Abstract

Is there a classifier that ensures optimal robustness against all adversarial attacks? This paper tackles the question by adopting a game-theoretic point of view. We present the adversarial attacks and defenses problem as an infinite zero-sum game where classical results (e.g. Nash or Sion theorems) do not apply. We demonstrate the non-existence of a Nash equilibrium in our game when the classifier and the Adversary are both deterministic, hence giving a negative answer to the above question in the deterministic regime. Nonetheless, the question remains open in the randomized regime. We tackle this problem by showing that, under mild conditions on the dataset distribution, any deterministic classifier can be outperformed by a randomized one. This gives arguments for using randomization, and leads us to a simple method for building randomized classifiers that are robust to state-or-the-art adversarial attacks. Empirical results validate our theoretical analysis, and show that our defense method considerably outperforms Adversarial Training against strong adaptive attacks, by achieving 0.55 accuracy under adaptive PGD-attack on CIFAR10, compared to 0.42 for Adversarial training.

1. Introduction

Adversarial example attacks recently became a major concern in the machine learning community. An adversarial attack refers to a small, imperceptible change of an input that is maliciously designed to fool a machine learning algorithm. Since the seminal work of (Biggio et al., 2013) and (Szegedy et al., 2014) it became increasingly important to understand the very nature of this phenomenon (Fawzi et al., 2016; 2018; Bubeck et al., 2019; Ilyas et al., 2019; Gourdeau et al., 2019). Furthermore, a large body of work has been published on designing attacks (Goodfellow et al., 2015; Papernot et al., 2016a; Madry et al., 2018; Carlini & Wagner, 2017; Athalye et al., 2018) and defenses (Goodfellow et al., 2015; Papernot et al., 2016b; Madry et al., 2018; Cohen et al., 2019).

Besides, in real-life scenarios such as for an autonomous car, errors can be very costly. It is not enough to just defend against new attacks as they are published. We would need an algorithm that behaves optimally against every single attack. However, it remains unknown whether such a defense exists. This leads to the following questions, for which we provide principled and theoretically-grounded answers.

Q1: Is there a deterministic classifier that ensures optimal robustness against any adversarial attack?

A1: To answer this question, in Section 3.1, we cast the adversarial examples problem as an infinite zero-sum game between a Defender (the classifier) and an Adversary that produces adversarial examples. Then we demonstrate, in Section 4, the non-existence of a Nash equilibrium in the deterministic setting of this game. This entails that no deterministic classifier can claim to be more robust than all other classifiers against any possible adversarial attack. Another consequence of our analysis is that there is no free lunch for transferable attacks: an attack that works on all classifiers will never be optimal against any of them.

Q2: Would randomized defense strategies be a suitable alternative to defend against strong adversarial attacks?

A2: We tackle this problem both theoretically and empirically. In Section 5, we demonstrate, under a mild condition on the data distribution, that for any deterministic defense there exists a mixture of classifiers that offers better worst-case theoretical guarantees. Building upon this, we devise a method that generates a robust randomized classifier with a 1 step boosting method. We evaluate this method, in Section 6 against strong adaptive attacks on the CIFAR10 and CIFAR100 datasets. It outperforms Adversarial Training against both $\ell_\infty$-PGD (Madry et al., 2018), and
2. Related Work

Many works have studied adversarial examples, in several different settings. We discuss hereafter the different frameworks that we believe to be related to our work, and discuss the aspects on which our contribution differs from them.

Distributionally robust optimization. The work in (Sinha et al., 2018) addresses the problem of adversarial examples through the lens of distributionally robust optimization. They study a min-max problem where the Adversary manipulates the test distribution while being constrained in a Wasserstein distance ball (they impose a global constraint when the players only have a finite set of possible strategies that we believe to be related to our work, and discuss (Lecuyer et al., 2018), followed by (Cohen et al., 2019; Li (Rota Bulò et al., 2017) and (Perdomo & Singer, 2019) formalized by a push-forward measure. This kind of setting setting, in which the Adversary manipulating the dataset is infinite mixture of perturbed classifiers.

classifier in which noise is being injected can be seen as an noise injection. However, a link could be made between defense method we propose in this paper are not based on defense against adversarial attacks. The analysis and demonstrated that noise injection can, in some cases, give provable 0\ell -C&W (resp. 0.14) over Adversarial Training.


classification task. Let \( \mathcal{X} \subset \mathbb{R}^d \) and \( \mathcal{Y} = \{-1, 1\} \). We consider a distribution \( \mathcal{D} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y}) \) that we assume to be of support \( \mathcal{X} \times \mathcal{Y} \). The Defender is looking for a hypothesis (classifier) \( h \) in a class of functions \( \mathcal{H} \), minimizing the risk of \( h \) w.r.t. \( \mathcal{D} \):

\[
\mathcal{R}(h) := \mathbb{E}_{(X,Y) \sim \mathcal{D}} \left[ \mathbb{I} \{ h(X) \neq Y \} \right] = \mathbb{E}_{Y \sim \nu} \left[ \mathbb{E}_{X \sim \mu_Y} \left[ \mathbb{I} \{ h(X) \neq Y \} \right] \right].
\] (1)

Where \( \mathcal{H} := \{ h : x \mapsto \text{sgn} \mathcal{g}(x) \mid g : \mathcal{X} \to \mathbb{R} \text{ continuous}, \nu \in \mathcal{P}(\mathcal{Y}) \text{ is the probability measure that defines the law of the random variable } Y, \text{ and for any } y \in \mathcal{Y}, \mu_y \in \mathcal{P}(\mathcal{X}) \text{ is the conditional law of } X \mid (Y = y) \).
Adversarial example attack (point-wise). Given a classifier $h : \mathcal{X} \to \mathcal{Y}$ and a data sample $(x, y) \sim D$, the Adversary seeks a perturbation $\tau \in \mathcal{X}$ that is visually imperceptible, but modifies $x$ enough to change its class, i.e. $h(x + \tau) \neq y$. Such a perturbation is called an adversarial example attack. In practice, it is hard to evaluate the set of visually imperceptible modifications of an image. However, a sufficient condition to ensure that the attack is undetectable is to constrain the perturbation $\tau$ to have a small norm, be it for the $\ell_\infty$ or the $\ell_2$ norm. Hence, one should always ensure that $\|\tau\|_\infty \leq \epsilon_\infty$, or $\|\tau\|_2 \leq \epsilon_2$, depending on the norm used to measure visual imperceptibility. The choice of the threshold depends on the application at hand. For example, on CIFAR datasets, typical values for $\epsilon_\infty$ and $\epsilon_2$ are respectively, 0.031 and 0.4/0.6/0.8. In the remaining of this work, we will define our constraint using an $\ell_2$ norm, but all our results remains valid for an $\ell_\infty$ based constraint.

Adversarial example attack (distributional). The Adversary chooses, for every $x \in \mathcal{X}$, a perturbation that depends on its true label $y$. This amounts to construct, for each label $y \in \mathcal{Y}$, a measurable function $\phi_y$ such that $\phi_y(x)$ is the perturbation associated with the labeled example $(x, y)$. This function naturally induces a probability distribution over adversarial examples, which is simply the push-forward function naturally induces a probability distribution over the example of the autonomous car, this would mean that the Adversary has perfect information on the dataset distribution and the classifier. It also has unlimited computational power and no constraint on the attack except on the size of the perturbation. Going back to the example of the autonomous car, this would mean that the Adversary can modify every single image that the camera may receive during any trip, which is highly unrealistic. The Adversary has no downside to attacking, even when the attack is unnecessary, e.g. if the attack cannot work or if the point is already misclassified.

Trivial Nash equilibrium. Our current definition of the problem implies that the Adversary has perfect information on the dataset distribution and the classifier. It also has unlimited computational power and no constraint on the attack except on the size of the perturbation. Going back to the example of the autonomous car, this would mean that the Adversary can modify every single image that the camera may receive during any trip, which is highly unrealistic. The Adversary has no downside to attacking, even when the attack is unnecessary, e.g. if the attack cannot work or if the point is already misclassified.

This type of behavior for the Adversary can lead to the existence of a pathological (and trivial) Nash Equilibrium as demonstrated in Figure 1 for the uni-dimensional setting with Gaussian distributions. The unbounded Adversary moves every point toward the decision boundary (each time maximizing the perturbation budget), and the Defender cannot do anything to mitigate the damage. In this case the
decision boundary for the Optimal Bayes Classifier remains unchanged, even though both curves have been moved toward the center, hence a trivial equilibrium. In the remaining of this work, we show that such an equilibrium does not exist as soon as there is a small restraint on the Adversary’s strength, i.e. as soon as it is not perfectly indifferent to produce unnecessary perturbations.

**Regularized Adversary.** To mitigate the Adversary strength, we introduce a penalization term:

\[
\inf_{h \in \mathcal{H}} \sup_{\phi \in (\mathcal{F}_{\epsilon h})^2} \left[ \mathcal{R}_\text{adv}(h, \phi) - \lambda \Omega(h, \phi) \right].
\]

The penalty function \( \Omega \) represents the limitations on the Adversary’s budget, be it because of computational resources or to avoid being detected. \( \lambda \in (0, 1) \) is some regularization weight. In this paper, we study two types of penalties: the mass penalty \( \Omega_{\text{mass}} \), and the norm penalty \( \Omega_{\text{norm}} \).

From a computer-security point of view, the first limitation that comes to mind is to limit the number of queries the Adversary can send to the classifier. In our distributional setting, this boils down to penalizing the mass of points that the function \( \phi \) moves. Hence we define the mass penalty as:

\[
\Omega_{\text{mass}}(\phi) := \mathbb{E}_{Y \sim \nu} \mathbb{E}_{X \sim \mu_Y} \mathbb{1}\{X \neq \phi_Y(X)\}.
\]

The mass penalty discourages the Adversary from attacking too many points by penalizing the overall mass of transported points. The second limitation we consider penalizes the expected norm under \( \phi \):

\[
\Omega_{\text{norm}}(\phi) := \mathbb{E}_{Y \sim \nu} \mathbb{E}_{X \sim \mu_Y} \|X - \phi_Y(X)\|_2.
\]

This regularization is very common in both the optimization and adversarial example communities. In particular, it is used by Carlini & Wagner (Carlini & Wagner, 2017) to compute the eponymous attack. In the following, we denote \( \mathcal{BR}_{\Omega_{\text{norm}}} \) (resp. \( \mathcal{BR}_{\Omega_{\text{mass}}} \)) the best responses for the Adversary w.r.t the mass (resp. norm) penalty. Section 4 shows that whatever penalty the Adversary has, no Pure Nash Equilibrium exists. We characterize the best responses for each player, and show that they can never satisfy Definition 2.

### 4. Deterministic regime

**Notations.** Let \( h \in \mathcal{H} \), we denote \( P_h := \{ x \in \mathcal{X} \mid h(x) = 1 \} \), and \( N_h := \{ x \in \mathcal{X} \mid h(x) = -1 \} \) respectively the set of positive and negative outputs of \( h \). We also denote the set of attackable points from the positive outputs \( P_h(\delta) := \{ x \in P_h \mid \exists z \in N_h \text{ and } \|z - x\|_2 \leq \delta \} \), and \( N_h(\delta) \) likewise.

**Adversary’s best response.** Let us first present the best responses of the Adversary under respectively the mass penalty and the norm penalty. Both best responses share a fundamental behavior: the optimal attack will only change points that are close enough to the decision boundary. This means that, when the Adversary has no chance of making the classifier change its decision about a given point, it will not attack it. However, for the norm penalty all attacked points are projected on the decision boundary, whereas with the mass penalty the attack moves the points across the border.

**Lemma 1.** Let \( h \in \mathcal{H} \) and \( \phi \in \mathcal{BR}_{\Omega_{\text{norm}}}(h) \). Then the following assertion holds:

\[
\begin{aligned}
\phi_1(x) &= \{ \pi(x) \text{ if } x \in P_h(\epsilon_2) \} \\
&\text{otherwise.}
\end{aligned}
\]

Where \( (P_h)^C \), the complement of \( P_h \) in \( \mathcal{X} \). \( \phi_1 \) is characterized symmetrically.

**Lemma 2.** Let \( h \in \mathcal{H} \) and \( \phi \in \mathcal{BR}_{\Omega_{\text{mass}}}(h) \). Then the following assertion holds:

\[
\phi_1(x) = \{ \pi(x) \text{ if } x \in P_h(\epsilon_2) \} \\
\text{otherwise.}
\]

Where \( \pi \) is the orthogonal projection on \( (P_h)^C \). \( \phi_1 \) is characterized symmetrically.

These best responses are illustrated in Figure 1 with two unidimensional Gaussian distributions. For the mass penalty, \( \mu_1 \) is set to 0 in \( P_h(\epsilon_2) \), and this mass is transported into \( N_h(\epsilon_2) \). The symmetric holds for \( \mu_1 \). After attack, we now have \( \mu_1(\mathcal{P}_h(\epsilon_2)) = 0 \), so a small value of \( \mu_1 \) in \( P_h(\epsilon_2) \) suffices to make it dominant, and that zone will now be classified -1 by the Optimal Bayes Classifier. For the norm penalty, the part of \( \mu_1 \) that was in \( P_h(\epsilon_2) \) is transported on a Dirac distribution at the decision boundary. Similarly to the mass penalty, the best response now predicts -1 for the zone \( P_h(\epsilon_2) \).

**Remark.** In practice, it might be computationally hard to generate the exact best response for the norm penalty, i.e. the projection on the decision boundary. That will happen for example if this boundary is very complex (e.g. highly non-smooth), or when \( \mathcal{X} \) is in a high dimensional space. To keep the attack tractable, the Adversary will have to compute an approximated best response by allowing the projection to reach the point within a small ball around the boundary. This means that the best responses of the norm penalty and the mass penalty problems will often match.

**Defender’s best response.** At a first glance, one would suspect that the best response for the Defender ought to be
the Optimal Bayes Classifier for the transported distribution. However, it is only well defined if the conditional distributions admit a probability density function. This might not always hold here for the transported distribution. Nevertheless, we show that there is a property, shared by the Optimal Bayes Classifier when defined, that always holds for the Defender’s best response.

**Lemma 3.** Let us consider \( \phi \in (\mathcal{F}_X|\epsilon_2)^2 \). If we take \( h \in \mathcal{BH}((\phi)) \), then for \( y = 1 \) (resp. \( y = -1 \)), and for any \( B \subset P_h \) (resp. \( B \subset N_h \)) one has

\[
\mathbb{P}(Y = y | X \in B) \geq \mathbb{P}(Y = -y | X \in B)
\]

with \( Y \sim \nu \) and for all \( y \in \mathcal{Y} \), \( X|(Y = y) \sim \phi_y#\mu_y \).

In particular, when \( \phi_1#\mu_1 \) and \( \phi_1#\mu_1 \) admit probability density functions, Lemma 3 simply means that \( h \) is the Optimal Bayes Classifier for the distribution \( (\nu, \phi_1#\mu_1, \phi_1#\mu_1)^2 \). We can now state our main theorem, as well as two of its important consequences.

**Theorem 1** (Non-existence of a pure Nash equilibrium).

In the zero-sum game (Eq. 4) with \( \lambda \in (0, 1) \) and penalty \( \Omega \in \{\Omega_{max}, \Omega_{norm}\} \), there is no Pure Nash Equilibrium.

**Consequence 1.** (No free lunch for transferable attacks)

To understand this statement, remark that, thanks to weak duality, the following inequality always holds:

\[
\sup_{\phi \in (\mathcal{F}_X|\epsilon_2)^2} \inf_{h \in \mathcal{H}} \mathcal{R}_\Omega^{\text{adv}}(h, \phi) \leq \inf_{h \in \mathcal{H}} \sup_{\phi \in (\mathcal{F}_X|\epsilon_2)^2} \mathcal{R}_\Omega^{\text{adv}}(h, \phi).
\]

On the left side problem (sup-inf), the Adversary looks for the best strategy \( \phi \) against any unknown classifier. This is tightly related to the notion of transferable attacks (see e.g. (Tramèr et al., 2017)), which refers to attacks successful against a wide range of classifiers. On the right side (our) problem (inf-sup), the Defender tries to find the best classifier under any possible attack, whereas the Adversary plays in second and specifically attacks this classifier. As a consequence of Theorem 1, the inequality is always strict:

\[
\sup_{\phi \in (\mathcal{F}_X|\epsilon_2)^2} \inf_{h \in \mathcal{H}} \mathcal{R}_\Omega^{\text{adv}}(h, \phi) < \inf_{h \in \mathcal{H}} \sup_{\phi \in (\mathcal{F}_X|\epsilon_2)^2} \mathcal{R}_\Omega^{\text{adv}}(h, \phi).
\]

This means that both problems are not equivalent. In particular, an attack designed to succeed against any classifier \( i.e. \) a transferable attack \( i.e. \) will not be as good as an attack tailored for a given classifier. Hence she has to trade-off between effectiveness and transferability of the attack.

**Consequence 2.** (No deterministic defense may be proof against every attack) Let us consider the state-of-the-art defense which is Adversarial Training (Goodfellow et al., 2015; Madry et al., 2018). The idea is to compute an efficient attack \( \phi \), and train the classifier on created adversarial examples, in order to move the decision boundary and make the classifier more robust to new perturbations by \( \phi \).

To be fully efficient, this method requires that \( \phi \) remains an optimal attack on \( h \) even after training. Our theorem shows that it is never the case: after training our classifier \( h \) to become \( (h') \) robust against \( \phi \), there will always be a different optimal attack \( \phi' \) that is efficient against \( h' \). Hence Adversarial Training will never achieve a perfect defense.

**5. Randomization matters**

As we showed that there is no Pure Nash Equilibrium, no deterministic classifier may be proof against every attack. We would therefore need to allow for a wider class of strategies. A natural extension of the game would thus be to allow randomization for both players, who would now choose a distribution over pure strategies, leading to this game:

\[
\inf_{\eta \in \mathcal{P}(\mathcal{H})} \sup_{\phi \in (\mathcal{F}_X|\epsilon_2)^2} \mathbb{E}_{h \sim \eta} [\mathcal{R}_\Omega^{\text{adv}}(h, \phi)].
\]

Without making further assumptions on this game (e.g. compactness), we cannot apply known results from game theory (e.g. Sion theorem) to prove the existence of an equilibrium. These assumptions would however make the problem loose much generality, and does not hold here.

**Randomization matters.** Even without knowing if an equilibrium exists in the randomized setting, we can prove that randomization matters. More precisely we show that, under mild condition on the data distribution, any deterministic classifier can be outperformed by a randomized one in terms...
of the worst case adversarial risk. To do so we simplify Equation 7 in two ways: 1. We do not consider the Adversary to be randomized, i.e. we restrict the search space of the Adversary to $\mathcal{F}_X^2$ instead of $\mathcal{P}(\mathcal{F}_X^2)$. This condition corresponds to the current state-of-the-art in the domain: to the best of our knowledge, no efficient randomized adversarial example attack has been designed (and so is used) yet. 2. We only consider a subclass of randomized classifiers, called mixtures, which are discrete probability measures on a finite set of classifier. We show that this kind of randomization is enough to strictly outperform any deterministic classifier. We will discuss later the use of more general randomization (such as noise injection) for the Defender.

Let us now define a mixture of classifiers.

**Definition 3 (Mixture of classifier).** Let $n \in \mathbb{N}$, $h = (h_1, ..., h_n) \in \mathcal{H}^n$, and $q \in \mathcal{P}\{1, ..., n\}$. A mixed classifier of $h$ by $q$ is a mapping $m^q_h$ from $\mathcal{X}$ to $\mathcal{P}(\mathcal{Y})$ such that for all $x \in \mathcal{X}$, $m^q_h(x)$ is the discrete probability distribution that is defined for all $y \in \mathcal{Y}$ as follows:

$$m^q_h(x)(y) := \mathbb{E}_{i \sim q} \left[ \mathbb{1}\{h_i(x) = y\} \right].$$

We call such a mixture a **mixed strategy** of the Defender. Given some $x \in \mathcal{X}$, this amounts to picking a classifier $h_i$ from $h$ at random following the distribution $q$, and use it to output the predicted class for $x$, i.e. $h_i(x)$. Note that a mixed strategy for the Defender is a non deterministic algorithm, since it depends on the sampling one makes on $q$. Hence, even if the attacks are defined in the same way as before, the Adversary now needs to maximize a new objective function which is the expectation of the adversarial risk under the distribution $m^q_h$. It writes as follows:

$$\mathbb{E}_{Y \sim \nu} \left[ \mathbb{E}_{X \sim \phi_Y \# \mu_Y} \left[ \mathbb{E}_{Y \sim m^q_h(X)} \left[ \mathbb{1}\{\hat{Y} \neq Y\} \right]\right]\right] - \lambda \Omega(\phi).$$

We also write $\mathcal{R}^\Omega_{\text{adv}}$ to mean the left part of Equation (8), when it is clear from context that the Defender uses a mixed classifier. Using this new set of strategies for the Defender, we can study whether mixed classifiers outperform deterministic ones, and how to efficiently design them.

**Mixed strategy.** We demonstrate that the efficiency of any deterministic defense can be improved using a simple mixed strategy. This method presents similarities with the notions of fictitious play (Brown, 1951) in game theory, and boosting in machine learning (Freund & Schapire, 1995). Given a deterministic classifier $h_1$, we combine it (via randomization) with the best response $h_2$ to its optimal attack.

The rational behind this idea is that, by construction, efficient attacks on one of these two classifiers will not work on the other. Mixing $h_1$ with $h_2$ has two opposite consequences on the adversarial risk. On one hand, where we only had to defend against attack on $h_1$, we are now also vulnerable to attacks on $h_2$, so the total set of possible attacks is now bigger. On the other hand, each attack will only work part of the time, depending on the probability distribution $q$. If we can calibrate the weights so that attacks on important zones have a low probability of succeeding, then the average risk under attack on the mixture will be low. This leads to the following condition on the data distribution.

**Definition 4 ($\epsilon$-dilation and vanishing measure).** Let $U \subset \mathcal{X}$, $\epsilon > 0$, and $\mu$ a probability measure.

1. The $\epsilon$-dilation of $U$ is as follows: $U \oplus \epsilon := \{u + v \mid (u, v) \in U \times \mathcal{X} \text{ and } \|v\|_2 \leq \epsilon\}$.

2. We say that $\mu$ is $\epsilon$-vanishing on $U$ if we have: $\mu(U \oplus \epsilon \setminus U) \leq \mu(U)$.

**On the vanishing measure condition.** To better understand this property, let us briefly introduce the following toy example. In Figure 2 we present a simple setting of
When we use Adversarial Training for the star points (Figure 2, left). The general tactic to defend against an attack is to change the classifier output when points are close to the boundary. This can be done all the time, as in Adversarial Training (where we move the decision boundary to incorporate adversarial examples), or part of the time as in a randomized algorithm (so that the attack only works with a given probability).

When we use Adversarial Training for the star points (Figure 2, middle), we change the output on the blue zone, so that four of the star (squared) points cannot be successfully attacked anymore. But in exchange, the dilation of the new boundary can now be attacked. For Adversarial Training to work, we need the number of new potential attacks (i.e. the points that are circled, 2 crosses in the dilatation and 2 stars that are close to the new boundary) to be smaller than the number of attacks we prevent (the squared points, 4 blue ones that an attack would send in the blue zone, and 3 red points that are far from the new decision boundary). Here we prevent 7 attacks at the cost of four new ones, so the Adversarial Training improves the total score from 10 to 7. As we just saw, Adversarial Training only works if we do not create more adversarial example than we prevent by moving the decision boundary.

Similarly, we observe what happens for the randomized defense (Figure 2, right). We mix the Optimal Bayes Classifier with the best response to attacking all the points. We get a classifier that is deterministic outside the gray area, and random inside it\(^3\). If the first classifier has a weight of \(\alpha = 0.5\), the 10 old attacks now succeed only with probability 0.5 (the new optimal attack for stars being to leave them in place), whereas 3 new attacks are created (stars outside of the gray area) that succeed with probability 0.5. The total attack score is of 6.5, which is lower than the old attack score of 10. Once again, the defense only works because there are lesser points that are far from the decision boundary than the the one that are close.

This toy examples highlights the fact that when no measure have any vanishing zone, neither randomized defense, nor Adversarial Training cannot bring any gain. Hence, the role of Definition 4 is to check that there is at least one defense that could help improving the classifier robustness. Then, Theorem 2 shows that whenever a deterministic classifier can be improved (e.g. by Adversarial Training), it will be outperformed under optimal attack by a randomized algorithm. We now can state our second main result:

**Theorem 2.** (Randomization matters) Let us consider \(h_1 \in H, \lambda \in (0, 1), \Omega = \Omega_{\text{norm}}, \phi \in \mathcal{B}_\lambda(h_1)\) and \(h_2 \in \mathcal{B}_\Omega(h_1)\). If \(\mu_1\) (resp. \(\mu_2\)) is \(\epsilon_2\)-vanishing on \(P_{h_1}(e_2)\) (resp. on \(N_{h_1}(e_2)\)), then for any \(\alpha \in (\frac{1+\lambda}{2}, 1)\) and for any \(\phi' \in \mathcal{B}_\Omega(m^q_h)\) one has

\[R_{\text{adv}}^\Omega(m^q_h, \phi') < R_{\text{adv}}(h_1, \phi).\]

Where \(h = (h_1, h_2), q = (\alpha, 1 - \alpha), \) and \(m^q_h\) is the mixture of \(h\) by \(q\). A similar result holds when \(\Omega = \Omega_{\text{max}}\), with \(\alpha \in (\frac{1+\lambda}{2}, 1)\).

Based on Theorem 2 we devise a new procedure called Boosted Adversarial Training (BAT) to construct a robust mixture of two classifiers. It is based on three core principles: Adversarial Training, Boosting and Randomization.

### 6. Experiments: How to build the mixture

**Simple mixture procedure (BAT).** Given a dataset \(D\) and a weight parameter \(\alpha \in [0, 1]\), we construct \(h_1\) the first classifier of the mixture using Adversarial Training\(^4\) on \(D\). Then, we train the second classifier \(h_2\) on a data set \(\tilde{D}\) that contains adversarial examples against \(h_1\) created from examples of \(D\). At the end we return the mixture constructed with those two classifiers where the first one has a weight of \(1 - \alpha\) and the second one a weight of \(\alpha\). The parameter \(\alpha\) is found by conducting a grid-search. In Table 1 we present results for \(\alpha = 0.2\) under strong state-of-the-art attacks. The procedure is summarized in Algorithm 1\(^5\).

**Algorithm 1** Boosted Adversarial Training

**Input**: \(D\) the training data set and \(\alpha\) the weight parameter.

Create and adversarially train \(h_1\) on \(D\)

Generate the adversarial data set \(\tilde{D}\) against \(h_1\).

Create and naturally train \(h_2\) on \(\tilde{D}\)

\[q \leftarrow (1 - \alpha, \alpha)\]

\[h \leftarrow (h_1, h_2)\]

return \(m^q_h\)

**Comparison to fictitious play.** Contrary to classical algorithms such as Fictitious play that also generates mixtures of classifiers, and whose theoretical guarantees rely on the existence of a Mixed Nash Equilibrium, the performance of our method is ensured by Theorem 2 to be at least as good as the classifier it uses as a basis. Moreover, the implementation of Fictitious Play would be impractical on high dimensional dataset we consider, due to computational costs.

\(^3\)The grey area should actually be bigger since the best response to the attack would also change the decision on the upper part between the OBC and the dotted line. We focus on what happens on the star points for simplicity.

\(^4\)We use \(\ell_{\infty}\)-PGD with 20 iterations and \(\epsilon_{\infty} = 0.031\) to train the first classifier and to build \(\tilde{D}\).

\(^5\)More algorithmic and implementation details can be found in the supplementary materials.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Natural Accuracy</th>
<th>Adaptive-$\ell_\infty$-PGD $\epsilon_\infty = 0.031$</th>
<th>Adaptive-$\ell_2$-C&amp;W $\epsilon_2 = 0.4$ $\epsilon_2 = 0.6$ $\epsilon_2 = 0.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>Natural</td>
<td>0.88</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>AT (Madry et al., 2018)</td>
<td>0.83</td>
<td>0.42</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.80</td>
<td>0.55</td>
<td>0.60</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>Natural</td>
<td>0.62</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>AT (Madry et al., 2018)</td>
<td>0.58</td>
<td>0.26</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.56</td>
<td>0.40</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 1. Evaluation on CIFAR10 and CIFAR100 without data augmentation. Accuracy under attack of a single adversarially trained classifier (AT) and the mixture formed with our method (Ours). The evaluation is made with Adaptive-$\ell_\infty$-PGD and Adaptive-$\ell_2$-C&W attacks both computed with 100 iterations. For Adaptive-$\ell_\infty$-PGD we use an epsilon equal to $8/255 \approx 0.031$, a step size equal to $2/255 \approx 0.008$ and we allow random initialization. For Adaptive-$\ell_2$-C&W we use a learning rate equal to 0.01, 9 binary search steps, the initial constant to 0.001, we allow the abortion when it has already converged and we give the results for the different values of rejection threshold $\epsilon_2 \in \{0.4, 0.6, 0.8\}$. As for EOT, we don’t need to estimate the expected accuracy of the mixture through Monte Carlo sampling since we have the exact weight of each classifier of the mixture. Thus we give the exact expected accuracy.

Evaluating against strong adversarial attacks. When evaluating a defense against adversarial examples, it is crucial to test the robustness of the method against the best possible attack. Accordingly, the defense method should be evaluated against attacks that were specifically tailored to it (a.k.a. adaptive attacks). In particular, when evaluating randomized algorithms, one should use Expectation over Transformation (EOT) to avoid gradient masking as pointed out by (Athalye et al., 2018) and (Carlini et al., 2019). More recently, (Tramer et al., 2020) emphasized that one should also make sure that EOT is computed properly in previous works such as (Dhillon et al., 2018) and (Pinot et al., 2019) estimate the EOT through a Monte Carlo sampling which can introduce a bias in the attack if the sample size is too small. Since we assume perfect information for the Adversary, it knows the exact distribution of the mixture. Hence it can directly compute the expectation without using a sampling method, which avoid any bias. Table 1 evaluates our method against strong adaptive attacks namely Adaptive-$\ell_\infty$-PGD and Adaptive-$\ell_2$-C&W.

Hard constraint parameter. The typical value of $\epsilon$ in the hard constraint depends on the norm we consider in the problem setting. In this paper, we use an $\ell_2$ norm, however, the constraint parameter for $\ell_\infty$-PGD attack was initially set to be an $\ell_\infty$ constraint. In order to compare attacks of similar strength, we choose different threshold ($\epsilon_2$ or $\epsilon_\infty$) values which result in balls of equivalent volumes. For CIFAR10 an CIFAR100 datasets (Krizhevsky & Hinton, 2009), which are $3 \times 32 \times 32$ dimensional spaces, this gives $\epsilon_\infty = 0.03$ and $\epsilon_2 = 0.8$ (we also give results for $\epsilon_2$ equal to 0.6 and 0.4 as this values are sometimes used in the literature). Since Adaptive-$\ell_2$-C&W attack creates an unbounded perturbation on the examples, we implemented the constraint from Equation 6 by checking at test time whether the $\ell_2$-norm of the perturbation exceeds a certain threshold $\epsilon_2 \in \{0.4, 0.6, 0.8\}$. If it does, the adversarial example is disregarded, and we keep the natural example instead.

Experimental results. In Table 1 we compare the accuracy, on the CIFAR10 and CIFAR100, of our method and classical Adversarial Training under attack with Adaptive-$\ell_\infty$-PGD and Adaptive-$\ell_2$-C&W, both run for 100 iterations. We used 5 times more iteration for the evaluation as we used during training, and carefully check for convergence. the rational behind this is that, for a classifier to be fully robust, its loss of accuracy should be controlled when the attacks are stronger than the ones it was trained on. For both attacks, both datasets and all thresholds (i.e. the budget for a perturbation), the accuracy under attack of our mixture is higher than the single classifier with Adversarial Training. Our defense is especially more robust than Adversarial Training when the threshold is high.

Extension to more than two classifiers. In this paper we focus our experiments on a mixture of two classifiers to present a proof of concept of Theorem 2. Nevertheless, a mixture of more than two classifiers can be constructed by adding at each step a new classifier trained naturally on the dataset $\hat{D}$ that contains adversarial examples against the mixture at step $t - 1$. Since $\hat{D}$ has to be constructed from a mixture, one would have to use an adaptive attack as Adaptive-$\ell_\infty$-PGD. We refer the reader to the supplementary material for this extended version of the algorithm and for all the implementation details related to our experiments (architecture of models, optimization settings, hyper-parameters, etc.).
7. Discussion & Conclusion

Finally, is there a classifier that ensures optimal robustness against all adversarial attacks? We gave a negative answer to this question in the deterministic regime, but part of the question remains open when considering randomized algorithms. We demonstrated that randomized defenses are more efficient than deterministic ones, and devised a simple method to implement them.

Game theoretical point of view. There remains to study whether an Equilibrium exists in the Randomized regime. This question is appealing from a theoretical point of view, and requires to investigate the space of randomized Adversaries \( \mathcal{P}(\mathcal{F}_X)^2 \). The characterization of this space is not straightforward, and would require strong results in the theory of optimal transport. A possible research direction is to quotient the space \( \mathcal{F}_X^2 \) so as to simplify the search in \( \mathcal{P}(\mathcal{F}_X)^2 \) and the characterization of the Adversary’s best responses. The study of this equilibrium is tightly related to that of the value of the game, which would be interesting for obtaining min-max bounds on the accuracy under attack, as well as certificates of robustness for a set of classifiers.

Advocating for more provable defenses. Although the experimental results show that our mixture of classifiers outperforms Adversarial Training, our algorithm do not provide guarantees in terms of certified accuracy. As the literature on adversarial attacks and defenses demonstrated, better attacks always exist. This is why, more theoretical works need to be done to prove the robustness of a mixture created from this particular algorithm. More generally, our work advocates for the study of mixtures as a provable defense against adversarial attacks. One could, for example, build upon the connection between mixtures and noise injection to investigate a broader range of randomized strategies for the Defender, and devise certificates accordingly.

Improving Boosted Adversarial Training. From an algorithmic point of view, BAT can be improved in several ways. For instance, the weights can be learned while choosing the new classifier for the mixture. This could lead to an improved accuracy under attack, but would lack some theoretical justifications that still need to be set up. Finally, tighter connections with standard boosting algorithms could be established to improve the analysis of BAT.

Acknowledgements

We thank anonymous reviewers, whose comments helped us improve the paper significantly. We also thank Rida Laraki and Guillaume Carlier for fruitful discussions on Game theory as well as Alexandre Araujo for proof reading our experiments. This work was granted access to the HPC resources of IDRIS under the allocation 2020-101141 made by GENCI.

References


