Adversarial Robustness with Semi-Infinite Constrained Learning

Alexander Robey⇤
University of Pennsylvania
arobey1@seas.upenn.edu

Luiz F. O. Chamon⇤
University of California, Berkeley
lfochamon@berkeley.edu

George J. Pappas
University of Pennsylvania
pappasg@seas.upenn.edu

Hamed Hassani
University of Pennsylvania
hassani@seas.upenn.edu

Alejandro Ribeiro
University of Pennsylvania
aribeiro@seas.upenn.edu

Abstract

Despite strong performance in numerous applications, the fragility of deep learning to input perturbations has raised serious questions about its use in safety-critical domains. While adversarial training can mitigate this issue in practice, state-of-the-art methods are increasingly application-dependent, heuristic in nature, and suffer from fundamental trade-offs between nominal performance and robustness. Moreover, the problem of finding worst-case perturbations is non-convex and underparameterized, both of which engender a non-favorable optimization landscape. Thus, there is a gap between the theory and practice of adversarial training, particularly with respect to when and why adversarial training works. In this paper, we take a constrained learning approach to address these questions and to provide a theoretical foundation for robust learning. In particular, we leverage semi-infinite optimization and non-convex duality theory to show that adversarial training is equivalent to a statistical problem over perturbation distributions, which we characterize completely. Notably, we show that a myriad of previous robust training techniques can be recovered for particular, sub-optimal choices of these distributions. Using these insights, we then propose a hybrid Langevin Monte Carlo approach of which several common algorithms (e.g., PGD) are special cases. Finally, we show that our approach can mitigate the trade-off between nominal and robust performance, yielding state-of-the-art results on MNIST and CIFAR-10. Our code is available at: https://github.com/arobey1/advbench.

1 Introduction

Learning is at the core of many modern information systems, with wide-ranging applications in clinical research [1–4], smart grids [5–7], and robotics [8–10]. However, it has become clear that learning-based solutions suffer from a critical lack of robustness [11–17], leading to models that are vulnerable to malicious tampering and unsafe behavior [18–22]. While robustness has been studied in statistics for decades [23–25], this issue has been exacerbated by the opacity, scale, and non-convexity of modern learning models, such as convolutional neural network (CNNs). Indeed, the pernicious nature of these vulnerabilities has led to a rapidly-growing interest in improving the so-called adversarial robustness of modern ML models. To this end, a great deal of empirical evidence has shown adversarial training to be the most effective way to obtain robust classifiers, wherein models are trained on perturbed samples rather than directly on clean data [26–31]. While this approach is now ubiquitous in practice, adversarial training faces two fundamental challenges.

⇤ Alexander Robey and Luiz F. O. Chamon contributed equally to this work.
At a high level, our goal is to learn a classifier which can correctly predict the label $y$ for a given instance $x$. To this end, we let $H$ be a hypothesis class containing functions $f_\theta : \mathbb{R}^d \rightarrow \mathcal{S}^K$ parameterized by vectors $\theta \in \Theta \subset \mathbb{R}^p$, where we assume that the parameter space $\Theta$ is compact and by $\mathcal{S}^K$ we denote the $(K-1)$-simplex. We also assume that $f_\theta(x)$ is differentiable with respect to $\theta$ and $x$. To make a prediction $\hat{y} \in \mathcal{Y}$, we assume that the simplex $\mathcal{S}^K$ is mapped to the set of classes $\mathcal{Y}$ via $\hat{y} \in \arg\max_{y \in \mathcal{Y}} [f_\theta(x)]_k$ with ties broken arbitrarily. In this way, we can think of the $k$-th output of the classifier as representing the probability that $y = k$. Given this notation, the statistical problem of learning a classifier that accurately predicts the label $y$ of a given instance $x$.

Firstly, it is well-known that obtaining worst-case, adversarial perturbations of data is challenging in the context of deep neural networks (DNNs) [52,53]. While gradient-based methods have been shown to be empirically effective at finding perturbations that lead to misclassification, there are no guarantees that these perturbations are truly worst-case due to the non-convexity of most commonly-used ML function classes [34]. Moreover, whereas optimizing the parameters of a DNNs is typically an overparameterized problem, finding worst-case perturbations is severely underparametrized and therefore does not enjoy the benign optimization landscape of standard training [55–59]. For this reason, state-of-the-art adversarial attacks increasingly rely on heuristics such as random initializations, multiple restarts, pruning, and other ad hoc training procedures [40–49].

The second challenge faced by adversarial training is that it engenders a fundamental trade-off between robustness and nominal performance [50–52]. In practice, penalty-based methods that incorporate clean data into the training objective are often used to overcome this issue [53,54]. However, while empirically successful, these methods cannot typically guarantee nominal or adversarial performance outside of the training samples. Indeed, classical learning theory [57,58] provides generalization bounds only for the aggregated objective and not each individual penalty term. Additionally, the choice of the penalty parameter is not straightforward and depends on the underlying learning task, making it difficult to transfer across applications and highly dependent on domain expert knowledge.

We show that adversarial training is equivalent to a stochastic optimization problem over a specific, non-atomic distribution, which we characterize using recent non-convex duality results [59,60]. Further, we show that a myriad of previous adversarial attacks reduce to particular, sub-optimal choices of this distribution.

We propose an algorithm to solve this problem based on stochastic optimization and Markov chain Monte Carlo. Gradient-based methods can be seen as limiting cases of this procedure.

We show that our algorithm outperforms state-of-the-art baselines on MNIST and CIFAR-10. In particular, our approach yields a ResNet-18 classifiers which simultaneously achieves greater than 50% adversarial accuracy and greater than 85% clean accuracy on CIFAR-10, which represents a significant improvement over the previous state-of-the-art.

We provide generalization guarantees for the empirical version of this algorithm, showing how to effectively limit the nominal performance degradation of robust classifiers.

## 2 Problem formulation

Throughout this paper, we consider a standard classification setting in which the data is distributed according to an unknown joint distribution $D$ over instance-label pairs $(x, y)$. In this setting, the instances $x \in \mathcal{X}$ are assumed to be supported on a compact subset of $\mathbb{R}^d$, and each label $y \in \mathcal{Y} := \{1, \ldots, K\}$ denotes the class of a given instance $x$. By $(\Omega, \mathcal{B})$ we denote the underlying measurable space for this setting, where $\Omega = \mathcal{X} \times \mathcal{Y}$ and $\mathcal{B}$ denotes its Borel $\sigma$-algebra. Furthermore, we assume that the joint distribution $D$ admits a density $p(x, y)$ defined over the sets of $\mathcal{B}$.

At a high level, our goal is to learn a classifier which can correctly predict the label $y$ of a corresponding instance $x$. To this end, we let $H$ be a hypothesis class containing functions $f_\theta : \mathbb{R}^d \rightarrow \mathcal{S}^K$ parameterized by vectors $\theta \in \Theta \subset \mathbb{R}^p$, where we assume that the parameter space $\Theta$ is compact and by $\mathcal{S}^K$ we denote the $(K-1)$-simplex. We also assume that $f_\theta(x)$ is differentiable with respect to $\theta$ and $x$. To make a prediction $\hat{y} \in \mathcal{Y}$, we assume that the simplex $\mathcal{S}^K$ is mapped to the set of classes $\mathcal{Y}$ via $\hat{y} \in \arg\max_{y \in \mathcal{Y}} [f_\theta(x)]_k$ with ties broken arbitrarily. In this way, we can think of the $k$-th output of the classifier as representing the probability that $y = k$. Given this notation, the statistical problem of learning a classifier that accurately predicts the label $y$ of a given instance $x$.

\[\text{Note that the classes of support vector machines, logistic classifiers, and convolutional neural networks (CNNs) with softmax outputs can all be described by this formalism.}\]
drawn randomly from \( \mathcal{D} \) can be formulated as follows:

\[
\min_{\theta \in \Theta} \mathbb{E}_{(x, y) \sim \mathcal{D}} \left[ \ell(f_{\theta}(x), y) \right].
\]  

(P-NOM)

Here \( \ell \) is a \([0, B]\)-valued loss function and \( \ell(\cdot, y) \) is \( M \)-Lipschitz continuous for all \( y \in \mathcal{Y} \). We assume that \((x, y) \mapsto \ell(f_{\theta}(x), y)\) is integrable so that the objective in (P-NOM) is well-defined; we further assume that this map is an element of the Lebesgue space \( L^p(\Omega, \mathcal{B}, p) \) for some fixed \( p \in (1, \infty) \).

Formulating the robust training objective. For common choices of the hypothesis class \( \mathcal{H} \), including DNNs, classifiers obtained by solving (P-NOM) are known to be sensitive to small, norm-bounded input perturbations \([61]\). In other words, it is often straightforward to find a relatively small perturbations \( \delta \) such that the classifier correctly predicts the label \( y \) of \( x \), but misclassifies the perturbed sample \( x + \delta \). This has led to increased interest in the robust analog of (P-NOM), namely,

\[
P_R^* = \min_{\theta \in \Theta} \mathbb{E}_{(x, y) \sim \mathcal{D}} \max_{\delta \in \Delta} \ell(f_{\theta}(x + \delta), y).
\]  

(P-RO)

In this optimization program, the set \( \Delta \subset \mathbb{R}^d \) denotes the set of valid perturbations. Typically, \( \Delta \) is chosen to be a ball with respect to a given metric on Euclidean space, i.e., \( \Delta = \{ \delta \in \mathbb{R}^d : \|\delta\| \leq \epsilon \} \). However, in this paper we make no particular assumption on the specific form of \( \Delta \). In particular, our results apply to arbitrary perturbation sets, such as those used in \([62, 66]\).

Analyzing conditions under which (P-RO) can be (probably approximately) solved from data remains an active area of research. While bounds on the Rademacher complexity \([67, 68]\) and VC dimension \([67, 71]\) of the robust loss

\[
\ell_{\text{adv}}(f_{\theta}(x), y) = \max_{\delta \in \Delta} \ell(f_{\theta}(x + \delta), y)
\]  

have been derived for an array of losses \( \ell \) and hypothesis classes \( \mathcal{H} \), there are still open questions on the effectiveness and sample complexity of adversarial learning \([68]\). Moreover, because in general the map \( \delta \mapsto \ell(f_{\theta}(x + \delta), y) \) is non-concave, evaluating the maximum in (1) is not straightforward. To this end, the most empirically effective strategy for approximating the robust loss is to leverage the differentiability of modern ML models with respect to their inputs. More specifically, by computing gradients of such models, one can approximate the value of (1) using projected gradient ascent. For instance, in \([72, 73]\) an perturbation \( \delta \) is computed for a fixed parameter \( \theta \) and data point \((x, y)\) by repeatedly applying

\[
\delta \leftarrow \Pi_{\Delta} \left[ \delta + \eta \text{sign} \left[ \nabla_{\delta} \ell(f_{\theta}(x + \delta), y) \right] \right],
\]  

where \( \Pi_{\Delta} \) denotes the projection onto \( \Delta \) and \( \eta > 0 \) is a fixed step size. This idea is at the heart of adversarial training, in which (2) is iteratively applied to approximately evaluate the robust risk in the objective of (P-RO); the parameters \( \theta \) can then be optimized with respect to this robust risk.

Common pitfalls for adversarial training. Their empirical success notwithstanding, gradient-based approaches to adversarial training are not without drawbacks. One pitfall is the fact that gradient-based algorithms are not guaranteed to provide optimal (or even near-optimal) perturbations, since \( x \mapsto \ell(f_{\theta}(x), y) \) is typically not a concave function. Because of this, heuristics \([72, 74]\) are often needed to improve the solutions obtained from (2). Furthermore, adversarial training often degrades the performance of the model on clean data \([52, 55, 76]\). In practice, penalty-based approaches are used to empirically overcome this issue \([53] \), although results are not guaranteed to generalize outside of the training sample. Indeed, classical learning theory guarantees generalization in terms of the aggregated objective and not in terms of the robustness requirements it may describe \([57, 58, 60]\).

In the remainder of this paper, we address these pitfalls by leveraging semi-infinite constrained learning theory. To do so, we explicitly formulate the problem of finding the most robust classifier among those that have good nominal performance. Next, we show that (P-RO) is equivalent to a stochastic optimization problem that can be related to numerous adversarial training methods (Section 3). We then provide generalization guarantees for the constrained robust learning problem when solve using empirical (unconstrained) risk minimization (Section 3). Finally, we derive an algorithm based on a Langevin MCMC sampler of which (2) is a particular case (Section 5).
3 Dual robust learning

In the previous section, we argued that while empirically successful, adversarial training is not without shortcomings. In this section, we develop the theoretical foundations needed to tackle the two challenges of (P-RO): (a) finding worst-case perturbations, i.e., evaluating the robust loss defined in (PII) and (b) mitigating the trade-off between robustness and nominal performance. To address these challenges, we first propose the following constrained optimization problem which explicitly captures the trade-off between robustness and nominal performance:

\[
P^* = \min_{\theta \in \Theta, t \in L^p} \mathbb{E}_{(x,y) \sim D} \left[ \max_{\delta \in \Delta} \ell(f_{\theta}(x + \delta), y) \right] \tag{P-CON}
\]

where \( \rho \geq 0 \) is a desired nominal performance level. At a high level, (P-CON) seeks the most robust classifier \( f_{\theta}(\cdot) \) among those classifiers that have strong nominal performance. In this way, (P-CON) is directly designed to address the trade-off between robustness and accuracy, and as such (P-CON) will be the central object of study in this paper. We note that at face value, the statistical constraint in (P-CON) is challenging to enforce in practice, especially given the well-known difficulty in solving the unconstrained analog (P-RO). Following [63] [74], our approach is to use duality to obtain solutions for (P-CON) that generalize with respect to both adversarial and nominal performance.

**Computing worst-case perturbations.** Before tackling the constrained problem (P-CON), we begin by consider its unconstrained version, namely, (P-RO). We start by rewriting (P-RO) using an epigraph formulation of the maximum function to obtain the following semi-infinite program:

\[
P^*_R = \min_{\theta \in \Theta, t \in L^p} \mathbb{E}_{(x,y) \sim D} \left[ t(x, y) \right] \tag{PI}
\]

subject to \( \ell(f_{\theta}(x + \delta), y) \leq t(x, y), \) for almost every \((x, \delta, y) \in \mathcal{X} \times \Delta \times \mathcal{Y}. \)

Note that (PI) is indeed equivalent to (P-RO) since

\[
\max_{\delta \in \Delta} \ell(f_{\theta}(x + \delta), y) \leq t(x, y) \iff \ell(f_{\theta}(x + \delta), y) \leq t(x, y) \text{ for all } \delta \in \Delta. \tag{3}
\]

While at first it may seem that we have made (P-CON) more challenging to solve by transforming an unconstrained problem into an infinitely-constrained problem, notice that (PI) is no longer a composite minimax problem. Furthermore, it is *linear* in \( t \), indicating that (PI) should be amenable to approaches based on Lagrangian duality. Indeed, the following proposition shows that (PI) can be used to obtain a statistical counterpart of (P-RO).

**Proposition 3.1.** If \((x, y) \mapsto \ell(f_{\theta}(x), y) \in L^p \) for \( p \in (1, \infty) \), then (P-RO) can be written as

\[
P^*_R = \min_{\theta \in \Theta} p(\theta), \tag{PII}
\]

for the primal function

\[
p(\theta) \triangleq \max_{\lambda \in \mathcal{P}_q} \mathbb{E}_{(x,y) \sim D} \left[ \mathbb{E}_{\delta \sim \lambda(\delta|x,y)} \left[ \ell(f_{\theta}(x + \delta), y) \right] \right], \tag{4}
\]

where \( \mathcal{P}_q \), with \( \frac{1}{p} + \frac{1}{q} = 1 \), is the subspace of \( L^q \) containing almost everywhere non-negative functions such that \( p(x, y) = 0 \Rightarrow \lambda(\delta \mid x, y) = 0 \) and \( \int \lambda(\delta \mid x, y) d\delta = 1 \) for almost every \((x, y) \in \mathcal{X} \times \mathcal{Y}. \)

The proof is provided in Appendix B. Informally, Proposition 3.1 shows that the robust learning problem in (P-RO) can be recast as a problem of optimizing over a set of probability distributions \( \mathcal{P}_q \) taking support over \( \Delta. \) This establishes an equivalence between the traditional robust learning problem (P-RO), where the maximum is taken over perturbations \( \delta \in \Delta \) of the input, and its stochastic version (PII), where the maximum is taken over a conditional distribution over perturbations \( \delta \sim \lambda(\delta \mid x, y). \) Notably, a variety of training formulations can be seen as special cases of (PII). In fact, for particular sub-optimal choices of this distribution, paradigms such as random data augmentation and distributionally robust optimization can be recovered (see Appendix A for details).

As we remarked in Section 2, for many modern function classes, the task of evaluating the adversarial loss (1) is a nonconcave optimization problem, which is challenging to solve in general. Thus,
Proposition 3.1 can be seen as lifting the nonconcave inner problem \( \max_{\delta \in \Delta} \ell(f_\theta(x), y) \) to the equivalent linear optimization problem in 4 over probability distributions \( \lambda \in \mathcal{P}^q \). This dichotomy parallels the one that arises in PAC vs. agnostic PAC learning. Indeed, while the former seeks a deterministic map \( (\theta, x, y) \mapsto \delta \), the latter considers instead a distribution of perturbations over \( \delta|x, y \) parametrized by \( \theta \). In fact, since (P-II) is obtained from (P-RO) through semi-infinite duality, the density of this distribution is exactly characterized by the dual variables \( \lambda \).

Note that while (P-II) was obtained using Lagrangian duality, it can also be seen as a linear lifting of the maximization in 4. From this perspective, while recovering 4 would require \( \lambda \) to be atomic, Proposition 3.1 shows that this is not necessary as long as \( \ell(f_\theta(x), y) \) is an element of \( L^p \). That is, because \( \mathcal{P}^q \) does not contain any Dirac distributions, the optimal distribution \( \lambda^* \) for the maximization problem in 4 is non-atomic. Hence, Proposition 3.1 does not show that (P-II) finds worst-case perturbations that achieve the maximum in the objective of (P-RO). It does, however, show that finding worst-case perturbation is not essential to find a solution of (P-RO).

Exact solutions for the maximization in (P-II). While (P-II) provides a new constrained formulation for (P-RO), the objectives of both (P-II) and (P-RO) still involve the solution of a non-trivial maximization. However, whereas the maximization problem in (P-RO) is a finite-dimensional problem which is nonconcave for most modern function classes, the maximization in (P-II) is a linear, variational problem regardless of the function class. We can therefore leverage variational duality theory to obtain a full characterization of the optimal distribution \( \lambda^* \) when \( p = 2 \).

**Proposition 3.2 (Optimal distribution for (P-II)).** Let \( p = 2 \) (and \( q = 2 \)) in Proposition 3.1 and let \( [z]_+ = \max(0, z) \). For each \((x, y) \in \Omega\), there exists constants \( \gamma(x, y) > 0 \) and \( \mu(x, y) \in \mathbb{R} \) s.t.

\[
\lambda^*(\delta | x, y) = \left[ \frac{\ell(f_\theta(x + \delta), y) - \mu(x, y)}{\gamma(x, y)} \right]_+
\]

is a solution of the maximization in 4. In particular, the value of \( \mu(x, y) \) is such that

\[
\int_\Delta [\ell(f_\theta(x + \delta), y) - \mu(x, y)]_+ d\delta = \gamma(x, y) \quad \forall (x, y) \in \mathcal{X} \times \mathcal{Y}.
\]

The proof is provided in Appendix C. This proposition shows that when \((x, y) \mapsto \ell(f_\theta(x), y) \in L^2\), we can obtain a closed-form expression for the distribution \( \lambda^* \) that maximizes the objective of (P-II). Moreover, this distribution is proportional to a truncated version of the loss of the classifier. Note that the assumption that the loss belongs to \( L^2 \) is mild given that the compactness of \( \mathcal{X} \), \( \mathcal{Y} \), and \( \Delta \) imply that \( L^{p_1} \subset L^{p_2} \) for \( p_1 > p_2 \). It is, however, fundamental to obtain the closed-form solution in Proposition 3.2 since it allows 4 to be formulated as a strongly convex constrained problem whose primal solution can be recovered from its dual variables (namely, \( \gamma \) and \( \mu \)). To illustrate this result, we consider two particular suboptimal choices for the constants \( \mu \) and \( \gamma \).

**Special case I: over-smoothed \( \lambda^* \).** Consider the case when \( \gamma(x, y) \) is taken to be the normalizing constant \( \int_\Delta \ell(f_\theta(x + \delta), y) d\delta \) for each \((x, y) \in \Omega\). As the loss function \( \ell \) is non-negative, (4) implies that \( \mu(x, y) = 0 \), and the distribution defined in (5) can be written as

\[
\lambda(\delta | x, y) = \left[ \frac{\ell(f_\theta(x + \delta), y)}{\int_\Delta \ell(f_\theta(x + \delta), y) d\delta} \right]_+,
\]

meaning that \( \lambda(\delta | x, y) \) is exactly proportional to the loss \( \ell(f_\theta(x + \delta), y) \) on a perturbed copy of the data. Thus, for this choice of \( \gamma \) and \( \mu \), the distribution \( \lambda \) in (7) is an over-smoothed version of the optimal distribution \( \lambda^* \). In our experiments, we will use this over-smoothed approximation of \( \lambda^* \) to derive an MCMC-style sampler, which yields state-of-the-art performance on standard benchmarks.

**Special case II: under-smoothed \( \lambda^* \).** It is also of interest to consider the case in which \( \gamma \) approaches zero. In the proof of Proposition 3.2, we show that the value of \( \mu \) is fully determined by \( \gamma \) and that \( \gamma \) is directly related to the smoothness of the optimal distribution; in fact, \( \gamma \) is equivalent to a bound on the \( L^2 \) norm of \( \lambda^* \). In this way, as we take \( \gamma \) to zero, we find that \( \mu \) approaches \( \max_{\delta \in \Delta} \ell(f_\theta(x + \delta), y) \), meaning that the distribution is truncated so that mass is only placed on those perturbations \( \delta \) which induce the maximum loss. Thus, in the limit, \( \lambda \) approaches an atomic distribution concentrated

---

1Proposition 3.1 does not account for \( p \in \{1, \infty\} \) for conciseness. Nevertheless, neither of the dual spaces \( L^1 \) or \( L^\infty \) contain Dirac distributions, meaning that for \( p \in \{1, \infty\} \), \( \lambda^* \) would remain non-atomic.
entirely at a perturbation $\delta^*$ that maximizes the loss. Interestingly, this is the same distribution that would be needed to recover the solution to the inner maximization as in (P-RO). This highlights the fact that although recovering the optimal $\delta^*$ in (P-CON) would require $\lambda^*$ to be atomic, the condition that $\gamma > 0$ means that $\lambda^*$ need not be atomic.

These two cases illustrate the fundamental difference between (P-RO) and (P-II). Whereas in (P-RO) we search for worst-case perturbations, in (P-II) we seek a method to sample perturbations $\delta$ from the perturbation distribution $\lambda^*$. Thus, given a method for sampling $\delta \sim \lambda^*(\delta|x, y)$, the max in (P-RO) can be replaced by an expectation, allowing us to consider the following optimization problem:

$$P_R^\star = \min_{\theta \in \Theta} \mathbb{E}_{(x, y) \sim D}[\mathbb{E}[\delta \sim \lambda^*(\delta|x, y)] (\ell(f_\theta(x + \delta), y)]].$$

(PIII)

Notice that crucially this problem is non-composite in the sense that it no longer contains an inner maximization. To this end, in Section 5, we propose a scheme that can be used to sample from a close approximation of $\lambda^*$ toward evaluating the inner expectation in (PIII).

4 Solving the constrained learning problem

So far, we have argued that (P-CON) captures the problem of finding the most robust model with high nominal performance and we have shown that the minimax objective of (P-CON) can be rewritten as a stochastic optimization problem over perturbation distributions. In this section, we address the distinct yet related issue of satisfying the constraint in (P-CON), which is a challenging task in practice given the statistical and potentially non-convex nature of the problem. Further complicating matters is the fact that by assumption we have access to the data distribution $D$ only through samples $(x, y) \sim D$, which means that in practice we cannot evaluate either of the expectations in (P-CON). To overcome these obstacles, given a dataset $\{(x_n, y_n)\}_{n=1}^N$ sampled i.i.d. from $D$, our approach is to use duality to approximate (P-CON) by the following empirical, unconstrained saddle point problem:

$$\hat{D}^\star = \max_{\nu \geq 0} \min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^N \left[ \max_{\delta \in \Delta} \ell(f_\theta(x_n + \delta), y_n) + \nu \left( \ell(f_\theta(x_n), y_n) - \rho \right) \right].$$

(DI)

Conditions under which solutions of (DI) are (probably approximately) near-optimal and near-feasible for (P-CON) were obtained in [60]. As one would expect, these guarantees only hold when the objective and constraint of (P-CON) are learnable individually. As we discussed in Section 2 this is known to hold in a variety of scenarios (e.g., when the Rademacher complexity or VC-dimension is bounded), although obtaining more general results remains an area of active research [67, 71].

In what follows, we formalize these generalization results in our setting, starting with the learning theoretic assumptions we require on the objective and constraint.

Learning theoretic assumptions for (P-CON). We first assume that the parameterization space $\Theta$ is sufficiently expressive (Assumption 4.1) and that there exists parameters $\theta \in \Theta$ that strictly satisfy the nominal performance constraint (Assumption 4.2). We also assume that uniform convergence holds for the objective and constraint (Assumption 4.3).

**Assumption 4.1.** The parametrization $f_\theta$ is rich enough so that for each $\theta_1, \theta_2 \in \Theta$ and $\beta \in (0, 1]$, there exists $\theta \in \Theta$ such that $\sup_{x \in \mathcal{X}} |\beta f_{\theta_1}(x) + (1 - \beta)f_{\theta_2}(x) - f_\theta(x)| \leq \alpha$.

**Assumption 4.2.** There exists $\theta' \in \Theta$ such that $\mathbb{E}_D[\ell(f_{\theta'}(x), y)] < \rho - M\alpha$.

**Assumption 4.3.** There exists $\zeta_R(N), \zeta_N(N) \geq 0$ monotonically decreasing in $N$ such that $\forall \theta \in \Theta$:  

$$\left| \mathbb{E}_{(x, y) \sim D} \left[ \max_{\delta \in \Delta} \ell(f_\theta(x + \delta), y) \right] - \frac{1}{N} \sum_{n=1}^N \max_{\delta \in \Delta} \ell(f_\theta(x_n + \delta), y_n) \right| \leq \zeta_R(N) \ 	ext{w.p.} \ 1 - \delta \quad (8a)$$

$$\left| \mathbb{E}_{(x, y) \sim D} \ell(f_\theta(x), y) - \frac{1}{N} \sum_{n=1}^N \ell(f_\theta(x_n), y_n) \right| \leq \zeta_N(N) \ 	ext{w.p.} \ 1 - \delta \quad (8b)$$

One natural question to ask is whether the bounds in (8a) and (8b) hold in practice. We note that in the non-adversarial bound (8b) has been shown to hold for a wide variety of hypothesis classes, including DNNs [73, 84]. And although these classical results do not imply the robust uniform
Algorithm 1 Semi-Infinite Dual Adversarial Learning (DALE)

\begin{algorithm}
\begin{algorithmic}[1]
\State Initialize $\theta \leftarrow \theta_0$ and $\nu \leftarrow 0$
\Repeat
\For{Batch $\{ (x_i, y_i) \}_{i=1}^m$}
\State $\delta_i \leftarrow 0$, for $i = 1, \ldots, m$
\For{$L$ steps}
\State $U_i \leftarrow \log \left[ \ell_{\text{pen}}(f_\theta(x_i + \delta_i), y_i) \right]$, for $i = 1, \ldots, m$
\State $\delta_i \leftarrow \prod_{i} \left[ \delta_i + \eta \nabla_\delta U_i + \sqrt{2\eta \mathcal{I}} \xi_i \right]$, where $\xi_i \sim \text{Laplace}(0, I)$ and $i = 1, \ldots, m$
\EndFor
\State $\theta \leftarrow \theta - \frac{\eta}{m} \sum_{i=1}^m \nabla_\theta \left[ \ell_{\text{ro}}(f_\theta(x_i + \delta_i), y_i) + \nu \ell_{\text{nom}}(f_\theta(x_i), y_i) \right]$
\EndFor
\State $\nu \leftarrow \left[ \nu + \eta_d \left( \frac{1}{N} \sum_{n=1}^N \ell(f_\theta(x_n), y_n) - \rho \right) \right]$
\Until convergence
\end{algorithmic}
\end{algorithm}

convergence property, there is a growing body of evidence which suggests that this property does in fact hold for the function class of DNNs.

Near-optimality and near-feasibility of \text{DALE}. By combining these assumptions with the techniques used in \text{DALE}, we can explicitly bound the empirical duality gap (with high probability) and characterize the feasibility of the empirical dual optimal solution for \text{P-CON}.

Proposition 4.4 (The empirical dual of \text{P-CON}). Let $\ell(\cdot, y)$ be a convex function for all $y \in \mathcal{Y}$. Under Assumptions 4.1–4.3, it holds with probability $1 - 5\delta$ that

1. $|P^* - \hat{D}^*| \leq M\alpha + (1 + \mathcal{D}) \max(\zeta_R(N), \zeta_N(N))$; and

2. There exists $\theta^1 \in \arg\min_{\theta \in \Theta} L(\theta, \hat{v}^*)$ such that $\mathbb{E}_{(x, y) \sim \mathcal{D}} [\ell(f_{\theta^1}(x), y)] \leq \rho + \zeta_N(N)$.

Here, $\hat{v}^*$ denotes a solution of \text{DALE}, $v^*$ denotes an optimal dual variable of \text{P-CON} solved over $\mathcal{H} = \text{conv}(\mathcal{H})$ instead of $\mathcal{H}$, and $\mathcal{D} = \max(\hat{v}^*, v^*)$. Additionally, for any interpolating classifier $\theta^\dagger$, i.e. such that $\mathbb{E}_{(x, y) \sim \mathcal{D}} [\ell(f_{\theta^\dagger}(x), y)] = 0$, it holds that

$$v^* \leq \rho^{-1} \mathbb{E}_{(x, y) \sim \mathcal{D}} \left[ \max_{\delta \in \Delta} \ell(f_{\theta^\dagger}(x + \delta), y) \right].$$

The proof is provided in Appendix D. At a high level, Proposition 4.4 tells us that it is possible to learn robust models with high clean accuracy using the empirical dual problem in \text{DALE} at little cost to the sample complexity. This means that seeking a robust classifier with a given nominal performance is (probably approximately) equivalent to seeking a classifier that minimizes a combination of the nominal and adversarial empirical loss. Notably, the majority of past approaches for solving \text{P-CON} cannot be endowed with similar guarantees in the spirit of Proposition 4.4. Indeed, while the objective resembles a penalty-based formulation, notice that $\nu$ is an optimization variable rather than a fixed hyperparameter. Concretely, the magnitude of this dual variable quantifies how hard it is to learn an adversarially robust model while maintaining strong nominal performance. Though seemingly innocuous, this caveat is the difference between guaranteeing generalization only on the aggregated loss and guaranteeing generalization jointly for the objective value and constraint feasibility.

5 Dual robust learning algorithm

Under the mild assumption that $(x, y) \mapsto \ell(f_\theta(x), y) \in L^2$, Propositions 3.1, 3.2, and 4.4 allow us to transform \text{P-CON} into the following Dual Adversarial LEarning problem

$$\hat{D}^* \triangleq \max_{\nu \geq 0} \min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^N \left[ \mathbb{E}_{\delta_n} \left[ \ell(f_\theta(x_n + \delta_n), y_n) \right] + \nu \left[ \ell(f_\theta(x_n), y_n) - \rho \right] \right]$$

(P-DALE)
Figure 1: Visualizing the distribution of adversarial perturbations. In this figure, we visualize the
distribution of adversarial perturbations by projecting the perturbations generated by PGD, FGSM,
and DALE onto their first two principal components. The first and second principal components
are shown on the $x$- and $y$-axes respectively. Notice that DALE varies much less along the second
principal component vis-a-vis PGD and FGSM; this indicates that DALE tends to focus more on
directions in which the data varies most, indicating that it finds stronger adversarial perturbations.

\[
\delta_n \sim \lambda^*_n := \gamma_n^{-1} \left[ \ell(f_\theta(x_n + \delta_n), y_n) - \mu_n \right] \quad \text{for each } n \in \{1, \ldots, N\} \quad \text{and } \gamma_n > 0 \quad \text{and } \mu_n
\]

where $\delta_n \sim \lambda^*_n := \gamma_n^{-1} \left[ \ell(f_\theta(x_n + \delta_n), y_n) - \mu_n \right] \quad \text{for each } n \in \{1, \ldots, N\} \quad \text{and } \gamma_n > 0 \quad \text{and } \mu_n$ are the constants specified in Proposition 3.2. Note that this formulation is considerably more
amenable than \((P-CON)\). Indeed, it is (i) empirical and therefore does not involve unknown statistical
quantities such as $D$; (ii) unconstrained and therefore more amendable to gradient-based optimization
techniques; and (iii) its objective does not involve a challenging maximization problem in view of
the closed-form characterization of $\lambda^*$ in Proposition 3.2. In fact, for models that are linear in $\theta$
but nonlinear in the input (e.g., kernel models or logistic regression), this implies that we can transform a
non-convex, composite optimization problem \((P-CON)\) into a convex problem \((P-DALE)\).

Nevertheless, for many modern ML models such as CNNs, \((P-DALE)\) remains a non-convex program
in $\theta$. And while there is overwhelming theoretical and empirical evidence that stochastic gradient-

based algorithms yield good local minimizers for such overparametrized problems \([35-39]\), the fact
remains that solving \((P-DALE)\) requires us to evaluate an expectation with respect to $\lambda^*$, which
is challenging due to the fact that $\mu_n$ and $\gamma_n$ are not known a priori. In the remainder of this section, we
propose a practical algorithm to solve \((P-DALE)\) based on the approximation discussed in Section 3.

**Sampling from the optimal distribution $\lambda^*_n$.** Although Proposition 3.2 provides a characterization of
the optimal distribution $\lambda^*_n$, obtaining samples from $\lambda^*_n$ can still be challenging in practice, especially
when the dimension of $\delta_n$ is large (e.g., for image-classification tasks). Moreover, in practice the
value of $\gamma$ for which \((5)\) is a solution of \((4)\) is not known \emph{a priori} and can be arbitrarily close to zero,
making $\lambda^*_n$ discontinuous and with a potentially vanishing support. Fortunately, these issues can be
addressed by using Hamiltonian Monte Carlo (HMC) methods, which leverage the geometry of the
distribution to overcome the curse of dimensionality.

In particular, we propose to use a projected Langevin Monte Carlo (LMC) sampler \([51]\). To derive
this sampler, we first make a simplifying assumption: Rather than seeking the optimal constants $\gamma$ and $\mu$, we consider the over-smoothed approximation of $\lambda^*_n$ derived in \([7]\), wherein the probability
mass allocated to a particular perturbation $\delta \in \Delta$ is proportional to the loss $\ell(f_\theta(x + \delta), y)$. We
note that while this choice of $\lambda^*_n$ may not be optimal, the sampling scheme that we derive under this
assumption yields strong numerical performance. Furthermore, even if we knew the true values of
$\gamma_n$ and $\mu_n$, the resulting distribution for $\mu_n \neq 0$ would be discontinuous and sampling from such
distributions in high-dimensional settings is challenging in and of itself (see, e.g., \([82]\)).

Given this approximate characterization of the optimal distribution, the following Langevin iteration
can be derived directly from the commonly-used leapfrog simpltic integrator for the Hamiltonian
dynamics induced by the distribution $\lambda_n$ defined in \([7]\) (see Appendix E for details). This, in turn,
yields the following update rule:

\[
\delta \leftarrow \Pi_\Delta \left[ \delta + \eta \text{sign} \left( \nabla_\delta \log \left[ \ell_{\text{pert}}(f_\theta(x + \delta), y) \right] + \sqrt{2\eta}T \xi \right) \right] \quad (10)
\]
Table 1: Adversarial robustness on MNIST and CIFAR-10. Test accuracies of DALE (Algorithm 1) and state-of-the-art baselines on MNIST and CIFAR-10. On both datasets, DALE surpasses the baselines against both adversaries, while simultaneously maintaining high nominal performance.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\rho$</th>
<th>MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Clean</td>
<td>FGSM</td>
</tr>
<tr>
<td>ERM</td>
<td>-</td>
<td>99.3</td>
<td>14.3</td>
</tr>
<tr>
<td>FGSM</td>
<td>-</td>
<td>98.3</td>
<td>98.1</td>
</tr>
<tr>
<td>PGD</td>
<td>-</td>
<td>98.1</td>
<td>95.5</td>
</tr>
<tr>
<td>CLP</td>
<td>-</td>
<td>98.0</td>
<td>95.4</td>
</tr>
<tr>
<td>ALP</td>
<td>-</td>
<td>98.1</td>
<td>95.5</td>
</tr>
<tr>
<td>TRADES</td>
<td>-</td>
<td>98.9</td>
<td>96.5</td>
</tr>
<tr>
<td>MART</td>
<td>-</td>
<td>98.9</td>
<td>96.1</td>
</tr>
<tr>
<td>DALE</td>
<td>0.5</td>
<td>99.3</td>
<td>96.6</td>
</tr>
<tr>
<td>DALE</td>
<td>0.8</td>
<td>99.0</td>
<td>96.9</td>
</tr>
<tr>
<td>DALE</td>
<td>1.0</td>
<td>99.1</td>
<td>97.7</td>
</tr>
</tbody>
</table>

where $\xi \sim \text{Laplace}(0, I)$. In this notation, $T > 0$ and $\eta > 0$ are constants which can be chosen as hyperparameters, and $\ell_{\text{pert}}$ is a loss functions for the perturbation. The resulting algorithm is summarized in Algorithm 1. Notice that Algorithm 1 accounts for scenarios in which the losses associated with the adversarial performance ($\ell_{\text{adv}}$), the perturbation ($\ell_{\text{pert}}$), and the nominal performance ($\ell_{\text{nom}}$) are different. It can therefore learn from perturbations that are adversarial for a different loss than the one used for training the model $\theta$. This generality allows it to tackle different applications, e.g., by replacing the adversarial error objective in (P-CON) by a measure of model invariance (e.g., ACE in [53]). This feature can also be used to show that existing adversarial training procedures can be seen as approximations of Algorithm 1 (see Appendix A). We refer the reader to Appendix H for further discussion of the convergence properties of Algorithm 1.

6 Experiments

In this section, we include an empirical evaluation of the DALE algorithm. In particular, we consider two standard datasets: MNIST and CIFAR-10. For MNIST, we train four-layer CNNs and set $\Delta = \{\delta : \|\delta\|_{\infty} \leq 0.3\}$; for CIFAR-10, we train ResNet-18 models and set $\Delta = \{\delta : \|\delta\|_{\infty} \leq 8/255\}$. All hyperparameters and performance metrics are chosen with respect to the robust accuracy of a PGD adversary evaluated on a small hold-out validation set. Further details concerning hyperparameters and architectures are provided in Appendix F. We also provide additional experiments in Appendix G.

Evaluating the adversarial robustness of DALE. We begin our empirical evaluation by comparing the adversarial robustness of DALE with numerous state-of-the-art baselines in Table 1. To evaluate the robust performance of these classifiers, we use a 1-step and an $L$-step PGD adversary to evaluate robust performance; we denote these adversaries by FGSM and PGD$^L$ respectively. Notice that on CIFAR-10, DALE with $\rho = 0.8$ is the only method to achieve higher than 85% clean accuracy and 50% adversarial accuracy against PGD$^{20}$. Furthermore, when DALE is run with $\rho = 1.1$, we see that it achieves nearly 52% adversarial accuracy, which is a significant improvement over all baselines. In Appendix G, we provide a more complete characterization of the role of $\rho$ in controlling the trade-off between robustness and accuracy.

Visualizing the distribution of adversarial perturbations. To visualize the distribution over perturbations generated by DALE, we use principal component analysis (PCA) to embed perturbations into a two-dimensional space. In particular, we performed PCA on the MNIST training set to extract the first two principal components of the images; we then projected the perturbations $\delta \in \Delta$ generated by PGD, FGSM, and DALE in the last iteration of training onto these principal components. A plot of these projections is shown in Figure 1 in which the first and second principal components are shown on the $x$- and $y$-axes respectively. Notice that the perturbations generated by FGSM are spread out somewhat unevenly in this space. In contrast, the perturbations found by PGD and DALE are spread out more evenly. Furthermore, the perturbations generated by PGD and FGSM vary more
Figure 2: Tracking the dual variables. Left: the clean and robust test accuracies of a ResNet-18 classifier trained on CIFAR-10 using DALE. Middle: the training losses for DALE. Right: The magnitude of the dual variable during training.

along the second principal component (y-axis) than the first (x-axis) relative to DALE. Since the first component describes the direction of largest variance of the data, this indicates that DALE tends to find perturbations that place more mass on the direction in which the data varies most.

Tracking the dual variables. In Figure 2, we study the performance of DALE over the course of training. In the leftmost panel, we plot the test accuracy on clean samples and on adversarially perturbed samples. Notably, this classifier exceeds 50% robust accuracy against PGD\textsuperscript{20} as well as 85% clean accuracy; these figures are higher than either of the corresponding metrics for any of the baselines in Table 1, indicating that our method is more effectively able to mitigate the trade-off between robustness and accuracy. In the middle panel of Figure 2, we track the nominal and robust training losses, and in the rightmost panel, we show the magnitude of the dual variable $\nu$. Observe that at the onset of training, the constraint in (P-CON) is not satisfied, as the blue curve is above the red dashed-line. In response, the dual variable places more weight on the nominal loss term in (P-DALE). After several epochs, this reweighting forces constraint satisfaction, after which the dual variable begins to decrease, which in turn decreases the weight on the nominal objective and allows the optimizer to focus on minimizing the robust loss.

Regularization vs. primal-dual. Our final ablation study is to consider the impact of performing the dual-update step in line 10 of Algorithm 1. In particular, in Table 2, we record the performance of DALE when Algorithm 1 is run without the dual update step. This corresponds to running DALE with a fixed weight $\nu$. Notice that although our method reaches the same level of robust performance as MART and TRADES, it does not match the performance of the DALE classifiers in Table 1. This indicates that the strong robust performance of our algorithm relies on adaptively updating the dual variable over the course of training.

7 Conclusion

In this paper, we studied robust learning from a constrained learning perspective. We proved an equivalence between the standard adversarial training paradigm and a stochastic optimization problem over a specific, non-atomic distribution. This insight provided a new perspective on robust learning and engendered a Langevin MCMC approach for adversarial robustness. We experimentally validated that this algorithm outperforms the state-of-the-art on standard benchmarks. Notably, our method simultaneously achieved greater than 50% adversarial accuracy and greater than 85% clean accuracy on CIFAR-10, which represents a significant improvement over the previous state-of-the-art.

Table 2: Regularized DALE. Test accuracies attained by running DALE without the dual-update step in line 10 of Algorithm 1.

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>Clean</th>
<th>FGSM</th>
<th>PGD\textsuperscript{20}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>86.4</td>
<td>55.3</td>
<td>49.5</td>
</tr>
<tr>
<td>0.2</td>
<td>86.8</td>
<td>54.2</td>
<td>49.3</td>
</tr>
<tr>
<td>0.3</td>
<td>86.3</td>
<td>54.8</td>
<td>48.2</td>
</tr>
<tr>
<td>0.4</td>
<td>86.2</td>
<td>54.6</td>
<td>47.3</td>
</tr>
<tr>
<td>0.5</td>
<td>86.5</td>
<td>54.3</td>
<td>46.8</td>
</tr>
<tr>
<td>0.6</td>
<td>85.7</td>
<td>53.3</td>
<td>46.4</td>
</tr>
<tr>
<td>0.7</td>
<td>85.8</td>
<td>53.3</td>
<td>46.0</td>
</tr>
<tr>
<td>0.8</td>
<td>84.9</td>
<td>53.1</td>
<td>45.9</td>
</tr>
<tr>
<td>0.9</td>
<td>85.0</td>
<td>53.4</td>
<td>45.7</td>
</tr>
<tr>
<td>1.0</td>
<td>84.5</td>
<td>52.7</td>
<td>45.8</td>
</tr>
</tbody>
</table>
8 Acknowledgements and disclosure of funding

The authors would like to thank Juan Cervino and Samuel Sokota for their helpful feedback.

The research of Alexander Robey and Hamed Hassani is supported by NSF Grants 1837253, 1943064, 1934876, AFOSR Grant FA9550-20-1-0111, and DCIST-CRA. Alexander Robey, George J. Pappas, Luiz F. O. Chamon, and Alejandro Ribeiro are all supported by the NSF-Simons Program on the Mathematics of Deep Learning. Alexander Robey and George J. Pappas are supported by ARL CRA DCIST W911NF-17-2-0181 and NSF 2038873.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]
References


13


