O(1/\sqrt{d}) \ll 0.1\gamma$. Now, a random unit vector satisfies $\langle \mathbf{w}, \mathbf{w}^* \rangle \geq 0.99$ with probability $2^{-O(d)}$, which (roughly speaking) means that EM should only require $O_\alpha(d/\epsilon)$ samples.

3.2 Approximate-DP Algorithm (Theorem 6)

To prove Theorem 6, we use the *DP Batch Perceptron* algorithm presented in Algorithm 1. DP Batch Perceptron is the batch and privatized version of the so-called *margin perceptron* algorithm (Duda and Hart, 1973;

Collobert and Bengio, 2004). That is, in each iteration, we randomly sample a batch of samples and for each sample $(\mathbf{x}, y)$ in the batch that is not correctly classified with margin γ', we add $y \cdot \mathbf{x}$ to the current weight of the halfspace. Furthermore, we add some Gaussian noise to the weight vector to make this algorithm private. We also have a "stopping condition" that terminates whenever the number of samples mislabeled at margin γ' is sufficiently small. (We add Laplace noise to the number of such samples to make it private.) To get a $(\gamma, 0.9\gamma)$-robust learner, it suffices for us to set, e.g., $\gamma' = 0.95\gamma$; we use this value of γ' in the subsequent discussions.

Algorithm 1 DP Batch Perceptron

DP-Batch-Perceptron$_{\gamma', p, T, b, \sigma}(\{(\mathbf{x}_j, y_j)\}_{j \in [n]})$

$\mathbf{w}_0 \leftarrow \mathbf{0}$
for $i = 1, \ldots, T$
  $S_i \leftarrow$ a set of samples where each $(\mathbf{x}_j, y_j)$ is independently included w.p. p
  $M_i \leftarrow \emptyset$
  for $(\mathbf{x}, y) \in S_i$
    if $\operatorname{sgn}\left(\left\langle \frac{\mathbf{w}_{i-1}}{\|\mathbf{w}_{i-1}\|}, \mathbf{x} \right\rangle - y \cdot \gamma'\right) \neq y$
      $M_i \leftarrow M_i \cup \{(\mathbf{x}, y)\}$
  Sample $\nu_i \sim \mathrm{Lap}(b)$
  if $|M_i| + \nu_i < 0.3\alpha pn$
    return $\mathbf{w}_{i-1}/\|\mathbf{w}_{i-1}\|$
  $\mathbf{u}_i \leftarrow \sum_{(\mathbf{x}, y) \in M_i} y \cdot \mathbf{x}$
  Sample $\mathbf{g}_i \sim \mathcal{N}(0, \sigma^2 \cdot I_{d \times d})$
  $\mathbf{w}_i \leftarrow \mathbf{w}_{i-1} + \mathbf{u}_i + \mathbf{g}_i$
return FAIL

Before we dive into the details of the proof, we remark that our runtime reduction, compared to the generic algorithm from (Bassily et al., 2014), comes from the fact that, in the accuracy analysis, we only need the number of iterations T to be $O_\alpha(1/\gamma^2)$, similar to the perceptron algorithm (Novikoff, 1963). On the other hand, the generic theorem of Bassily et al. (2014) requires $n^2 = \tilde{O}_\alpha(d^2/\gamma^2)$ iterations.

The accuracy analysis of DP Batch Perceptron follows the blueprint of that of perceptron (Novikoff, 1963). Specifically, we keep track of the following two quantities: $\langle \mathbf{w}_i, \mathbf{w}^* \rangle$, the dot product between the current halfspace $\mathbf{w}_i$ and the "true" halfspace $\mathbf{w}^*$, and $\|\mathbf{w}_i\|$, the (Euclidean) norm of $\mathbf{w}_i$. We would like to show that, after the first few iterations, $\langle \mathbf{w}_i, \mathbf{w}^* \rangle$ increases at a faster rate than $\|\mathbf{w}_i\|$. Since $\langle \mathbf{w}_i, \mathbf{w}^* \rangle$ is bounded above by $\|\mathbf{w}_i\|$, we may use this to bound the number T of iterations required for the algorithm to converge.

For simplicity, we assume that in each iteration $|M_i|$ is equal to $m > 0$. Let us first consider the case where no noise is added (i.e., $\sigma = 0$). From the definition of $\mathbf{w}^*$, it is simple[7] to check that $\langle \mathbf{w}^*, y \cdot \mathbf{x} \rangle \geq \gamma$ for all samples $(\mathbf{x}, y)$. This means that $\langle \mathbf{w}^*, \mathbf{u}_i \rangle \geq \gamma m$, resulting in

$$\langle \mathbf{w}^*, \mathbf{w}_i \rangle \geq \langle \mathbf{w}^*, \mathbf{w}_{i-1} \rangle + \gamma m. \tag{1}$$

On the other hand, the check condition before we add each example $(\mathbf{x}, y)$ to M_i (Line 6) ensures that $\langle \mathbf{w}_i, y \cdot \mathbf{x} \rangle \leq \gamma' \cdot \|\mathbf{w}_i\|$ for all $(\mathbf{x}, y) \in M_i$. From this one can derive the following bound:

$$\|\mathbf{w}_i\| \leq \|\mathbf{w}_{i-1}\| + \gamma' m + \frac{0.5m^2}{\|\mathbf{w}_{i-1}\|},$$

which, when $\|\mathbf{w}_{i-1}\| \geq 50m/\gamma$, implies that

$$\|\mathbf{w}_i\| \leq \|\mathbf{w}_{i-1}\| + 0.96\gamma m. \tag{2}$$

Combining (1) and (2), we arrive at

$$50m/\gamma + 0.96\gamma mi \geq \|\mathbf{w}_i\| \geq \langle \mathbf{w}^*, \mathbf{w}_i \rangle \geq \gamma mi.$$

This implies that the algorithm must stop after $T = O\left(1/\gamma^2\right)$ iterations. (When $m = 1$, this noiseless analysis is essentially the same as the original convergence analysis of perceptron (Novikoff, 1963).)

The previous paragraphs outlined the analysis for the noiseless case where $\sigma = 0$. Next, we will describe how the noise σ affects the analysis and our choices of parameters. Roughly speaking, we would like the inequalities (1) and (2) to "approximately" hold even after adding noise. In particular, this means that we would like the right-hand side of these inequalities to be affected by at most $o(\gamma m)$ by the noise addition, with high probability. This condition will determine our selection of parameters.

For (1), the inclusion of the noise term $\mathbf{g}_i$ adds to the right-hand side by $\langle \mathbf{w}^*, \mathbf{g}_i \rangle$. The expectation of this term is $\|\mathbf{w}^*\| \cdot \sigma \leq \sigma$, which means that it suffices to ensure that $m \geq \tilde{\omega}(\sigma/\gamma)$. For (2), it turns out that the dominant additional term is $\frac{\|\mathbf{g}_i\|^2}{\|\mathbf{w}_{i-1}\|}$ which, under the assumption that $\|\mathbf{w}_{i-1}\| \geq 50m/\gamma$, is at most $O(\gamma\|\mathbf{g}_i\|^2/m)$; this term is $O(\gamma d\sigma^2/m)$ in expectation. Since we would like this term to be $o(\gamma m)$, it suffices to have $m = \tilde{\omega}(\sigma \cdot \sqrt{d})$. By combining these two requirements, we may pick $m = \sigma \cdot \tilde{\omega}(\sqrt{d} + 1/\gamma)$. We remark that the number of iterations still remains $T = O(1/\gamma^2)$, as in the noiseless case above.

While we have so far assumed for simplicity that $|M_i| = m$ in all iterations, in the actual analysis we only require that $|M_i| \geq m$. Furthermore, it is simple to show that, as long as the current hypothesis

[7] Specifically, since $\mathcal{R}_\gamma(h_{\mathbf{w}^*}, \mathcal{D}) = 0$, we have $y \langle \mathbf{w}^*, \mathbf{z} \rangle \geq 0$ for all $\mathbf{z}$ such that $\|\mathbf{x} - \mathbf{z}\| \leq \gamma$. Plugging in $\mathbf{z} = \mathbf{x} - \gamma y \mathbf{w}^*$ yields the claimed inequality.

has robust risk significantly more than α, we will have $|M_i| \geq \Omega_\alpha(pn)$ with high probability. Combining with the previous paragraph, this gives us the following condition (assuming that α is constant):

$$pn \geq \sigma \cdot \tilde{\omega}(\sqrt{d} + 1/\gamma).$$

This leads us to pick the following set of parameters:

$$n = \tilde{O}\left(\frac{1}{\gamma}\left(\sqrt{d} + \frac{1}{\gamma}\right)\right), \quad p = \tilde{O}(\gamma), \quad \sigma = \tilde{O}(1).$$

The privacy analysis of our algorithm is similar to that of DP-SGD (Bassily et al., 2014). Specifically, by the choice of $\sigma = \tilde{O}(1)$ and subsampling rate $p = \tilde{O}(\gamma)$, each iteration of the algorithm is $\left(O(\gamma\epsilon), O(\gamma^2\delta)\right)$-DP (e.g., Dwork et al., 2010; Balle et al., 2018). Since the number of iterations is $T = O(1/\gamma^2)$, advanced composition theorem (Dwork et al., 2010) implies that the entire algorithm is $(O(\sqrt{T \cdot \gamma\epsilon}), O(T \cdot \gamma^2\delta)) = (\epsilon, \delta)$-DP as desired.

We end by noting that, despite the popularity of perceptron-based algorithms, we are not aware of any work that analyzes the above noised and batched variant. The most closely related analysis we are aware of is that of Blum et al. (2005), whose algorithm uses the entire dataset in each iteration. While it is possible to adapt their analysis to the batch setting, it unfortunately does not give an optimal sample complexity. Specifically, their analysis requires the batch size to be $\Omega(\sqrt{d}/\gamma)$, resulting in sample complexity of $\Omega(\sqrt{d}/\gamma^2)$. On the other hand, our more careful analysis works even with batch size $\tilde{O}(\sqrt{d} + 1/\gamma)$, which results in the desired $\tilde{O_{\alpha,\epsilon}}(\sqrt{d}/\gamma)$ sample complexity when $d \geq 1/\gamma^2$.

4 Experiments

We run our DP Batch Perceptron algorithm on the MNIST (LeCun et al., 2010) and USPS (Hull, 1994) datasets, both of which involve 10-class digit classification. We train a separate halfspace classifier $\mathbf{w}^{(y)}$ for each class $y \in \{1, \ldots, 10\}$ for one epoch. To predict on an image $\mathbf{x}$, we output a class y^* that maximizes $\langle \mathbf{w}^{(y^*)}, \mathbf{x} \rangle$. We tune batch size as a hyperparameter with values 1, 10, 50, 100, 500, 1000, and γ' with values 1, 0.1, 0.01, 0.001, 0.0001. Each set of experiments is repeated for 20 random trials. To reduce the number of hyperparameters, we slightly modify our algorithm so that we do not stop early (i.e., removing Lines 9 and 10) but instead return the weight vector w_T at the end of the Tth iteration (where T is set in the algorithm).

The standard deviation σ of the Gaussian noise added is determined by a fixed (ϵ, δ)-DP budget, computed using Renyi DP (Abadi et al., 2016; Mironov, 2017). The calculations for this follow the implementation in the official TensorFlow Privacy repository (`https://github.com/tensorflow/privacy`). For experiments with varying ϵ (first column of Figure 1), we fix δ to 10^{-5} for MNIST and 10^{-4} for USPS. We observe that despite the robustness and privacy constraints, DP Batch Perceptron still achieves competitive accuracy on both datasets. We also report performance with varying δ values of $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$ (second column of Figure 1), while keeping ϵ fixed at 1.0.

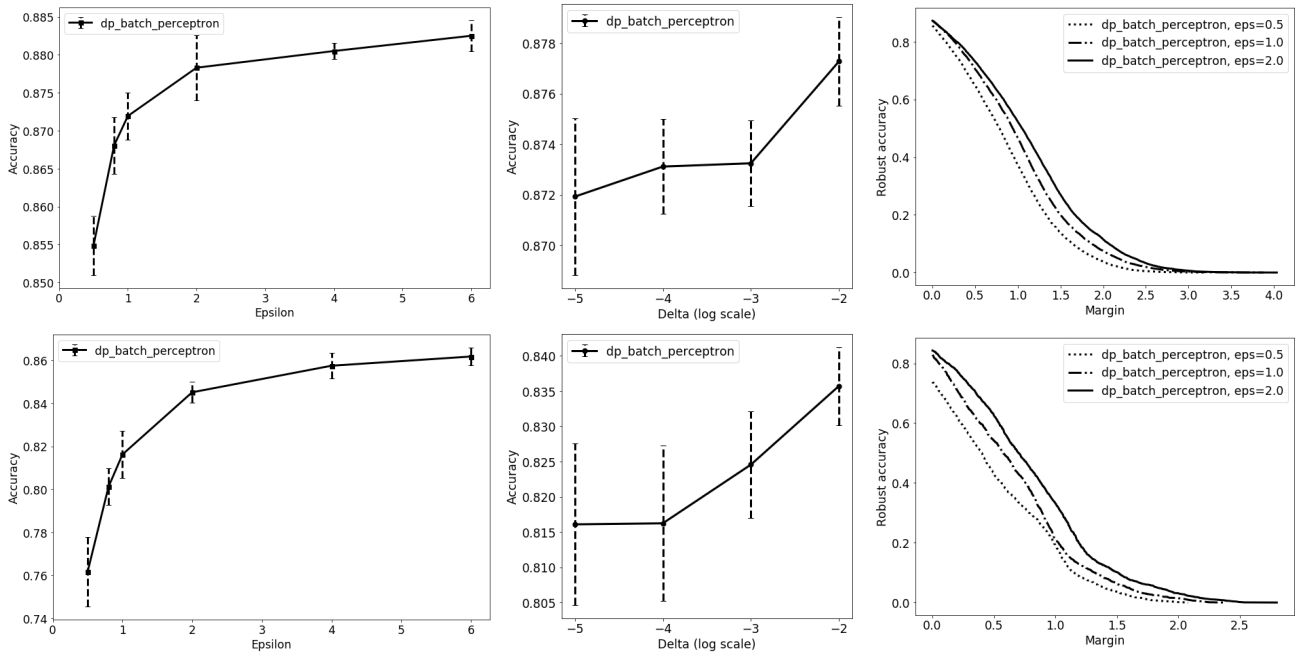
Adversarial Robustness Evaluation. We compare the robustness of our models against those of neural networks trained with DP-SGD. For the latter, we follow the architecture found in the official TensorFlow Privacy tutorial, which consists of two convolutional layers, each followed by a MaxPool operation, and a dense layer that outputs predicted logits. The network is then trained with batch size 250, learning rate 0.15, L_2 clipping-norm 1.0, and for 60 epochs. This configuration yields competitive performance on the MNIST dataset.

To evaluate the robustness of the models, we calculate the robust risk on the test dataset for varying values of the perturbation norm (i.e., margin) γ. Following common practice in the field, we plot the *robust accuracy* on the test data, which is defined as one minus the robust risk (i.e., $1 - \mathcal{R}_\gamma(h, \mathcal{D})$ where $\mathcal{D}$ is the test dataset distribution), instead of the robust risk itself. Similarly, our plots use the *unnormalized* margin, meaning that the images are not re-scaled to have L_2 norm equal to 1 before prediction. Note that each of their pixel values is still scaled (i.e., divided by 255 if necessary) to have values in the range [0, 1].

In the case of DP Batch Perceptron, it is known (see, e.g., Hein and Andriushchenko, 2017) that an example $(\mathbf{x}, y)$ cannot be perturbed (using $\mathbb{P}_\gamma$) to an incorrect label if $\gamma < \min_{y' \neq y} \frac{\langle \mathbf{w}^{(y)}, \mathbf{x} \rangle - \langle \mathbf{w}^{(y')}, \mathbf{x} \rangle}{\|\mathbf{w}^{(y)} - \mathbf{w}^{(y')}\|}$. This formula allows us to exactly calculate the robust risk of our linear classifiers. We stress that this is a *provable* robustness guarantee, i.e., it holds against *all* adversarial attacks with perturbation norm (at most) γ.

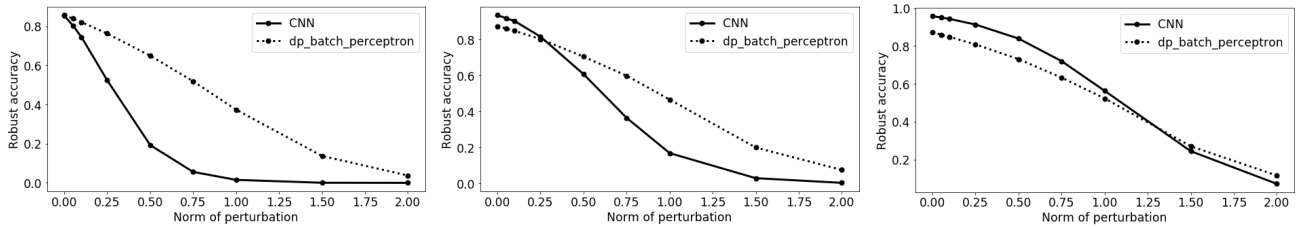
We demonstrate the effect of the change in the required privacy level on the robust risk of our linear classifiers in the right most column of Figure 1. The x-axis of the plots represents the parameter γ and the y-axis represents the γ-robust accuracy on the test dataset.

In contrast to linear models, there is no efficiently-computable formula to calculate robust risk for general neural networks. In this case, we use a variant of a popular adversarial robustness attack (outlined below) to estimate the robust risk of DP-SGD-trained neural



(a) Accuracy as ϵ varies (b) Accuracy as δ varies (c) Robust accuracy as ϵ varies

Figure 1: Performance of DP Batch Perceptron halfspace classifiers on the MNIST (top row) and USPS (bottom row) datasets.



(a) $\epsilon = 0.5$ (b) $\epsilon = 1$ (c) $\epsilon = 2$

Figure 2: Robustness accuracy comparison between DP-SGD-trained Convolutional neural networks and DP Batch Perceptron halfspace classifiers on MNIST dataset for a fixed privacy budget. In all three plots, $\delta = 10^{-5}$ but ϵ varies from 0.5, 1, and 2.

networks. Unlike the linear classifier case, this method only gives a *lower bound* on the robust risk, meaning that more sophisticated attacks might result in even more incorrect classifications.

We now briefly summarize the attack we use against DP-SGD-trained neural networks; (a version of) this method was already presented in (Szegedy et al., 2014). Let M denote a trained model; recall that the last layer of our model consists of 10 outputs corresponding to each class and to predict an image we take y^* with the maximum output. Given a sample $(\mathbf{x}, y)$, we would like to determine whether there exists a perturbation $\Delta \in \mathbb{R}^d$ with $\|\Delta\| \leq \gamma$ such that M predict $\mathbf{x} + \Delta$ to be some other class $y' \neq y$. Instead of solving this (intractable) problem directly, the attack considers a modified objective of

$$\min_{\|\Delta\| \leq \gamma} \ell(M(\mathbf{x} + \Delta), y')$$

where ℓ is some loss function. This optimization problem is then solved using Projected Gradient Descent (PGD). We use the cross entropy loss in our attack.

Comparisons of the robust accuracy of models trained via DP Batch Perceptron and those trained via DP-SGD are shown in Figure 2 for $\delta = 10^{-5}$ and $\epsilon = 0.5, 1, 2$. In the case of $\epsilon = 0.5$, while both classifiers have similar test accuracies (without any perturbation, $\gamma = 0$), as γ increases, the robust accuracy rapidly degrades for the DP-SGD-trained neural network compared to that of the DP Batch Perceptron model. This overall trend persists for $\epsilon = 1, 2$; in both cases, the

neural networks start off with noticeably larger test accuracy when $\gamma = 0$ but are eventually surpassed by halfspace classifiers as γ increases.

5 Other Related Work

Learning Halfspaces with Margin. L_2 robustness is a classical setting closely related to the notion of margin (e.g., Rosenblatt, 1958; Novikoff, 1963). (See the supplementary material for formal definitions.) Known margin-based learning methods include the classic perceptron algorithm (Rosenblatt, 1958; Novikoff, 1963) and its many generalizations and variants (e.g., Duda and Hart, 1973; Collobert and Bengio, 2004; Freund and Schapire, 1999; Gentile and Littlestone, 1999; Li and Long, 2002; Gentile, 2001), as well as Support Vector Machines (SVMs) (Boser et al., 1992; Cortes and Vapnik, 1995). A number of works also explore a closely related *agnostic* setting, where the distribution $\mathcal{D}$ is not guaranteed to be realizable with a margin (e.g., Ben-David and Simon, 2000; Shalev-Shwartz et al., 2010; Long and Servedio, 2011; Birnbaum and Shalev-Shwartz, 2012; Diakonikolas et al., 2019, 2020).

Generalization aspects of margin-based learning of halfspaces is also a widely studied topic (e.g., Bartlett, 1998; Zhang, 2002; Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2002; McAllester, 2003; Kakade et al., 2008), and it is known that the sample complexity of robust learning of halfspaces is $O(1/(\alpha\gamma)^2)$ (Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2002).

To the best of our knowledge, the first work that combines the study of learning halfspaces with margin and DP is (Nguyen et al., 2020); their results are represented in Table 1. Recently, Ghazi et al. (2020) gave alternative proofs for some results of Nguyen et al. (2020) via reductions to clustering problems, but these do not provide any improved sample complexity or running time.

Adversarially Robust Learning. There has been a rapidly growing literature on adversarial robustness. Some of these works have presented evidence that training robust classifiers might be harder than non-robust ones (e.g., Awasthi et al., 2019; Bubeck et al., 2018, 2019; Degwekar et al., 2019). Other works aim to demonstrate the accuracy cost of robustness (e.g., Tsipras et al., 2019; Raghunathan et al., 2019). Another line of work seeks to determine the right quantity that governs adversarial generalization (e.g., Schmidt et al., 2018; Montasser et al., 2019; Khim and Loh, 2018; Yin et al., 2019; Awasthi et al., 2020).

Differentially Private Learning. Private learning has been a popular topic since the early days of differential privacy (e.g., Kasiviswanathan et al., 2008)). Apart from the work of Nguyen et al. (2020) on privately learning halfspaces with a margin, a line of work closely related to our setting is the study of the sample complexity of learning threshold functions (Beimel et al., 2016; Feldman and Xiao, 2014; Bun et al., 2015; Alon et al., 2019; Kaplan et al., 2020a) and halfspaces (Beimel et al., 2019; Kaplan et al., 2020b,c). These works study the setting where the unit ball $\mathbb{B}^d$ is discretized so that the domain $\mathcal{X}$ is $X^d \cap \mathbb{B}^d$ (i.e., each coordinate is an element of X). Interestingly, it has been shown that when X is infinite, halfspaces become unlearnable, i.e., the sample complexity becomes unbounded (Alon et al., 2019). On the other hand, an $(\epsilon, o(1/n))$-DP learner with sample complexity $\tilde{O}\left(\frac{d^{2.5}}{\alpha\epsilon}\right) \cdot 2^{O(\log^* |X|)}$ exists (Kaplan et al., 2020b).

While the above setting is not directly comparable to ours, it is possible to reduce between the margin setting and the discretized setting, albeit with some loss. For example, we may use the grid discretization with $|X| = 0.01\gamma/\sqrt{d}$, to obtain a $(\gamma, 0)$-learner with sample complexity $\tilde{O}\left(\frac{d^{2.5}}{\alpha\epsilon}\right) \cdot 2^{O(\log^*(1/\gamma))}$. This is better than the (straightforward) bound of $O(d \cdot \log(1/\gamma))$ obtained by applying the exponential mechanism (McSherry and Talwar, 2007) when γ is very small (e.g., $\gamma \leq 2^{-d^{1.6}}$). It remains an interesting open problem to close such a gap for very small values of γ.

Several works (e.g., Rubinstein et al., 2012; Chaudhuri et al., 2011) have studied differentially private SVMs. However, to the best of our knowledge, there is no straightforward way to translate their theoretical results to those shown in our paper as the objectives in the two settings are different.

6 Conclusions and Future Directions

In this work, we prove new trade-offs, measured in terms of sample complexity, between privacy and robustness—two crucial properties within the domain of AI ethics and safety—for the classic task of halfspace learning. Our theoretical results demonstrate that DP *and* adversarially robust learning requires a larger number of samples than either DP *or* adversarially robust learning alone. We then propose a learning algorithm that meets both criteria, and test it on two multi-class classification datasets. We also provide empirical evidence that despite having a slight advantage in terms of test accuracy on the main task, standard neural networks trained with DP-SGD are not as robust as those trained with our algorithm.

We conclude with a few future research directions. First, it would be interesting to close the gap for the sample complexity of *improper* approximate-DP $(\gamma, 0.9\gamma)$-robust learners; for $d \gg 1/\gamma^2$, the upper bound is $O(\sqrt{d}/\gamma)$ (Theorem 6) but the lower bound is only $\Omega(1/\gamma^2)$ (Theorem 4). This is the only case where there is still a super-polylogarithmic gap for $d \gg 1/\gamma^2$.

Another technical open question is to improve the lower bounds in Theorem 3 to $\Omega(\min\{\sqrt{d}/\gamma, d\})/\epsilon$. Currently, we are missing the ϵ term because we invoke a lower bound from Steinke and Ullman (2017) (Theorem 11), which was specifically proved only for $\epsilon = 1$.

Furthermore, it would be natural to extend our study to L_p perturbations for $p \neq 2$. An especially noteworthy case is when $p = \infty$, which is a well-studied setting in the adversarial robustness literature.

Finally, it would be very interesting to provide a theoretical understanding of private and robust learning beyond halfspaces, to accommodate complex algorithms (e.g., deep neural networks) that are better suited for more challenging tasks.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, pages 308–318, 2016.

John M Abowd. The US Census Bureau adopts differential privacy. In *KDD*, pages 2867–2867, 2018.

Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. In *FOCS*, pages 544–553, 1990.

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite Littlestone dimension. In *STOC*, pages 852–860, 2019.

Apple Differential Privacy Team. Learning with privacy at scale. *Apple Machine Learning Journal*, 2017.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pages 274–283, 2018.

Pranjal Awasthi, Abhratanu Dutta, and Aravindan Vijayaraghavan. On robustness to adversarial examples and polynomial optimization. In *NeurIPS*, pages 13760–13770, 2019.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. Adversarial learning guarantees for linear hypotheses and neural networks. In *ICML*, pages 431–441, 2020.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *NeurIPS*, pages 6280–6290, 2018.

Peter L. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. Inf. Theory*, 44(2):525–536, 1998.

Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463–482, 2002.

Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473, 2014.

Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. In *NeurIPS*, pages 11282–11291, 2019.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. *Theory Comput.*, 12(1):1–61, 2016.

Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. In *COLT*, pages 269–282, 2019.

Shai Ben-David and Hans Ulrich Simon. Efficient learning of linear perceptrons. In *NIPS*, pages 189–195, 2000.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *ECML/PKDD*, pages 387–402, 2013.

Aharon Birnbaum and Shai Shalev-Shwartz. Learning halfspaces with the zero-one loss: Time-accuracy tradeoffs. In *NIPS*, pages 935–943, 2012.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *PODS*, pages 128–138, 2005.

Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152, 1992.

Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *ICML*, pages 831–840, 2018.

Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *ICML*, pages 831–840, 2019.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *FOCS*, pages 634–649, 2015.

Mark Bun, Jonathan Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. *SIAM J. Comput.*, 47(5):1888–1938, 2018.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *IEEE SPW*, pages 1–7, 2018.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *JMLR*, 12(3):1069–1109, 2011.

Ronan Collobert and Samy Bengio. Links between perceptrons, MLPs and SVMs. In *ICML*, 2004.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *KDD*, pages 99–108, 2004.

Akshay Degwekar, Preetum Nakkiran, and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. In *COLT*, pages 994–1028, 2019.

Ilias Diakonikolas, Daniel Kane, and Pasin Manurangsi. Nearly tight bounds for robust proper learning of halfspaces with a margin. In *NeurIPS*, pages 10473–10484, 2019.

Ilias Diakonikolas, Daniel M. Kane, and Pasin Manurangsi. The complexity of adversarially robust proper learning of halfspaces with agnostic noise. In *NeurIPS*, 2020.

Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *NIPS*, pages 3571–3580, 2017.

Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006b.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.

Cynthia Dwork, Adam D. Smith, Thomas Steinke, Jonathan Ullman, and Salil P. Vadhan. Robust traceability from trace amounts. In *FOCS*, pages 650–669, 2015.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *CCS*, pages 1054–1067, 2014.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *COLT*, pages 1000–1019, 2014.

Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *STOC*, pages 439–449, 2020.

Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, 1999.

Claudio Gentile. A new approximate maximal margin classification algorithm. *JMLR*, 2:213–242, 2001.

Claudio Gentile and Nick Littlestone. The robustness of the p-norm algorithms. In *COLT*, pages 1–11, 1999.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. In *NeurIPS*, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv Preprint:1810.12715*, 2018.

Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy Mann, and Pushmeet Kohli. An alternative surrogate loss for PGD-based adversarial testing. *arXiv Preprint: 1910.09338*, 2019.

Andy Greenberg. Apple's "differential privacy" is about collecting your data – but not your data. *Wired, June*, 13, 2016.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *STOC*, pages 705–714, 2010.

Jamie Hayes. Provable trade-offs between private & robust machine learning. *arXiv Preprint: 2006.04622*, 2020.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NeurIPS*, pages 2266–2276, 2017.

Jonathan J. Hull. A database for handwritten text recognition research. *IEEE PAMI*, 16(5):550–554, 1994.

Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, pages 793–800, 2008.

Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. In *COLT*, pages 2263–2285, 2020a.

Haim Kaplan, Yishay Mansour, Uri Stemmer, and Eliad Tsfadia. Private learning of halfspaces: Simplifying the construction and reducing the sample complexity. In *NeurIPS*, 2020b.

Haim Kaplan, Micha Sharir, and Uri Stemmer. How to Find a Point in the Convex Hull Privately. In *SoCG*, pages 52:1–52:15, 2020c.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? In *FOCS*, pages 531–540, 2008.

Justin Khim and Po-Ling Loh. Adversarial risk bounds via function transformation. *arXiv Preprint: 1810.09519*, 2018.

Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

Zico Kolter and Aleksander Madry. Adversarial robustness: Theory and practice. *Tutorial at NeurIPS*, 2018.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database, 2010. `http://yann.lecun.com/exdb/mnist`.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE S&P*, pages 656–672, 2019.

Yi Li and Philip M. Long. The relaxed online maximum margin algorithm. *Mach. Learn.*, 46(1-3):361–387, 2002.

Philip M. Long and Rocco A. Servedio. Learning large-margin halfspaces with more malicious noise. In *NIPS*, pages 91–99, 2011.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

David A. McAllester. Simplified PAC-Bayesian margin bounds. In *COLT*, pages 203–215, 2003.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

Ilya Mironov. Rényi differential privacy. In *CSF*, pages 263–275, 2017.

Omar Montasser, Steve Hanneke, and Nathan Srebro. VC classes are adversarially robustly learnable, but only improperly. In *COLT*, pages 2512–2530, 2019.

Huy Le Nguyen, Jonathan Ullman, and Lydia Zakynthinou. Efficient private algorithms for learning large-margin halfspaces. In *ALT*, pages 704–724, 2020.

Albert B Novikoff. On convergence proofs for perceptrons. Technical report, Stanford Research Institute, Menlo Park, CA, 1963.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv Preprint: 1605.07277*, 2016.

NhatHai Phan, Minh Vu, Yang Liu, Ruoming Jin, Dejing Dou, Xintao Wu, and My T Thai. Heterogeneous Gaussian mechanism: Preserving differential privacy in deep learning with provable robustness. In *IJCAI*, pages 4753–4759, 2019.

NhatHai Phan, My T Thai, Han Hu, Ruoming Jin, Tong Sun, and Dejing Dou. Scalable differential privacy with certified robustness in adversarial learning. In *ICML*, pages 7683–7694, 2020.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Adversarial training can hurt generalization. *arXiv Preprint: 1906.06032*, 2019.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

Benjamin IP Rubinstein, Peter L Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *J. Priv. Confidentiality*, 4(1), 2012.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*, pages 5014–5026, 2018.

Bernhard Schölkopf, Kah Kay Sung, Christopher J. C. Burges, Federico Girosi, Partha Niyogi, Tomaso A. Poggio, and Vladimir Vapnik. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Trans. Signal Process.*, 45 (11):2758–2765, 1997.

Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on MNIST. In *ICLR*, 2019.

Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the zero-one loss. In *COLT*, pages 441–450, 2010.

Stephen Shankland. How Google tricks itself to protect Chrome user privacy. *CNET, October*, 2014.

Yash Sharma and Pin-Yu Chen. Attacking the Madry defense model with l_1-based adversarial examples. *arXiv Preprint: 1710.10733*, 2017.

Liwei Song, Reza Shokri, and Prateek Mittal. Membership inference attacks against adversarially robust deep learning models. In *IEEE SPW*, pages 50–56, 2019a.

Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *CCS*, pages 241–257, 2019b.

Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *J. Priv. Confidentiality*, 7(2), 2016.

Thomas Steinke and Jonathan Ullman. Tight lower bounds for differentially private selection. In *FOCS*, pages 552–563, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *ICML*, pages 7085–7094, 2019.

Tong Zhang. Covering number bounds of certain regularized linear function classes. *JMLR*, 2:527–550, 2002.