



HAL
open science

Randomization matters How to defend against strong adversarial attacks

Rafael Pinot, Raphael Ettetdgui, Geovani Rizk, Yann Chevaleyre, Jamal Atif

► **To cite this version:**

Rafael Pinot, Raphael Ettetdgui, Geovani Rizk, Yann Chevaleyre, Jamal Atif. Randomization matters How to defend against strong adversarial attacks. Thirty-seventh International Conference on Machine Learning, Jul 2020, Vienna, Austria. hal-02892161

HAL Id: hal-02892161

<https://hal.science/hal-02892161>

Submitted on 7 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Randomization matters

How to defend against strong adversarial attacks

Rafael Pinot^{*12} Raphael Ettetdgui^{*1} Geovani Rizk¹ Yann Chevaleyre¹ Jamal Atif¹

Abstract

Is there a classifier that ensures optimal robustness against all adversarial attacks? This paper answers this question by adopting a game-theoretic point of view. We show that adversarial attacks and defenses form an *infinite* zero-sum game where classical results (*e.g.* 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 new algorithm for building randomized classifiers that are robust to *strong* adversarial attacks. Empirical results validate our theoretical analysis, and show that our defense method considerably outperforms Adversarial Training against state-of-the-art attacks.

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 vulnerability of machine learning models to adversarial attacks. Accordingly, 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). At the same time, there has been a growing interest in understand the very nature of this phenomenon (Fawzi et al., 2016; 2018; Bubeck et al., 2019; Ilyas et al., 2019; Gourdeau et al., 2019). Despite these significant efforts, the existence of a classifier with optimal robustness against all attacks remains an open problem. In this paper we tackle 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, 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, including Adversarial Training. 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 new algorithm that generates robust randomized classifiers using a boosting-type procedure. We evaluate this method, that we call Boosted Adversarial Training, in Section 6 against *strong* attacks on the CIFAR10 dataset. It outperforms Adversarial Training against both ℓ_∞ -PGD (Madry et al., 2018), and ℓ_2 -C&W attacks (Carlini & Wagner, 2017). More precisely, our algorithm achieves 0.58 (resp. 0.59) accuracy under attack against ℓ_∞ -PGD (resp. ℓ_2 -C&W) with 100 iterations, which is an improvement of 0.16 (resp. 0.08) over Adversarial Training.

^{*}Equal contribution ¹Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France ²Institut LIST, CEA, Université Paris-Saclay, France. Correspondence to: Rafael Pinot <rafael.pinot@dauphine.fr>, Raphael Ettetdgui <raphael.ettetdgui@polytechnique.edu>.

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. [Sinha et al. \(2018\)](#) address 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. A similar analysis was presented in [Lee & Raginsky \(2018\)](#) in a more general setting that does not focus on adversarial examples. Even though our work studies the same problem, our reasoning is very different. We adopt a game theoretic standpoint, which allows us to investigate randomized defenses and endow them with strong theoretical evidences.

Game Theory. Some works have tackled the problem of adversarial examples as a two player game. For example [Brückner & Scheffer \(2011\)](#) views adversarial example attacks and defenses as a Stackelberg game. More recently, [Rota Bulò et al. \(2017\)](#) and [Perdomo & Singer \(2019\)](#) investigated zero-sum games. They consider restricted versions of the game where classical theorems apply, such as when the defender only has a finite set of possible strategies. We study a more general setting. Finally, [Dhillon et al. \(2018\)](#) motivated the use of noise injection as a defense mechanism by game theoretic arguments but only present empirical results.

Randomization. Following the work of [Dhillon et al. \(2018\)](#) and [Xie et al. \(2018\)](#), several recent works studied noise injection as a defense mechanism. In particular, [Lecuyer et al. \(2018\)](#), followed by [Cohen et al. \(2019\)](#); [Li et al. \(2019\)](#); [Pinot et al. \(2019\)](#); [Wang et al. \(2019\)](#) demonstrated that noise injection can, in some cases, give provable defense against adversarial attacks. The analysis and algorithm we propose in this paper are not based on noise injection. However, a link could be made between these works and the mixture algorithm we propose, by noting that a classifier in which noise is being injected can be seen as an infinite mixture of perturbed classifiers.

Optimal transport. Our work considers a distributional setting, in which the Adversary manipulating the dataset is formalized by a push-forward measure. This kind of setting is close to optimal transport settings recently developed by [Bhagoji et al. \(2019\)](#) and [\(Pydi & Jog, 2019\)](#). Specifically, these works investigate classifier-agnostic lower bounds on the risk for binary classification under attack, with some hypothesis on the data distribution. Even though they do not treat the same problem, we believe that these works are profoundly related and complementary to ours.

3. Problem statement

Notations. For any set \mathcal{Z} with σ -algebra $\sigma(\mathcal{Z})$, if there is no ambiguity on the considered σ -algebra, we denote $\mathcal{P}(\mathcal{Z})$ the set of all probability measures over $(\mathcal{Z}, \sigma(\mathcal{Z}))$, and $\mathcal{F}_{\mathcal{Z}}$ the set of all measurable functions from $(\mathcal{Z}, \sigma(\mathcal{Z}))$ to $(\mathcal{Z}, \sigma(\mathcal{Z}))$. For $\mu \in \mathcal{P}(\mathcal{Z})$ and $\phi \in \mathcal{F}_{\mathcal{Z}}$, the *pushforward measure* of μ by ϕ is the measure $\phi\#\mu$ such that $\phi\#\mu(B) = \mu(\phi^{-1}(B))$ for any $B \in \sigma(\mathcal{Z})$.

Binary 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} , which is defined as the probability of misclassification:

$$\begin{aligned} \mathcal{R}(h) &:= \mathbb{E}_{(X,Y) \sim \mathcal{D}} [\mathbb{1}\{h(X) \neq Y\}] \\ &= \mathbb{E}_{Y \sim \nu} \left[\mathbb{E}_{X \sim \mu_Y} [\mathbb{1}\{h(X) \neq Y\}] \right]. \end{aligned} \quad (1)$$

Where $\mathcal{H} := \{h : x \mapsto \text{sgn } g(x) \mid g : \mathcal{X} \rightarrow \mathbb{R} \text{ continuous}\}$, $\nu \in \mathcal{P}(\mathcal{Y})$ is the probability measure that defines the law of the random variable Y , and for any $y \in \mathcal{Y}$, $\mu_y \in \mathcal{P}(\mathcal{X})$ is the conditional law of $X \mid (Y = y)$.

Adversarial example attack (point-wise). Given a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$ and a data sample $(x, y) \sim \mathcal{D}$, 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 τ to have a small norm, be it for the ℓ_{∞} or the ℓ_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 CIFAR10, typical values for ϵ_{∞} and ϵ_2 are respectively, 0.031 and 0.4/0.6/0.8.

Note that our definition of the problem implies that the Adversary has perfect information on the dataset and the classifier, so we consider here the strongest type of attacks, called *white-box attacks*. In particular, the same point x may have different perturbations depending on what its true class is: since the Adversary wants the classifier to do as many mistakes as possible, it may not attack a point when it is already misclassified.

Adversarial example attack (distributional). The Adversary is choosing, 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 ϕ_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 measure $\phi_y \# \mu_y$. The goal of the Adversary is thus to find $\phi = (\phi_{-1}, \phi_1) \in (\mathcal{F}_X)^2$ that maximizes the adversarial risk $\mathcal{R}_{\text{adv}}(h, \phi)$ defined as:

$$\mathcal{R}_{\text{adv}}(h, \phi) := \mathbb{E}_{Y \sim \nu} \left[\mathbb{E}_{X \sim \phi_Y \# \mu_Y} [\mathbb{1} \{h(X) \neq Y\}] \right] - \lambda \Omega(\phi). \quad (2)$$

The penalty function Ω represents the transportation cost of the Adversary, and $\lambda \in (0, 1)$ some regularization weight. More precisely, Ω encodes the constraints that the Adversary enforces on the attacks, to remain undetected.

Transportation costs. The analysis of Equation (2) will heavily depend on the choice of the penalty function Ω . In this paper, we study two types of penalties: the *norm penalty* Ω_{norm} , and the *mass penalty* Ω_{mass} . The first one penalizes the expected norm of the perturbation as follows:

$$\Omega_{\text{norm}}(\phi) := \mathbb{E}_{Y \sim \nu} \left[\mathbb{E}_{X \sim \mu_Y} [\|X - \phi_Y(X)\|_2 + \infty \mathbb{1} \{\|X - \phi_Y(X)\|_2 > \epsilon_2\}] \right], \quad (3)$$

with the convention $\infty \times 0 = 0$. The first term of the regularization has been proposed by [Carlini & Wagner \(2017\)](#) to compute the eponymous attack (C&W)¹. However, using this alone may lead to create perceptible attacks. To deal with this, but also for numerical experiments to be fair, [Carlini et al. \(2019\)](#) proposed to reject the perturbation when the ℓ_2 norm is greater than some threshold ϵ_2 , hence the second term².

We define the mass penalty as follows:

$$\Omega_{\text{mass}}(\phi) := \mathbb{E}_{Y \sim \nu} \left[\mathbb{E}_{X \sim \mu_Y} [\mathbb{1} \{X \neq \phi_Y(X)\} + \infty \mathbb{1} \{\|X - \phi_Y(X)\|_\infty > \epsilon_\infty\}] \right]. \quad (4)$$

The mass penalty discourages the Adversary from attacking too many points by penalizing the overall mass of transported points. This type of constraint, although not often discussed in the literature, is very important to understand and analyze real life scenarios, where the Defender tries to detect attacks based on the occurrences of its failures. Since the attack should also remain visually undetected, the penalty ensures that all attacks have an ℓ_∞ norm smaller than a given ϵ_∞ . This kind of ℓ_∞ constraint is related to other attacks from the literature (e.g. FGSM ([Goodfellow et al., 2015](#)) or PGD ([Madry et al., 2018](#))).

¹ Ω_{norm} is not limited to ℓ_2 norm. The results we present hold as long as the norm used to compare X and $\phi_Y(X)$ comes from a scalar product on \mathcal{X} .

²One could also use a threshold on the ℓ_∞ norm. We chose the ℓ_2 norm to be consistent with the first term of the penalty.

We will also see in Section 4 that the mass penalty can be useful to study approximate solutions for the norm penalty.

Adversarial defense, a two-player zero-sum game.

Whatever penalty we choose, its value does not depend on h (hence, the optimal Defender minimizing Eq. (1) or (2) is the same). The adversarial examples problem can thus be seen as a two-player zero-sum game, where the Defender tries to find the best possible hypothesis h , while a strong Adversary is manipulating the dataset distribution:

$$\inf_{h \in \mathcal{H}} \sup_{\phi \in (\mathcal{F}_X)^2} \mathcal{R}_{\text{adv}}(h, \phi). \quad (5)$$

This means that the Defender tries to design the classifier with the best performance under attack, whereas the Adversary will each time design the optimal attack on this specific classifier. In the game theoretical terminology, the choice of a classifier h (resp. an attack ϕ) for the Defender (resp. the Adversary) is called a *strategy*.

It is crucial to note that in our game, the sup-inf and inf-sup problems do not coincide. In this paper, we mainly focus on the Defender's point of view which corresponds to the inf-sup problem.

Definition 1 (Best Response). *Let $h \in \mathcal{H}$, and $\phi \in (\mathcal{F}_X)^2$. A best response from the Defender to ϕ is a classifier $h^* \in \mathcal{H}$ such that $\mathcal{R}_{\text{adv}}(h^*, \phi) = \min_{h \in \mathcal{H}} \mathcal{R}_{\text{adv}}(h, \phi)$. Similarly, a best response from the Adversary to h is an attack ϕ^* such that $\mathcal{R}_{\text{adv}}(h, \phi^*) = \max_{\phi \in (\mathcal{F}_X)^2} \mathcal{R}_{\text{adv}}(h, \phi)$.*

In the remaining, we denote $\text{BR}_\Omega(h)$ the set of all best responses of the Adversary to a classifier h , under penalty Ω . Similarly $\text{BR}(\phi)$ denotes the set of best responses to an attack ϕ . Since Ω does not impact the Defender's optimization problem, we omit to mention Ω .

Definition 2 (Pure Nash Equilibrium). *In the zero-sum game (Eq. 5) with penalty Ω , a Pure Nash Equilibrium is a couple of strategies $(h, \phi) \in \mathcal{H} \times (\mathcal{F}_X)^2$ such that*

$$\begin{cases} h & \in \text{BR}(\phi), \text{ and,} \\ \phi & \in \text{BR}_\Omega(h). \end{cases}$$

When it exists, a Pure Nash Equilibrium is a state of the game in which no player has any incentive to modify its strategy. In our setting, this simultaneously means that no attack could better fool the current classifier, and that the classifier is optimal for the current attack.

Remark. All the definitions in this section assume a deterministic regime, i.e. that neither the Defender nor the Adversary use randomization, hence the notion of *Pure Nash Equilibrium* in the game theory terminology. We will study this deterministic regime in Section 4. The randomized regime will be studied in Section 5.

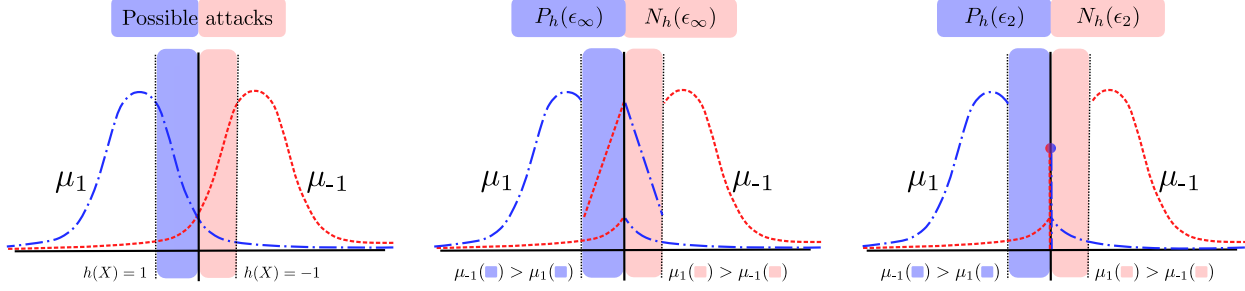


Figure 1. Comparison of the distributions μ_1 and μ_{-1} before attack (left) and after attack, according either to the mass penalty (middle) or the norm penalty (right). The blue (dotted and dashed) curve represents the distribution of $X|Y = 1$, and the red (dashed) one $X|Y = -1$. The black vertical line is the Bayes optimal classifier for the initial distributions. Finally, the red (lighter) and the blue (darker) areas represents the zones on which the Adversary can change the distributions for mass (middle) and norm (right) penalty.

4. Deterministic regime

Notations. Let $h \in \mathcal{H}$, $y \in \mathcal{Y}$, and $p \in \{2, \infty\}$. 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 $P_h^p(\delta) := \{x \in P_h \mid \exists z \in N_h \text{ and } \|z - x\|_p \leq \delta\}$, and in the same way $N_h^p(\delta)$ the points that are closer than δ from the other class. We omit p , when it is clear from the context³.

In this section we show that whatever penalty the Adversary has, no Pure Nash Equilibrium exists. To do so, we characterize the best responses for each player, and show that they can never satisfy Definition 2.

Adversary's best response. Let us first present the best responses of the Adversary under respectively the norm penalty and the mass penalty.

Lemma 1. Let $h \in \mathcal{H}$, $\epsilon_2 \leq 1$ the perceptibility parameter, and $\phi \in BR_{\Omega_{norm}}(h)$. Then the following holds:

$$\phi_1(x) = \begin{cases} \pi(x) & \text{if } x \in P_h(\epsilon_2) \\ x & \text{otherwise.} \end{cases}$$

Where π is the orthogonal projection on $(P_h)^c$, the complement of P_h in \mathcal{X} . ϕ_{-1} is characterized symmetrically.

Lemma 2. Let $h \in \mathcal{H}$, $\epsilon_\infty \leq 1$ the perceptibility parameter, and $\phi \in BR_{\Omega_{mass}}(h)$. Then the following holds:

$$\begin{cases} \phi_1(x) \in (P_h)^c & \text{if } x \in P_h(\epsilon_\infty) \\ \phi_1(x) = x & \text{otherwise.} \end{cases}$$

and ϕ_{-1} is characterized symmetrically.

Both best responses share a fundamental behavior: the optimal attack will only change points that are close enough to the decision boundary by it w.r.t ℓ_2 or ℓ_∞ norm. 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. This difference is illustrated in Figure 1 with two uni-dimensional Gaussian distributions. For the norm penalty (on the right), the part of μ_1 (dashed and dotted lines) that was in $P_h(\epsilon_2)$ (blue zone) is transported on a Dirac distribution at the decision boundary. The same holds for μ_{-1} (dashed lines). After attack, we now have $\mu_1(P_h(\epsilon_2)) = 0$, so a small value of μ_{-1} in $P_h(\epsilon_2)$ suffices to make it dominant, and that zone will now be classified -1 by the Bayes classifier.

For the norm penalty (in the middle), μ_1 (dashed and dotted lines) is set to 0 in $P_h(\epsilon_\infty)$ (blue zone), and this mass is transported into $N_h(\epsilon_\infty)$ (red zone), and added to the small amount of μ_1 that was already there. Similarly to the norm penalty case, the small value of μ_{-1} (dashed lines) in the $P_h(\epsilon_\infty)$ now suffices to make the best response classify that zone as -1.

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. In order to keep the attack computationally 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 (for a well chosen ϵ).

Defender's best response. At a first glance, one would suspect that the best response for the Defender ought to be the Bayes classifier for the transported distribution. However, the Bayes classifier is only well defined for distributions that have probability density functions. This does not always

³In particular when $\delta = \epsilon_p$ it depends on the ℓ_p norm.

hold here for the transported distribution. Typically, we see on Figure 1 that in the 1 dimensional setting, projecting on the decision boundary creates a Dirac distribution. Nevertheless, we show that there is a property, shared by the Bayes classifier when defined, that always holds for the Defender’s best response.

Lemma 3. *Let us consider $\phi \in (\mathcal{F}_X)^2$. If we take $h \in \text{BR}(\phi)$, 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 $(\nu, \phi_1 \# \mu_1, \phi_{-1} \# \mu_{-1})^4$. 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. 5) with penalty $\Omega \in \{\Omega_{mass}, \Omega_{norm}\}$, 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:

$$\sup_{\phi \in (\mathcal{F}_X)^2} \inf_{h \in \mathcal{H}} \mathcal{R}_{adv}(h, \phi) \leq \inf_{h \in \mathcal{H}} \sup_{\phi \in (\mathcal{F}_X)^2} \mathcal{R}_{adv}(h, \phi).$$

On the left side problem (sup-inf), the Adversary looks for the best attacking strategy ϕ against any *unknown* classifier. This is tightly related to the notion of *transferable attacks* (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), where 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:

$$\sup_{\phi \in (\mathcal{F}_X)^2} \inf_{h \in \mathcal{H}} \mathcal{R}_{adv}(h, \phi) < \inf_{h \in \mathcal{H}} \sup_{\phi \in (\mathcal{F}_X)^2} \mathcal{R}_{adv}(h, \phi).$$

This means that both problems are not equivalent. In particular, an attack designed to succeed against *any* classifier (i.e. a transferable attack) will not be as good as an attack tailored for a given classifier.

Consequence 2. *No deterministic defense (including Adversarial Training) may be proof against every attack.*

Let us consider the state-of-the-art defense which is Adversarial Training. The idea is to compute an efficient attack ϕ , 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 ϕ .

To be fully efficient, this method requires that ϕ 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 ϕ , there will always be a different optimal attack ϕ' that is efficient against h' . Hence Adversarial Training will never achieve a perfect defense.

5. Randomization matters

Fully randomized regime. We have shown, that since 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:

$$\inf_{\eta \in \mathcal{P}(\mathcal{H})} \sup_{\varphi \in \mathcal{P}((\mathcal{F}_X)^2)} \mathbb{E}_{\substack{h \sim \eta \\ \phi \sim \varphi}} [\mathcal{R}_{adv}(h, \phi)]. \quad (6)$$

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 in this setting. 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 6 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).

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 $n \in \mathbb{N}$, $\mathbf{h} = (h_1, \dots, h_n) \in \mathcal{H}^n$, and $\mathbf{q} \in \mathcal{P}([n])$. A mixed classifier of \mathbf{h} by \mathbf{q} is a mapping $m_{\mathbf{h}}^{\mathbf{q}}$ from \mathcal{X} to $\mathcal{P}(\mathcal{Y})$ such that for all $x \in \mathcal{X}$, $m_{\mathbf{h}}^{\mathbf{q}}(x)$ is the discrete probability distribution that is defined for all $y \in \mathcal{Y}$ by:*

$$m_{\mathbf{h}}^{\mathbf{q}}(x)(y) := \mathbb{E}_{i \sim \mathbf{q}} [\mathbb{1}\{h_i(x) = y\}].$$

⁴We prove this result in the supplementary material.

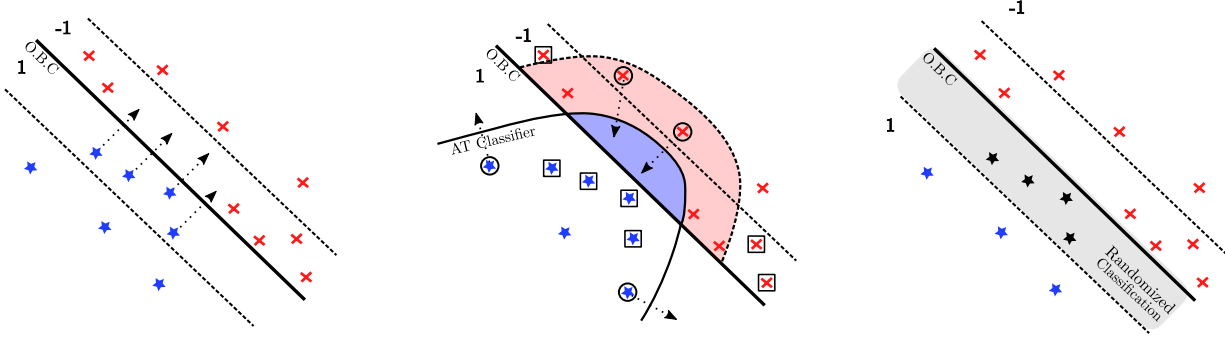


Figure 2. Illustration of adversarial examples (only on class 1 for more readability) crossing the decision boundary (left), adversarially trained classifier for the class 1 (middle), and a randomized classifier that defends class 1. Stars are natural examples for class 1, and crosses are natural examples for class -1. The straight line is the optimal Bayes classifier, and dashed lines delimit the points close enough to the boundary to be attacked resp. for class 1 and -1.

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 \mathbf{h} at random following the distribution \mathbf{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 \mathbf{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 $m_{\mathbf{h}}^{\mathbf{q}}$. It writes as follows:

$$\mathbb{E}_{Y \sim \nu} \left[\mathbb{E}_{X \sim \phi_Y \# \mu_Y} \left[\mathbb{E}_{\hat{Y} \sim m_{\mathbf{h}}^{\mathbf{q}}(X)} \left[\mathbb{1} \left\{ \hat{Y} \neq Y \right\} \right] \right] \right] - \lambda \Omega(\phi). \quad (7)$$

We also write \mathcal{R}_{adv} to mean Equation (7), 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.

To do so, let us demonstrate that the efficiency of any deterministic defense can be improved using a simple mixture algorithm. 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 rationale behind this idea is that, by construction, efficient attacks on one of these two classifiers will not work on the other. If we can then 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. We will thus need the following condition on the data distribution :

Definition 4 ((ϵ, p) -dilation and vanishing measure). *Let U be a subset of \mathcal{X} , ϵ a positive value, $p \in \{2, \infty\}$, and μ a*

probability measure.

1. The (ϵ, p) -dilation of U is defined as follows:

$$U \oplus^p \epsilon := \left\{ u + v \mid (u, v) \in U \times \mathcal{X} \text{ and } \|v\|_p \leq \epsilon \right\}.$$

2. We say that μ is (ϵ, p) -vanishing⁵ on U if we have:

$$\mu \left(U \oplus^p \epsilon \setminus U \right) \leq \mu(U).$$

This is because 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 \mathbf{q} . When Definition 4 applies on the attackable zones, it ensures that we gain more than we lose.

With these new definitions, we now can state our second main result: mixtures outperform deterministic classifiers.

Theorem 2. (*Randomization matters*) *Let $h_1 \in \mathcal{H}$, $\lambda \in (0, 1)$ the regularization parameter, $\phi \in \text{BR}_{\Omega_{\text{norm}}}(h_1)$, and $h_2 \in \text{BR}(\phi)$. If μ_1 (resp. μ_{-1}) is ϵ_2 -vanishing on $P_{h_1}(\epsilon_2)$ (resp. on $N_{h_1}(\epsilon_2)$), then for any $\alpha \in (\frac{1+\lambda\epsilon_2}{2}, 1)$ one has:*

$$\forall \phi' \in \text{BR}_{\Omega_{\text{norm}}}(m_{\mathbf{h}}^{\mathbf{q}}), \mathcal{R}_{\text{adv}}(m_{\mathbf{h}}^{\mathbf{q}}, \phi') < \mathcal{R}_{\text{adv}}(h_1, \phi).$$

Where $\mathbf{h} = (h_1, h_2)$, $\mathbf{q} = (\alpha, 1 - \alpha)$, and $m_{\mathbf{h}}^{\mathbf{q}}$ is the mixture of \mathbf{h} by \mathbf{q} . A similar result holds for the mass penalty, with $\alpha \in (\frac{1+\lambda}{2}, 1)$.

On the vanishing measure condition. Let us briefly explain this property. To defend against an attack, the general tactic is to change the classifier output, when points are

⁵As for P_h^p we omit p when it is clear from the context.

close to the border (either all the time, as in Adversarial Training where we move the decision boundary to incorporate adversarial examples, or part of the time as in our randomized algorithm so that the attack only works with a given probability).

For example on figure 2, we mix the Bayes classifier (left) with its optimal attack that swaps the blue and red zone between the dotted line, on the gray area that is the former attack zone for the blue class. This gives the figure on the right. If the first classifier has a weight $\alpha = 0.5$, the 10 old attacks (points between the dotted lines) now succeed only with probability 0.5 (the new optimal attack for star points being to leave them in place), whereas 3 new attacks are created (blue points outside of the gray area) that succeed with probability 0.5, for a total attack score of 6.5, which is lower than the old attack score of 10.

When adversarially training a classifier (Figure 2, middle), we change its output on the blue zone, so that four of the star points cannot be successfully attacked anymore. But in exchange, the dilation of this zone (in red) 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 red ones in the dilatation and 2 blue ones that are close to the new boundary) to be smaller than the number of attacks we prevent (the points that are in a square, 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.

This discussion shows that when no measure have any vanishing zone, Adversarial Training cannot bring any gain. By contraposition, whenever a deterministic classifier can be improved by Adversarial Training, it will also be outperformed under optimal attack by a randomized algorithm (see Theorem 2).

6. Experiments: How to build the mixture

Based on Theorem 2 we devise a new procedure (Algorithm 1), called Boosted Adversarial Training (BAT) to construct robust classifiers. It is based on three core principles: Adversarial Training, Boosting and Randomization.

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 algorithm 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.

Given a dataset D and a weight update parameter $\alpha \in [0, 1]$,

Algorithm 1 Boosted Adversarial Training

Input : n the number of classifiers, D the training data set and α the weight update parameter.

```

Create and adversarially train  $h_1$  on  $D$ 
 $\mathbf{h} = (h_1)$  ;  $\mathbf{q} = (1)$ 
for  $i = 2, \dots, n$  do
    Generate the adversarial data set  $\tilde{D}$  against  $m_{\mathbf{h}}^{\mathbf{q}}$ .
    Create and naturally train  $h_i$  on  $\tilde{D}$ 

     $q_k \leftarrow (1 - \alpha)q_k \quad \forall k \in [i - 1]$ 
     $q_i \leftarrow \alpha$ 

     $\mathbf{q} \leftarrow (q_1, \dots, q_i)$ 
     $\mathbf{h} \leftarrow (h_1, \dots, h_i)$ 
end
return  $m_{\mathbf{h}}^{\mathbf{q}}$ 

```

BAT starts by constructing an adversarially trained classifier on D , and gives it a weight of 1. Then, at each step of the algorithm, we train a new classifier on a data set \tilde{D} built from D that contains adversarial examples created to fool the current mixture. This new classifier is added to the mixture with a weight of α . Previous weights are then multiplied by $1 - \alpha$.

At each step, we use ℓ_∞ -PGD with 20 iterations and $\epsilon_\infty = 0.031$ to attack the current mixture and build the adversarial dataset \tilde{D} . We choose this attack to fairly compare against Adversarial Training, which uses it during the training procedure.

On evaluating against ℓ_∞ -PGD We use Expectation over Transformation (EOT) following (Athalye et al., 2018) and (Carlini et al., 2019), when implementing an ℓ_∞ -PGD attack against a mixture of classifier. Indeed, it is important to compute the expected loss over the mixture, so that the attack optimizes Equation (7). Previous works such as (Dhillon et al., 2018) and (Pinot et al., 2019) estimate the expected loss through a Monte Carlo sampling. Since we assume perfect information for the Adversary, it knows the exact distribution of the mixture. Hence it can directly compute the expected loss without using a sampling method.

We conducted a grid-search to evaluate the influence of α (see the supplementary material for more details). For the results we present here, the optimal α we found is equal to 0.06 for 10 classifiers. In Table 1 we compare the accuracy (on the CIFAR10 dataset (Krizhevsky & Hinton, 2009)) of Boosted and classical Adversarial Training under attack with ℓ_∞ -PGD run for 100 iterations.

Randomization matters

Training method	Natural	ℓ_∞ -PGD	ℓ_2 -C&W 0.4	ℓ_2 -C&W 0.6	ℓ_2 -C&W 0.8
Natural	0.88	0.00	0.00	0.00	0.00
(Madry et al., 2018)	0.83	0.42	0.67	0.60	0.51
BAT ($n = 10, \alpha = 0.06$)	0.80	0.58	0.70	0.65	0.59

Table 1. Evaluation on CIFAR10 without *data augmentation*. Accuracy under attack of a single classifier adversarially trained and the mixture formed with our Algorithm 1. The evaluation is made with ℓ_∞ -PGD and ℓ_2 -C&W attacks both computed with 100 steps. For ℓ_∞ -PGD we use an epsilon equal to $8/255$ (≈ 0.031), a step size equal to $2/255$ (≈ 0.008) and we allow random initialization. For ℓ_2 -C&W we use a learning rate equal to 0.1, 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 $\epsilon_2 \in \{0.4, 0.6, 0.8\}$. Since the mixture draws a classifier in \mathbf{h} according to \mathbf{q} to predict a class for each sample, we run 100 times the evaluation to compute the expected accuracy under attack of the mixture. The width of the 95% confidence interval is negligible (< 0.01). For this reason, we chose to omit it.

Results against ℓ_∞ -PGD. We compute 100 steps of ℓ_∞ -PGD for the attack at test time, while only 20 steps during the training. The idea behind this difference is that the Adversary may target only a few specific points, and so may have access to more computational power for attacks than the Defender that trains on the whole dataset. For a classifier to be fully robust, its loss of accuracy should be controlled when the attacks are strongest than what it was trained on.

As shown in Table 1, the mixture generated by BAT with 10 classifiers and $\alpha = 0.06$ outperforms adversarial training on all four attacks. This is already the case for 2 classifiers, which corroborates the result from Theorem 2. We refer the reader to the supplementary material for additional results on how the size of the mixture influences the performance.

On Evaluating against ℓ_2 -C&W. Adversarial Training can also be used to defend against ℓ_2 -C&W. We conducted experiments to evaluate whether the mixture constructed with BAT also outperforms it against this attack. Since the basic ℓ_2 -C&W attack creates an unbounded perturbation on examples, we implemented the constraint from Equation 3 by checking at test time whether the ℓ_2 -norm of the perturbation exceeds a certain threshold $\epsilon_2 \in \{0.4, 0.6, 0.8\}$. If this holds, we keep the natural example, instead of its adversary version.

For the attacks to be comparable, the radiuses of the balls must be chosen carefully. For CIFAR10, which is a $3 \times 32 \times 32$ dimensional space, this gives $\epsilon_2 = 0.8$ and $\epsilon_\infty = 0.03$. The results of this evaluation are presented in Table 1. Note that we ran 100 steps for the ℓ_2 -C&W as well.

Results against ℓ_2 -C&W. The accuracy under attack of our mixture is higher than that of adversarial training for all the thresholds. Our mixture is especially more robust than Adversarial Training when the threshold (*i.e. the budget for a perturbation*), is high. Here again, we see that with two classifiers the mixture already gives an accuracy under attack of 0.53 against ℓ_2 -C&W with $\epsilon_2 = 0.8$ and outperforms Adversarial Training. This result also corroborates Theorem 2.

We refer the reader to the supplementary material for all implementation details (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 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 an algorithm to implements them. 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. To do so, one could use an equivalence relation that matches two functions when they have the same adversarial risk.

The study of the mixed 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. One could also build upon the connection between mixtures and noise injection to investigate a broader range of randomized strategies for the Defender, and to devise such certificates.

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.

References

- Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 274–283, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.
- Bhagoji, A. N., Cullina, D., and Mittal, P. Lower bounds on adversarial robustness from optimal transport. In *Advances in Neural Information Processing Systems 32*, pp. 7496–7508. Curran Associates, Inc., 2019.
- Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.
- Brown, G. W. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.
- Brückner, M. and Scheffer, T. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’11, pp. 547–555, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450308137. doi:10.1145/2020408.2020495.
- Bubeck, S., Lee, Y. T., Price, E., and Razenshteyn, I. Adversarial examples from computational constraints. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 831–840, Long Beach, California, USA, 09–15 Jun 2019. PMLR.
- Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017.
- Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., and Madry, A. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.
- Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. *CoRR*, abs/1902.02918, 2019.
- Dhillon, G. S., Azzadenesheli, K., Bernstein, J. D., Kosai, J., Khanna, A., Lipton, Z. C., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018.
- Fawzi, A., Moosavi-Dezfooli, S.-M., and Frossard, P. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems 29*, pp. 1632–1640. Curran Associates, Inc., 2016.
- Fawzi, A., Fawzi, H., and Fawzi, O. Adversarial vulnerability for any classifier. In *Advances in Neural Information Processing Systems 31*, pp. 1186–1195. Curran Associates, Inc., 2018.
- Freund, Y. and Schapire, R. E. A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting. In Vitányi, P. M. B. (ed.), *Second European Conference on Computational Learning Theory (EuroCOLT-95)*, pp. 23–37, 1995. URL citeseer.nj.nec.com/freund95decisiontheoretic.html.
- Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.
- Gourdeau, P., Kanade, V., Kwiatkowska, M., and Worrell, J. On the hardness of robust classification. In *Advances in Neural Information Processing Systems 32*, pp. 7444–7453. Curran Associates, Inc., 2019.
- Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems 32*, pp. 125–136. Curran Associates, Inc., 2019.
- Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.
- Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 727–743, 2018.
- Lee, J. and Raginsky, M. Minimax statistical learning with wasserstein distances. In *Advances in Neural Information Processing Systems 31*, pp. 2687–2696. Curran Associates, Inc., 2018.
- Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems 32*, pp. 9459–9469. Curran Associates, Inc., 2019.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

- Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pp. 372–387. IEEE, 2016a.
- Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016b.
- Perdomo, J. C. and Singer, Y. Robust attacks against multiple classifiers. *CoRR*, abs/1906.02816, 2019.
- Pinot, R., Meunier, L., Araujo, A., Kashima, H., Yger, F., Gouy-Pailler, C., and Atif, J. Theoretical evidence for adversarial robustness through randomization. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*. 2019.
- Pydi, M. S. and Jog, V. Adversarial risk via optimal transport and optimal couplings, 2019.
- Rota Bulò, S., Biggio, B., Pillai, I., Pelillo, M., and Roli, F. Randomized prediction games for adversarial machine learning. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2466–2478, Nov 2017.
- Sinha, A., Namkoong, H., and Duchi, J. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.
- Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. The space of transferable adversarial examples. *arXiv*, 2017. URL <https://arxiv.org/abs/1704.03453>.
- Wang, B., Shi, Z., and Osher, S. Resnets ensemble via the feynman-kac formalism to improve natural and robust accuracies. In *Advances in Neural Information Processing Systems 32*, pp. 1655–1665. Curran Associates, Inc., 2019.
- Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.
- Zagoruyko, S. and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87.

Supplementary Material

1. Technical results

Notations: Let us suppose that \mathcal{X} is a normed vector space, with norm $\|\cdot\|$. $B_{\|\cdot\|}(x, \epsilon) = \{z \in \mathcal{X} \mid \|x - z\| \leq \epsilon\}$ is the closed ball of center x and radius ϵ for the norm $\|\cdot\|$. Note that $\mathcal{H} := \{h : x \mapsto \text{sgn } g(x) \mid g : \mathcal{X} \rightarrow \mathbb{R} \text{ continuous}\}$, with sgn the function that outputs 1 if $g(x) > 0$, -1 if $g(x) < 0$, and 0 otherwise. Hence for any $(x, y) \sim D$, and $h \in \mathcal{H}$ one has $\mathbb{1}\{h(x) \neq y\} = \mathbb{1}\{g(x)y \leq 0\}$.

Lemma 1. *Let $h \in \mathcal{H}$, $\epsilon_2 \leq 1$ the perceptibly parameter, and $\phi \in BR_{\Omega_{\text{norm}}}(h)$. Then the following holds:*

$$\phi_1(x) = \begin{cases} \pi(x) & \text{if } x \in P_h(\epsilon_2) \\ x & \text{otherwise.} \end{cases}$$

Where π is the orthogonal projection on P_h^{\complement} , the complement of P_h in \mathcal{X} . ϕ_{-1} is characterized symmetrically.

Proof. Ω_{norm} is defined with an ℓ_2 norm, but this result holds as long as \mathcal{X} is an Hilbert space with dot product $\langle \cdot | \cdot \rangle$ and associated norm $\|\cdot\| = \sqrt{\langle \cdot | \cdot \rangle}$. We demonstrate the result with these general notations.

Let us first simplify the worst case adversarial risk for h . Recall that $h = \text{sgn}(g)$ with g continuous. From the definition of adversarial risk we have:

$$\sup_{\phi \in (\mathcal{F}_{\mathcal{X}})^2} \mathcal{R}_{\text{adv}}(h, \phi) = \sup_{\phi \in (\mathcal{F}_{\mathcal{X}})^2} \sum_{y=\pm 1} \nu_y \mathbb{E}_{X \sim \mu_y} [\mathbb{1}\{h(\phi_y(X)) \neq y\} - \lambda \|X - \phi_y(X)\| - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\}] \quad (8)$$

$$= \sup_{\phi \in (\mathcal{F}_{\mathcal{X}})^2} \sum_{y=\pm 1} \nu_y \mathbb{E}_{X \sim \mu_y} [\mathbb{1}\{g(\phi_y(X))y \leq 0\} - \lambda \|X - \phi_y(X)\| - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\}] \quad (9)$$

$$= \sum_{y=\pm 1} \nu_y \sup_{\phi_y \in \mathcal{F}_{\mathcal{X}}} \mathbb{E}_{X \sim \mu_y} [\mathbb{1}\{g(\phi_y(X))y \leq 0\} - \lambda \|X - \phi_y(X)\| - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\}] \quad (10)$$

Since finding ϕ_1 and ϕ_{-1} are two independent optimization problems, hereafter, we focus on characterizing ϕ_1 (i.e. $y = 1$).

$$\sup_{\phi_1 \in \mathcal{F}_{\mathcal{X}}} \mathbb{E}_{X \sim \mu_1} [\mathbb{1}\{g(\phi_1(X)) \leq 0\} - \lambda \|X - \phi_1(X)\| - \infty \mathbb{1}\{\|X - \phi_1(X)\| > \epsilon_2\}] \quad (11)$$

$$= \mathbb{E}_{X \sim \mu_1} \left[\text{essup}_{z \in B_{\|\cdot\|}(X, \epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \|X - z\| \right] \quad (12)$$

$$= \int_{\mathcal{X}} \text{essup}_{z \in B_{\|\cdot\|}(x, \epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \|x - z\| \, d\mu_1(x). \quad (13)$$

Let us now consider $(H_j)_{j \in J}$ a partition of \mathcal{X} , we can write.

$$\sup_{\phi_1 \in \mathcal{F}_{\mathcal{X}}} \mathbb{E}_{X \sim \mu_1} [\mathbb{1}\{g(\phi_1(X)) \leq 0\} - \lambda \|X - \phi_1(X)\| - \infty \mathbb{1}\{\|X - \phi_1(X)\| > \epsilon_2\}] \quad (14)$$

$$= \sum_{j \in J} \int_{H_j} \text{essup}_{z \in B_{\|\cdot\|}(x, \epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \|x - z\| \, d\mu_1(x) \quad (15)$$

In particular, we consider here $H_0 = P_h^{\complement}$, $H_1 = P_h \setminus P_h(\epsilon_2)$, and $H_2 = P_h(\epsilon_2)$.

• **For** $x \in H_0 = P_h^{\mathbb{C}}$, taking $z = x$ we get $\mathbb{1}\{g(z) \leq 0\} - \lambda\|x - z\| = 1$. Since for any $z \in \mathcal{X}$ we have $\mathbb{1}\{g(z) \leq 0\} - \lambda\|x - z\| \leq 1$, this strategy is optimal. Furthermore, for any other optimal strategy z' , we would have $\|x - z'\| = 0$, hence $z' = x$, and an optimal attack will never move the points of $H_0 = P_h^{\mathbb{C}}$.

• **For** $x \in H_1 = P_h \setminus P_h(\epsilon_2)$, we have $B_{\|\cdot\|}(x, \epsilon_2) \subset P_h$ by definition of $P_h(\epsilon_2)$. Hence, for any $z \in B_{\|\cdot\|}(x, \epsilon_2)$, one gets $g(z) > 0$. Then $\mathbb{1}\{g(z) \leq 0\} - \lambda\|x - z\| \leq 0$. The only optimal z will thus be $z = x$, giving value 0.

• **Let us now consider** $x \in H_2 = P_h(\epsilon_2)$, which is the interesting case where an attack is possible. We know that $B_{\|\cdot\|}(x, \epsilon_2) \cap P_h^{\mathbb{C}} \neq \emptyset$, and for any z in this intersection, $\mathbb{1}\{g(z) \leq 0\} = 1$. Hence :

$$\operatorname{essup}_{z \in B_{\|\cdot\|}(x, \epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda\|x - z\| = \max(1 - \lambda \operatorname{essinf}_{z \in B_{\|\cdot\|}(x, \epsilon_2) \cap P_h^{\mathbb{C}}} \|x - z\|, 0) \quad (16)$$

$$= \max(1 - \lambda \pi_{B_{\|\cdot\|}(x, \epsilon_2) \cap P_h^{\mathbb{C}}}(x), 0) \quad (17)$$

Where $\pi_{B_{\|\cdot\|}(x, \epsilon_2) \cap P_h^{\mathbb{C}}}$ is the projection on the closure of $B_{\|\cdot\|}(x, \epsilon_2) \cap P_h^{\mathbb{C}}$. Note that $\pi_{B_{\|\cdot\|}(x, \epsilon_2) \cap P_h^{\mathbb{C}}}$ exists: g is continuous, so $B_{\|\cdot\|}(x, \epsilon_2) \cap P_h^{\mathbb{C}}$ is a closed set, bounded, and thus compact, since we are in finite dimension. The projection is however not guaranteed to be unique since we have no evidence on the convexity of the set. Finally, let us remark that, since $\lambda \in (0, 1)$, and $\epsilon_2 \leq 1$, one has $1 - \lambda \pi_{B_{\|\cdot\|}(x, \epsilon_2) \cap P_h^{\mathbb{C}}}(x) \geq 0$ for any $x \in H_2$. Hence, on $P_h(\epsilon_2)$, the optimal attack projects all the points on the decision boundary. For simplicity, and since there is no ambiguity, we write the projection π .

Finally, since $H_0 \cup H_1 \cup H_2 = \mathcal{X}$, Lemma 1 holds. Furthermore, the score for this optimal attack is:

$$\sup_{\phi \in (\mathcal{F}_{\mathcal{X}})^2} \mathcal{R}_{\text{adv}}(h, \phi) = \sum_{y=\pm 1} \nu_y \sum_{j \in J_{H_j}} \int \operatorname{essup}_{z \in B_{\|\cdot\|}(x, \epsilon_2)} \mathbb{1}\{g(z)y \leq 0\} - \lambda\|x - z\| d\mu_y(x) \quad (18)$$

Since the value is 0 on $P_h \setminus P_h(\epsilon_2)$ (resp. on $N_h \setminus N_h(\epsilon_2)$) for ϕ_1 (resp. ϕ_{-1}), one gets:

$$= \nu_1 \left[\int_{P_h(\epsilon_2)} (1 - \lambda \pi(x)) d\mu_1(x) + \int_{P_h^{\mathbb{C}}} 1 d\mu_1(x) \right] + \nu_{-1} \left[\int_{N_h(\epsilon_2)} (1 - \lambda \pi(x)) d\mu_{-1}(x) + \int_{N_h^{\mathbb{C}}} 1 d\mu_{-1}(x) \right] \quad (19)$$

$$= \nu_1 \left[\int_{P_h(\epsilon_2)} (1 - \lambda \pi(x)) d\mu_1(x) + \mu_1(P_h^{\mathbb{C}}) \right] + \nu_{-1} \left[\int_{N_h(\epsilon_2)} (1 - \lambda \pi(x)) d\mu_{-1}(x) + \mu_{-1}(N_h^{\mathbb{C}}) \right] \quad (20)$$

$$= \mathcal{R}(h) + \nu_1 \int_{P_h(\epsilon_2)} (1 - \lambda \pi(x)) d\mu_1(x) + \nu_{-1} \int_{N_h(\epsilon_2)} (1 - \lambda \pi(x)) d\mu_{-1}(x) \quad (21)$$

Since $\mathcal{R}(h) = \mathbb{P}(h(X) \neq Y) \mathbb{P}(g(X)Y \leq 0) = \nu_1 \mu_1(P_h^{\mathbb{C}}) + \nu_{-1} \mu_{-1}(N_h^{\mathbb{C}})$.

This provides an interesting decomposition of the adversarial risk into the risk without attack and the loss on the attack zone. \square

Lemma 2. Let $h \in \mathcal{H}$, $\epsilon_\infty \leq 1$ the perceptibility parameter, and $\phi \in BR_{\Omega_{\text{mass}}}(h)$. Then the following holds:

$$\begin{cases} \phi_1(x) \in P_h^{\mathbb{C}} & \text{if } x \in P_h(\epsilon_\infty) \\ \phi_1(x) = x & \text{otherwise.} \end{cases}$$

and ϕ_{-1} is characterized symmetrically.

Proof. Following the same proof schema as before the adversarial risk writes as follows:

$$\sup_{\phi \in (\mathcal{F}_X)^2} \mathcal{R}_{\text{adv}}(h, \phi) = \sup_{\phi \in (\mathcal{F}_X)^2} \sum_{y=\pm 1} \nu_y \mathbb{E}_{X \sim \mu_y} [\mathbb{1}\{h(\phi_y(X)) \neq y\} - \lambda \mathbb{1}\{X \neq \phi_y(X)\} - \infty \mathbb{1}\{\|X - \phi_y(X)\|_\infty > \epsilon_\infty\}] \quad (22)$$

$$= \sup_{\phi \in (\mathcal{F}_X)^2} \sum_{y=\pm 1} \nu_y \mathbb{E}_{X \sim \mu_y} [\mathbb{1}\{g(\phi_y(X)) y \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_y(X)\} - \infty \mathbb{1}\{\|X - \phi_y(X)\|_\infty > \epsilon_\infty\}] \quad (23)$$

$$= \sum_{y=\pm 1} \nu_y \sup_{\phi_y \in \mathcal{F}_X} \mathbb{E}_{X \sim \mu_y} [\mathbb{1}\{g(\phi_y(X)) y \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_y(X)\} - \infty \mathbb{1}\{\|X - \phi_y(X)\|_\infty > \epsilon_\infty\}] \quad (24)$$

Since finding ϕ_1 and ϕ_{-1} are two independent optimization problem, hereafter, we focus on characterizing ϕ_1 (i.e. $y = 1$).

$$\sup_{\phi_1 \in \mathcal{F}_X} \mathbb{E}_{X \sim \mu_1} [\mathbb{1}\{g(\phi_1(X)) \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_1(X)\} - \infty \mathbb{1}\{\|X - \phi_1(X)\|_\infty > \epsilon_\infty\}] \quad (25)$$

$$= \mathbb{E}_{X \sim \mu_1} \left[\operatorname{essup}_{z \in B_{\|\cdot\|_\infty}(X, \epsilon_\infty)} \mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_1(X)\} \right] \quad (26)$$

$$= \int_{\mathcal{X}} \operatorname{essup}_{z \in B_{\|\cdot\|_\infty}(x, \epsilon_\infty)} \mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{x \neq \phi_1(x)\} d\mu_1(x). \quad (27)$$

Let us now consider $(H_j)_{j \in J}$ a partition of \mathcal{X} , we can write.

$$\sup_{\phi_1 \in \mathcal{F}_X} \mathbb{E}_{X \sim \mu_1} [\mathbb{1}\{g(\phi_1(X)) \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_1(X)\} - \infty \mathbb{1}\{\|X - \phi_1(X)\|_\infty > \epsilon_\infty\}] \quad (28)$$

$$= \sum_{j \in J} \int_{H_j} \operatorname{essup}_{z \in B_{\|\cdot\|_\infty}(x, \epsilon_\infty)} \mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{x \neq \phi_1(x)\} d\mu_1(x) \quad (29)$$

In particular, we can take $H_0 = P_h^{\mathbb{G}}$, $H_1 = P_h \setminus P_h(\epsilon_2)$, and $H_2 = P_h(\epsilon_2)$.

For $x \in H_0 = P_h^{\mathbb{G}}$ or $x \in H_1 = P_h \setminus P_h(\epsilon_2)$, with the same reasoning as before, any optimal attack will choose $\phi_1(x) = x$.

Let $x \in H_2 = P_h(\epsilon_2)$. We know that $B_{\|\cdot\|_\infty}(x, \epsilon_\infty) \cap P_h^{\mathbb{G}} \neq \emptyset$, and for any z in this intersection, one has $g(z) \leq 0$ and $z \neq x$. Hence $\operatorname{essup}_{z \in B_{\|\cdot\|_\infty}(x, \epsilon_\infty)} \mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{z \neq x\} = \max(1 - \lambda, 0)$. Since $\lambda \in (0, 1)$ one has $\mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{z \neq x\} =$

$1 - \lambda$ for any $z \in B_{\|\cdot\|_\infty}(x, \epsilon_\infty) \cap P_h^{\mathbb{G}}$. Then any function that given a $x \in \mathcal{X}$ outputs $\phi_1(x) \in B_{\|\cdot\|_\infty}(x, \epsilon_\infty) \cap P_h^{\mathbb{G}}$ is optimal on H_2 .

Finally, since $H_0 \cup H_1 \cup H_2 = \mathcal{X}$, Lemma 2 holds. □

Lemma 3. Let us consider $\phi \in (\mathcal{F}_X)^2$. If we take $h \in \text{BR}(\phi)$, 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 $y \in \mathcal{Y}$, $X | (Y = y) \sim \phi_y \# \mu_y$.

Proof. We reason ad absurdum. Let us consider $y = 1$, the proof for $y = -1$ is symmetrical. Let us suppose that there exists $C \subset P_h$ such that $\nu_{-1} \phi_{-1} \# \mu_{-1}(C) > \nu_1 \phi_1 \# \mu_1(C)$. We can then construct h_1 as follows:

$$h_1(x) = \begin{cases} h(x) & \text{if } x \notin C \\ -1 & \text{otherwise.} \end{cases}$$

Since h and h_1 are identical outside C , the difference between the adversarial risks of h and h_1 writes as follows:

$$\mathcal{R}_{\text{adv}}(h, \phi) - \mathcal{R}_{\text{adv}}(h_1, \phi) = \sum_{y=\pm 1} \nu_y \int_C (\mathbb{1}\{h(x) \neq y\} - \mathbb{1}\{h_1(x) \neq y\}) d(\phi_y \# \mu_y)(x) \quad (30)$$

$$= \nu_{-1} \mathbb{1}\{h(x) = 1\} \phi_{-1} \# \mu_{-1}(C) - \nu_1 \mathbb{1}\{h_1(x) \neq 1\} \phi_1 \# \mu_1(C) \quad (31)$$

$$= \nu_{-1} \phi_{-1} \# \mu_{-1}(C) - \nu_1 \phi_1 \# \mu_1(C) \quad (32)$$

Since by hypothesis $\nu_{-1} \phi_{-1} \# \mu_{-1}(C) > \nu_1 \phi_1 \# \mu_1(C)$ the difference between the adversarial risks of h and h_1 is strictly positive. This means that h_1 gives strictly better adversarial risk than the best response h . Since, by definition h is supposed to be optimal, this leads to a contradiction. Hence Lemma 3 holds. \square

Additional Result. *Let us assume that there is a probability measure ζ that dominates both $\phi_1 \# \mu_1$ and $\phi_{-1} \# \mu_{-1}$. Let us consider $\phi \in (\mathcal{F}_X)^2$. If we take $h \in \text{BR}(\phi)$, then h is the Bayes optimal classifier for the distribution $(\nu, \phi_1 \# \mu_1, \phi_{-1} \# \mu_{-1})$.*

Proof. For simplicity, we denote $f_1 = \frac{d(\phi_1 \# \mu_1)}{d\zeta}$ and $f_{-1} = \frac{d(\phi_{-1} \# \mu_{-1})}{d\zeta}$ the Radon-Nikodym derivatives of $\phi_1 \# \mu_1$ and $\phi_{-1} \# \mu_{-1}$ w.r.t. ζ . The best response h minimizes adversarial risk under attack ϕ . This minimal risk writes:

$$\inf_{h \in \mathcal{H}} \mathcal{R}_{\text{adv}}(h, \phi) = \inf_{h \in \mathcal{H}} \sum_{y=\pm 1} \nu_y \mathbb{E}_{x \sim \mu_y} [\mathbb{1}\{h(\phi_y(x)) \neq y\}] - \lambda \Omega(\phi) \quad (33)$$

Since the the penalty function does not depend on h , it suffices to seek $\inf_{h \in \mathcal{H}} \sum_{y=\pm 1} \nu_y \int_X \mathbb{1}\{h(x) \neq y\} d(\phi_y \# \mu_y)(x)$.

Moreover thanks to the transfer theorem, one gets the following:

$$\inf_{h \in \mathcal{H}} \sum_{y=\pm 1} \nu_y \int_X \mathbb{1}\{h(x) \neq y\} d(\phi_y \# \mu_y)(x) = \inf_{h \in \mathcal{H}} \sum_{y=\pm 1} \nu_y \int_X \mathbb{1}\{h(x) \neq y\} f_y(x) d\zeta(x) \quad (34)$$

$$= \inf_{h \in \mathcal{H}} \int_X \sum_{y=\pm 1} \nu_y \mathbb{1}\{h(x) \neq y\} f_y(x) d\zeta(x). \quad (35)$$

Finally, since the integral is bounded we get:

$$\inf_{h \in \mathcal{H}} \int_X \sum_{y=\pm 1} \nu_y \mathbb{1}\{h(x) \neq y\} f_y(x) d\zeta(x) = \int_X \left[\inf_{h \in \mathcal{H}} \sum_{y=\pm 1} \nu_y \mathbb{1}\{h(x) \neq y\} f_y(x) \right] d\zeta(x). \quad (36)$$

Hence, the best response h is such that for every $x \in X$, and $y \in \mathcal{Y}$, one has $h(x) = y$ if and only if $f_y(x) \leq f_{-y}(x)$. Thus, h is the optimal Bayes classifier for the distribution $(\nu, \phi_1 \# \mu_1, \phi_{-1} \# \mu_{-1})$. Furthermore, 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 $y \in \mathcal{Y}$, $X | (Y = y) \sim \phi_y \# \mu_y$. \square

Theorem 1 (Non-existence of a pure Nash equilibrium). *In our zero-sum game with penalty $\Omega \in \{\Omega_{\text{mass}}, \Omega_{\text{norm}}\}$, there is no Pure Nash Equilibrium.*

Proof. Let h be a classifier, $\phi \in \text{BR}_\Omega(h)$ an optimal attack against h . We will show that $h \notin \text{BR}(\phi)$, i.e. that h does not satisfy the condition from Lemma 3. This suffices for Theorem 1 to hold since it implies that there is no $(h, \phi) \in \mathcal{H} \times (\mathcal{F}_X)^2$ such that $h \in \text{BR}(\phi)$ and $\phi \in \text{BR}_\Omega(h)$.

According to Lemmas 1 and 2, whatever penalty we use, there exists $p \in \{2, \infty\}$, and $\delta > 0$ such that $\phi_1 \# \mu_1(P_h^p(\delta)) = 0$ or $\phi_{-1} \# \mu_{-1}(N_h^p(\delta)) = 0^6$. Both cases are symmetrical, so let us assume that $P_h^p(\delta)$ is of null measure for the transported

⁶ $p = 2$ for Ω_{norm} and $p = \infty$ for Ω_{mass} .

distribution conditioned by $y = 1$. Furthermore we have $\phi_{-1}\#\mu_{-1}(P_h^p(\delta)) = \mu_{-1}(P_h^p(\delta)) > 0$ since ϕ_{-1} is the identity function on $P_h^p(\delta)$, and since μ_{-1} is of full support on \mathcal{X} . Hence we get the following:

$$\phi_{-1}\#\mu_{-1}(P_h^p(\delta)) > \phi_1\#\mu_1(P_h^p(\delta)). \quad (37)$$

Since the right side of the inequality is null, we also get:

$$\phi_{-1}\#\mu_{-1}(P_h^p(\delta))\nu_{-1} > \phi_1\#\mu_1(P_h^p(\delta))\nu_1 \quad (38)$$

This inequality is incompatible with the characterization of best response for the Defender of Lemma 3. Hence $h \notin \text{BR}(\phi)$. \square

Theorem 2. (Randomization matters) Let $h_1 \in \mathcal{H}$, $\lambda \in (0, 1)$ the regularization parameter, $\phi \in \text{BR}_{\Omega_{\text{norm}}}(h_1)$, and $h_2 \in \text{BR}(\phi)$. If μ_1 (resp. μ_{-1}) is ϵ_2 -vanishing on $P_{h_1}(\epsilon_2)$ (resp. on $N_{h_1}(\epsilon_2)$), then for any $\alpha \in (\frac{1+\lambda\epsilon_2}{2}, 1)$ one has:

$$\forall \phi' \in \text{BR}_{\Omega_{\text{norm}}}(m_{\mathbf{h}}^{\mathbf{q}}), \mathcal{R}_{\text{adv}}(m_{\mathbf{h}}^{\mathbf{q}}, \phi') < \mathcal{R}_{\text{adv}}(h_1, \phi).$$

Where $\mathbf{h} = (h_1, h_2)$, $\mathbf{q} = (\alpha, 1 - \alpha)$, and $m_{\mathbf{h}}^{\mathbf{q}}$ is the mixture of \mathbf{h} by \mathbf{q} . A similar result holds for the mass penalty, with $\alpha \in (\frac{1+\lambda}{2}, 1)$.

Proof. Here we consider Ω_{norm} but the proof is similar for Ω_{mass} . To demonstrate Theorem 2, we actually show a more general result, where we only need μ_1 to be ϵ_2 -vanishing on some $U \subset P_{h_1}(\epsilon_2)$. In particular this will be true when $U = P_{h_1}(\epsilon_2)$. Let us assume that such an U exists.

We can construct h_2 as follows:

$$h_2(x) = \begin{cases} -h_1(x) & \text{if } x \in U \\ h_1(x) & \text{otherwise.} \end{cases}$$

This means that h_2 changes the class of all points in U , and do not change the rest, compared to h_1 . Let $\alpha \in (0, 1)$, and the corresponding $m_{\mathbf{h}}^{\mathbf{q}}$, and $\phi' \in \text{BR}_{\Omega_{\text{norm}}}(m_{\mathbf{h}}^{\mathbf{q}})$. We will find a condition on α so that the score of $m_{\mathbf{h}}^{\mathbf{q}}$ is lower than the score of h_1 .

$$\mathcal{R}_{\text{adv}}(m_{\mathbf{h}}^{\mathbf{q}}, \phi') = \sum_{y=\pm 1} \nu_y \int_{\mathcal{X}} \text{essup}_{z \in B_{\|\cdot\|}(x, \epsilon)} \alpha \mathbb{1}\{h_1(z) \neq y\} + (1 - \alpha) \mathbb{1}\{h_2(z) \neq y\} - \lambda \|x - z\| d\mu_y(x) \quad (39)$$

The only terms that may vary between the score of h_1 and the score of $m_{\mathbf{h}}^{\mathbf{q}}$ are the integrals on U , $U \oplus \epsilon_2$ and $\phi_{-1}^{-1}(U)$ (inverse image of U by ϕ_{-1}), respectively the points we mix on, the points that may become attackable when $y = 1$ by moving them on U , and the ones that were attacked for $y = -1$ by moving them on U . Hence, for simplicity, we only write those terms in the following. Let us first consider the score of h_1 under optimal attack. Thanks to the analysis of the Lemma 1, it writes:

$$\sup_{\phi \in (\mathcal{F}_{\mathcal{X}})^2} \mathcal{R}_{\text{adv}}(h_1, \phi) = \nu_1 \int_U (1 - \lambda \|x - \pi_{P_{h_1}^c}(x)}\|) d\mu_1(x) + \nu_{-1} \mu_{-1}(U) \quad (40)$$

$$+ \nu_{-1} \mu_{-1}(U \oplus \epsilon_2 \setminus P_{h_1}(\epsilon_2)) + \nu_1 \int_{(U \oplus \epsilon_2 \setminus U) \setminus P_{h_1}(\epsilon_2)} 0 d\mu_1(x) \quad (41)$$

$$+ \nu_{-1} \mu_{-1}(U \oplus \epsilon_2 \cap P_{h_1}(\epsilon_2)) + \nu_1 \int_{(U \oplus \epsilon_2 \setminus U) \cap P_{h_1}(\epsilon_2)} (1 - \lambda \|x - \pi_{P_{h_1}^c}(x)}\|) d\mu_1(x) \quad (42)$$

$$+ \nu_{-1} \int_{\phi_{-1}^{-1}(U)} (1 - \lambda \|x - \pi_U(x)\|) d\mu_{-1}(x). \quad (43)$$

For $y = 1$ all points in $P_{h_1}(\epsilon_2)$ are attacked by projecting on the decision boundary, and no point outside is attacked. For $y = -1$ some points of $N_{h_1}(\epsilon_2)$, that are attacked, may be sent into U , and others may not. Now let us consider the score of the mixture under its optimal attack.

$$\sup_{\phi \in (\mathcal{F}_X)^2} \mathcal{R}_{\text{adv}}(m_{\mathbf{h}}^{\mathbf{q}}, \phi) = \nu_1 \int_U \max \left(1 - \alpha, \alpha - \lambda \|x - \pi_{P_{h_1}^{\mathbf{c}}}(x)}\| \right) d\mu_1(x) \quad (44)$$

$$+ \nu_{-1} \int_U \max \left(\alpha, 1 - \alpha - \lambda \|x - \pi_{U \oplus \epsilon_2 \setminus U}(x)}\| \right) d\mu_{-1}(x) \quad (45)$$

$$+ \nu_1 \int_{(U \oplus \epsilon_2 \setminus U) \cap P_{h_1}(\epsilon_2)} \max \left(1 - \alpha - \lambda \|x - \pi_U(x)\|, 1 - \lambda \|x - \pi_{P_{h_1}^{\mathbf{c}}}(x)}\| \right) d\mu_1(x) \quad (46)$$

$$+ \nu_{-1} \mu_{-1}((U \oplus \epsilon_2 \setminus U) \cap P_{h_1}(\epsilon_2)) + \nu_1 \int_{(U \oplus \epsilon_2 \setminus U) \setminus P_{h_1}(\epsilon_2)} \max(0, 1 - \alpha - \lambda \|x - \pi_U(x)\|) d\mu_1(x) \quad (47)$$

$$+ \nu_{-1} \mu_{-1}((U \oplus \epsilon_2 \setminus U) \setminus P_{h_1}(\epsilon_2)) + \nu_{-1} \int_{\phi_{-1}^{-1}(U)} \max \left(0, 1 - \lambda \|x - \pi_{N_{h_1}^{\mathbf{c}}}(x)}\|, \alpha - \lambda \|x - \pi_U(x)\| \right) d\mu_{-1}(x) \quad (48)$$

We need to take into account the special case of the points in the dilation that were already in the attacked zone before, and that can now be attacked in two ways, either by projecting on U (but that works with probability α , since the classification on U is now randomized) or by projecting on $P_{h_1}^{\mathbf{c}}$, which works with probability 1 but may use more distance and so pay more penalty. For $y = -1$, attacks on U now work with probability α instead of 1, so the attacker may choose to attack on other points instead, even if that takes more distance.

We can now compute the difference between both risks, and show that it is strictly positive:

$$\Delta \mathcal{R}_{\text{adv}} = \sup_{\phi \in (\mathcal{F}_X)^2} \mathcal{R}_{\text{adv}}(h_1, \phi) - \sup_{\phi \in (\mathcal{F}_X)^2} \mathcal{R}_{\text{adv}}(m_{\mathbf{h}}^{\mathbf{q}}, \phi) \quad (49)$$

$$> \nu_1 \int_U 1 - \lambda \|x - \pi_{P_{h_1}^{\mathbf{c}}}(x)}\| - \max \left(1 - \alpha, \alpha - \lambda \|x - \pi_{P_{h_1}^{\mathbf{c}}}(x)}\| \right) d\mu_1(x) \quad (50)$$

$$+ \nu_{-1} \mu_{-1}(U) - \nu_{-1} \int_U \max \left(\alpha, 1 - \alpha - \lambda \|x - \pi_{U \oplus \epsilon_2 \setminus U}(x)}\| \right) d\mu_{-1}(x) \quad (51)$$

$$+ \nu_1 \int_{(U \oplus \epsilon_2 \setminus U) \cap P_{h_1}(\epsilon_2)} 1 - \lambda \|x - \pi_{P_{h_1}^{\mathbf{c}}}(x)}\| - \max \left(1 - \alpha - \lambda \|x - \pi_U(x)\|, 1 - \lambda \|x - \pi_{P_{h_1}^{\mathbf{c}}}(x)}\| \right) d\mu_1(x) \quad (52)$$

$$+ \nu_{-1} \int_{\phi_{-1}^{-1}(U)} 1 - \lambda \|x - \pi_U(x)\| - \max \left(0, 1 - \lambda \|x - \pi_{N_{h_1}^{\mathbf{c}}}(x)}\|, \alpha - \lambda \|x - \pi_U(x)\| \right) d\mu_{-1}(x) \quad (53)$$

$$- \nu_1 \int_{(U \oplus \epsilon_2 \setminus U) \setminus P_{h_1}(\epsilon_2)} \max(1 - \alpha - \lambda \|x - \pi_U(x)\|, 0) d\mu_1(x) \quad (54)$$

Let us simplify Equation (49) using using additional hypothesis:

- A sufficient condition for the adversarial risk to decrease will be to choose $\max \left(1 - \alpha, \alpha - \lambda \|x - \pi_{P_{h_1}^{\mathbf{c}}}(x)}\| \right) =$

$\alpha - \lambda \|x - \pi_{P_{h_1}^{\epsilon}}(x)\|$, so that the attacker continues to attack on U even with a smaller probability of success, thus reducing the adversarial risk. This gives us $\alpha > \frac{1 + \lambda \max_{x \in U} \|x - \pi_{P_{h_1}^{\epsilon}}\|}{2}$. In the remaining we consider such an α .

- In particular, this gives $\alpha > 1/2$ and $\max(\alpha, 1 - \alpha - \lambda \|x - \pi_{U \oplus \epsilon_2 \setminus U}(x)\|) = \alpha$. Hence line (51) = $(1 - \alpha)\nu_{-1}\mu_{-1}(U) > 0$.
- Furthermore, we have that $1 - \lambda \|x - \pi_{P_{h_1}^{\epsilon}}(x)\| - \max\left(1 - \alpha - \lambda \|x - \pi_U(x)\|, 1 - \lambda \|x - \pi_{P_{h_1}^{\epsilon}}(x)\|\right)$ is equal to :

$$\begin{cases} 0 & \text{if } \max = 1 - \lambda \|x - \pi_{P_{h_1}^{\epsilon}}(x)\| \\ 1 - \lambda \|x - \pi_{P_{h_1}^{\epsilon}}(x)\| - (1 - \alpha) + \lambda \|x - \pi_U(x)\| > -(1 - \alpha) & \text{elsewhere} \end{cases}$$

Thus the expression on line (52) $> -\nu_1(1 - q)\mu_1((U \oplus \epsilon_2 \setminus U) \cap P_{h_1}(\epsilon_2))$.

- Also note that, $\max(1 - \alpha - \lambda \|x, \pi_U(x)\|, 0) < 1 - \alpha$. Hence line (54) $> -\nu_1(1 - \alpha)\mu_1((U \oplus \epsilon_2 \setminus U) \setminus P_{h_1}(\epsilon_2))$.

Finally, (52) + (54) $> -\nu_1(1 - \alpha)\mu_1((U \oplus \epsilon_2 \setminus U))$, hence the difference between the adversarial risks is as follows:

$$\Delta \mathcal{R}_{\text{adv}} > \nu_1(1 - \alpha)(\mu_1(U) - \mu_1((U \oplus \epsilon_2) \setminus U)) \quad (55)$$

Since μ_1 is vanishing on U , the expected result holds for $\alpha > \frac{1 + \lambda \max_{x \in U} \|x - \pi_{P_{h_1}^{\epsilon}}\|}{2}$. Not that for any $U \subset P_h(\epsilon_2)$, one have $\max_{x \in U} \|x - \pi_{P_{h_1}^{\epsilon}}\| \leq \epsilon_2$. Moreover, when $U = P_h(\epsilon_2)$, we get $\max_{x \in U} \|x - \pi_{P_{h_1}^{\epsilon}}\| = \epsilon_2$, which gives the expected result. \square

2. Experimental results

In the experimental section, we consider $\mathcal{X} = [0, 1]^{32 \times 32}$ to be the set of images, and $\mathcal{Y} = \{1, \dots, 10\}$ or $\mathcal{Y} = \{1, \dots, 100\}$ according to the dataset at hand.

2.1. Adversarial attacks

Let $(x, y) \sim D$ and $h \in \mathcal{H}$. We consider the following attacks:

- (i) **ℓ_{∞} -PGD attack.** In this scenario, the Adversary maximizes the loss objective function, under the constraint that the ℓ_{∞} norm of the perturbation remains bounded by some value ϵ_{∞} . To do so, it recursively computes:

$$x^{t+1} = \Pi_{B_{\|\cdot\|_{\infty}}(x, \epsilon_{\infty})} [x^t + \beta \text{sgn}(\nabla_x \mathcal{L}(h(x^t), y))] \quad (56)$$

where \mathcal{L} is some differentiable loss (such as the cross-entropy), β is a gradient step size, and Π_S is the projection operator on S . One can refer to (Madry et al., 2018) for implementation details.

- (ii) **ℓ_2 -C&W attack.** In this attack, the Adversary optimizes the following objective:

$$\underset{\tau \in \mathcal{X}}{\text{argmin}} \|\tau\|_2 + \lambda \times \text{cost}(x + \tau) \quad (57)$$

where $\text{cost}(x + \tau) < 0$ if and only if $h(x + \tau) \neq y$. The authors use a change of variable $\tau = \frac{1}{2}(\tanh(w) - x + 1)$ to ensure that $x + \tau \in \mathcal{X}$, a binary search to optimize the constant λ , and Adam or SGD to compute an approximated solution. One should refer to (Carlini & Wagner, 2017) for implementation details.

2.2. Experimental setup

Datasets. To illustrate our theoretical results we did experiments on the **CIFAR10** and **CIFAR100** datasets. See (Krizhevsky et al., 2009) for more details.

Classifiers. All the classifiers we use are WideResNets (see (Zagoruyko & Komodakis, 2016)) with 28 layers, a widen factor of 10, a dropout factor of 0.3 and LeakyRelu activations with a 0.1 slope.

Natural Training. To train an undefended classifier we use the following hyperparameters.

- **Number of Epochs:** 200
- **Batch size:** 128
- **Loss function:** Cross Entropy Loss
- **Optimizer :** SGD algorithm with momentum 0.9, weight decay of 2×10^{-4} and a learning rate that decreases during the training as follows:

$$lr = \begin{cases} 0.1 & \text{if } 0 \leq \text{epoch} < 60 \\ 0.02 & \text{if } 60 \leq \text{epoch} < 120 \\ 0.004 & \text{if } 120 \leq \text{epoch} < 160 \\ 0.0008 & \text{if } 160 \leq \text{epoch} < 200 \end{cases}$$

Adversarial Training. To adversarially train a classifier we use the same hyperparameters as above, and generate adversarial examples using the ℓ_∞ -**PGD** attack with 20 iterations. When considering that the input space is $[0, 255]^{32 \times 32}$, on **CIFAR10** and **CIFAR100**, a perturbation is considered to be imperceptible for $\epsilon_\infty = 8$. Here, we consider $\mathcal{X} = [0, 1]^{32 \times 32}$ which is the normalization of the pixel space $[0.255]^{32 \times 32}$. Hence, we choose $\epsilon_\infty = 0.031 (\approx 8/255)$ for each attack. Moreover, the step size we use for ℓ_∞ -**PGD** is 0.008 ($\approx 2/255$), and we use a random initialization for the gradient descent. For the ℓ_∞ -**PGD** attack against the mixture m_h^q , we use the same parameters as above, but compute the gradient over the expected loss (as explained in the main paper).

Evaluation Under Attack. At evaluation time, we use 100 iterations instead of 20 for ℓ_∞ -**PGD**, and the same remaining hyperparameters as before. For the ℓ_2 -**C&W** attack, we use 100 iterations, a learning rate equal to 0.1, 9 binary search steps, and an initial constant of 0.001. We give results for several different values of the rejection threshold: $\epsilon_2 \in \{0.4, 0.6, 0.8\}$.

Library used. We used the Pytorch and Advertorch libraries for all implementations.

Machine used. 6 Tesla V100-SXM2-32GB GPUs

Computing ℓ_2 -C&W on a mixture with Advertorch. The implementation of the ℓ_2 -**C&W** attack in Advertorch takes as input not only the loss, but also the probits of the classifier. Hence, when attacking a mixture we need to compute both the expected loss, *and* the expected probits of the mixture. See the submitted code for more details.

2.3. Experimental details on CIFAR10

Grid search on α . Let us provide some results on the grid search that helped us select the parameter $\alpha = 0.06$ we present in the main paper. Note that as described in the main paper, α here represents the weight of the new classifier added to the mixture at each step. As shown in Figure 3, when α is small (e.g. 0.01), the accuracy under attack increases slowly for every new classifier added to the mixture. Conversely, when α is too large (e.g. bigger than 0.07) the accuracy under attack rapidly increases for the first classifiers and then plummet when the mixture becomes too big (more than 4 classifiers). This phenomenon can be explained as follows: for bigger values of α , the probability of selecting the first classifiers decreases too quickly, so that after a few steps they are not taken into account anymore. Since the newest classifiers in the mixture are designed not to be robust by themselves but to compensate the weaknesses of the others, the accuracy under attack will then decrease. Hence $1-\alpha$ is a parameter that represents the 'memory' of the boosting procedure.

The parameter α and the size of the mixture have an influence not only on the accuracy under ℓ_∞ -**PGD** attack, but also on the natural accuracy of the mixture. This is why we selected a mixture with 10 classifiers and $\alpha = 0.06$. Indeed, this mixture offers the best trade-off between natural accuracy and accuracy under attack. For completeness, we present all evaluations in Table 2.

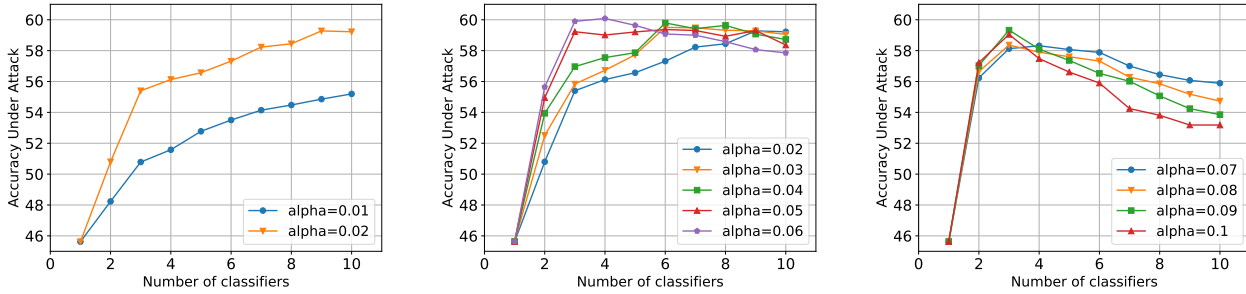


Figure 3. Evolution of the accuracy under attack of the mixture according to the number of classifiers with different value of $\alpha \in [0.01, 0.1]$. The evaluation is made with the ℓ_∞ -PGD attack with 100 iterations.

Even though we presented in the main paper a mixture with 10 classifiers, it is worth noting that similar accuracy under attack can be achieved with smaller mixtures, and different values for α . For instance with only 3 classifiers, one can achieve accuracy under ℓ_∞ -PGD attack of 0.59 by taking $\alpha = 0.1$. Thus, when having less computation power, one can still use our algorithm with good results under attack, at the cost of some natural accuracy.

$\alpha =$	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
$n = 1$	0.46									
$n = 2$	0.48	0.51	0.53	0.54	0.55	0.56	0.56	0.57	0.57	0.57
$n = 3$	0.51	0.55	0.56	0.57	0.59	0.60	0.58	0.58	0.59	0.59
$n = 4$	0.52	0.56	0.57	0.58	0.59	0.60	0.58	0.58	0.58	0.57
$n = 5$	0.53	0.57	0.58	0.58	0.59	0.60	0.58	0.58	0.57	0.57
$n = 6$	0.54	0.57	0.60	0.60	0.59	0.59	0.58	0.57	0.57	0.56
$n = 7$	0.54	0.58	0.59	0.59	0.59	0.59	0.57	0.56	0.56	0.54
$n = 8$	0.54	0.58	0.59	0.60	0.59	0.59	0.56	0.56	0.55	0.53
$n = 9$	0.55	0.59	0.59	0.59	0.59	0.58	0.56	0.55	0.54	0.53
$n = 10$	0.55	0.59	0.59	0.59	0.58	0.58	0.56	0.55	0.54	0.53

Table 2. Grid-search showing the accuracy under attack for $\alpha \in [0.01, 0.1]$ and $n \in [1, 10]$. The evaluation is made with the ℓ_∞ -PGD attack with 100 iterations.

Selecting the first element of the mixture. Our algorithm creates classifiers in a boosting fashion, starting with an adversarially trained classifier. There are several ways of selecting this first element of the mixture: use the classifier with the best accuracy under attack (option 1, called bestAUA), or rather the one with the best natural accuracy (option 2). Table 3 compares both options.

Beside the fact that any of the two mixtures outperforms the first classifier, we see that the first option always outperforms the second. In fact, when taking option 1 (bestAUA = True) the accuracy under ℓ_∞ -PGD attack of the mixture is 3% better than with option 2 (bestAUA = False). One can also note that both mixtures have the same natural accuracy (0.80), which makes the choice of option 1 natural.

Training method	NA of the 1 st clf	AUA of the 1 st clf	NA of the mixture	AUA of the mixture
BAT (bestAUA=True)	0.77	0.46	0.80	0.58
BAT (bestAUA=False)	0.83	0.42	0.80	0.55

Table 3. Comparison of the mixture that has as first classifier the best one in term of natural accuracy and the mixture that has as first classifier the best one in term of Accuracy under attack. The accuracy under attack is computed with the ℓ_∞ -PGD attack. NA means natural accuracy, and AUA means accuracy under attack.

2.4. Additional results on CIFAR100

As we did for **CIFAR10**, we compare the robustness of Adversarial Training with our boosting procedure on **CIFAR100**. Since the images have the same dimension, we use the same architectures and sets of hyperparameters as listed in Section 2.2, and only change the dimension of the output to be equal to 100. As shown in Table 4, a mixture of 5 classifiers constructed with our algorithm and $\alpha = 0.06$ has a better accuracy under attack, as well as a better natural accuracy, than Adversarial Training.

Training method	Natural	ℓ_∞ -PGD
(Madry et al., 2018)	0.585	0.271
BAT ($n = 5, \alpha = 0.06, \text{bestAUA}=\text{True}$)	0.592	0.402

Table 4. Evaluation on CIFAR100 without *data augmentation*. Accuracy under attack of a single classifier adversarially trained and the mixture formed with our algorithm. The evaluation is made with ℓ_∞ -PGD computed with 100 steps.