Randomization matters
How to defend against strong adversarial attacks

Rafael Pinot [*1,2] **Raphael Ettedgui** [*1] **Geovani Rizk** [1] **Yann Chevaleyre** [1] **Jamal Atif** [1]

Abstract

Is there a classifier that ensures optimal robustness against all adversarial attacks? This paper tackles this 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 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 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 one step boosting method. We evaluate this method, in Section 6, against strong adaptive attacks on CIFAR10 and CIFAR100 datasets. It outperforms Adversarial Training against both ℓ_∞**-PGD** (Madry et al., 2018), and ℓ_2**-C&W** (Carlini & Wagner, 2017) attacks. More precisely, on CIFAR10, our algorithm achieves 0.55 (resp. 0.53) accuracy

[*]Equal contribution [1]Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France [2]Institut LIST, CEA, Université Paris-Saclay, France. Correspondence to: Rafael Pinot <rafael.pinot@dauphine.fr>, Raphael Ettedgui <raphael.ettedgui@dauphine.eu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

under attack against these attacks, which is an improvement of 0.13 (resp. 0.18) over Adversarial Training.

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 on distributions for the Adversary, while we study a local, pointwise constraint, leading to different attack policies). 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 a close 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 players only have a finite set of possible strategies. We study a more general setting. Finally, (Dhillon et al., 2018) motivates 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 defense method we propose in this paper are not based on noise injection. However, a link could be made between these works and the mixture 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 distributionnal 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. The main differences are that we focus on studying equilibria and not deriving bounds. Moreover, these works do not study the influence of randomization. Finally they express the optimal risk of the Defender in terms of transportation costs between two distributions, whereas we explicitly study the Adversary's behaviour as a transport from one distribution to another. Even though they do not treat the problem from the same prism, we believe that these works are profoundly related and complementary to ours.

Ensemble of classifiers. Some works have been done to improve the robustness of a model by constructing ensemble of classifiers (Abbasi & Gagné, 2017; Xu et al., 2017; Verma & Swami, 2019; Pang et al., 2019; Sen et al., 2020). However all the defense methods proposed in those papers subsequently proved to be ineffective against adaptive attacks introduced in (He et al., 2017; Tramer et al., 2020). The main difference with our method is that it is not an ensemble method since it uses sampling instead of voting to aggregate the classifiers' output. Hence in terms of volatility, in voting methods, whenever a majority agrees on an opinion, all others votes will be ignored, whereas here each classifier always contributes according to its probability weights, which do not depend on the others.

3. A Game Theoretic point of view.

3.1. Initial 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. $\mathcal{D}$:

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

Where $\mathcal{H} := \{h : x \mapsto \operatorname{sgn} g(x) \mid g : \mathcal{X} \to \mathbb{R} \text{ continuous}\}$, $\nu \in \mathcal{P}(\mathcal{Y})$ 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 $X|(Y = y)$.

Adversarial example attack (point-wise). Given a classifier $h : \mathcal{X} \to \mathcal{Y}$ 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 CIFAR datasets, typical values for ϵ_∞ and ϵ_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 ℓ_2 norm, but all our results are valid for an ℓ_∞ 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 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 $\boldsymbol{\phi} = (\phi_{-1}, \phi_1) \in (\mathcal{F}_{\mathcal{X}|\epsilon_2})^2$ that maximizes the adversarial risk $\mathcal{R}_{\text{adv}}(h, \boldsymbol{\phi})$ defined as follows:

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

Where for any $\epsilon_2 \in (0, 1)$, $\mathcal{F}_{\mathcal{X}|\epsilon_2}$ is the set of functions that imperceptibly modifies a distribution:

$$\mathcal{F}_{\mathcal{X}|\epsilon_2} := \left\{ \psi \in \mathcal{F}_{\mathcal{X}} \mid \operatorname*{essup}_{x \in \mathcal{X}} \|\psi(x) - x\|_2 \leq \epsilon_2 \right\}.$$

Adversarial defense, a two-player zero-sum game. With the setting defined above, the adversarial examples problem can 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:

$$\inf_{h \in \mathcal{H}} \sup_{\boldsymbol{\phi} \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}_{\text{adv}}(h, \boldsymbol{\phi}). \quad (3)$$

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 $\boldsymbol{\phi}$) for the Defender (resp. the Adversary) is called a *strategy*. It is crucial to note that the $\sup$-$\inf$ and $\inf$-$\sup$ problems do not necessarily coincide. In this paper, we mainly focus on the Defender's point of view which corresponds to the $\inf$-$\sup$ problem. We will be interested in understanding the behaviour of players in this game, *i.e.* the best responses they have to a given strategy, and whether some equilibria may arise. This motivates the following definitions.

Definition 1 (Best Response)**.** *Let $h \in \mathcal{H}$, and $\boldsymbol{\phi} \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$. A* best response *from the Defender to $\boldsymbol{\phi}$ is a classifier $h^* \in \mathcal{H}$ such that $\mathcal{R}_{\text{adv}}(h^*, \boldsymbol{\phi}) = \min_{h \in \mathcal{H}} \mathcal{R}_{\text{adv}}(h, \boldsymbol{\phi})$. Similarly, a* best response *from the Adversary to h is an attack $\boldsymbol{\phi}^* \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$ such that $\mathcal{R}_{\text{adv}}(h, \boldsymbol{\phi}^*) = \max_{\boldsymbol{\phi} \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}_{\text{adv}}(h, \boldsymbol{\phi})$.*

In the remaining, we denote $\mathfrak{BR}(h)$ the set of all best responses of the Adversary to a classifier h. Similarly $\mathfrak{BR}(\boldsymbol{\phi})$ denotes the set of best responses to an attack $\boldsymbol{\phi}$.

Definition 2 (Pure Nash Equilibrium)**.** *In the zero-sum game (Eq. 3), a* Pure Nash Equilibrium *is a couple of strategies $(h, \boldsymbol{\phi}) \in \mathcal{H} \times \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$ such that*

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

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. The randomized regime will be studied in Section 5.

3.2. Trivial solution and Regularized Adversary

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:

$$\inf_{h\in\mathcal{H}} \sup_{\boldsymbol{\phi}\in\left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \underbrace{\left[\mathcal{R}_{\text{adv}}(h,\boldsymbol{\phi}) - \lambda\,\Omega\left(\boldsymbol{\phi}\right)\right]}_{\mathcal{R}^{\Omega}_{\text{adv}}(h,\boldsymbol{\phi})}. \tag{4}$$

The penalty function Ω 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* Ω_{mass}, and the *norm penalty* Ω_{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 $\boldsymbol{\phi}$ moves. Hence we define the mass penalty as:

$$\Omega_{\text{mass}}(\boldsymbol{\phi}) := \mathop{\mathbb{E}}_{Y\sim\nu}\left[\mathop{\mathbb{E}}_{X\sim\mu_Y}\left[\mathbb{1}\left\{X \neq \phi_Y(X)\right\}\right]\right]. \tag{5}$$

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 $\boldsymbol{\phi}$:

$$\Omega_{\text{norm}}(\boldsymbol{\phi}) := \mathop{\mathbb{E}}_{Y\sim\nu}\left[\mathop{\mathbb{E}}_{X\sim\mu_Y}\left[\|X - \phi_Y(X)\|_2\right]\right]. \tag{6}$$

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[1]. In the following, we denote $\mathfrak{BR}_{\Omega_{\text{mass}}}$ (resp. $\mathfrak{BR}_{\Omega_{\text{norm}}}$) 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 $N_h := \{x \in \mathcal{X} \mid h(x) = \text{-}1\}$ 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 $\boldsymbol{\phi} \in \mathfrak{BR}_{\Omega_{mass}}(h)$. Then the following assertion holds:*

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

Where $(P_h)^{\complement}$, the complement of P_h in $\mathcal{X}$. $\phi_{\text{-}1}$ is characterized symmetrically.

Lemma 2. *Let $h \in \mathcal{H}$ and $\boldsymbol{\phi} \in \mathfrak{BR}_{\Omega_{norm}}(h)$. Then the following assertion 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 $(P_h)^{\complement}$. $\phi_{\text{-}1}$ is characterized symmetrically.

These best responses are illustrated in Figure 1 with two unidimensional Gaussian distributions. For the mass penalty, μ_1 is set to 0 in $P_h(\epsilon_2)$, and this mass is transported into $N_h(\epsilon_2)$. The symmetric holds for $\mu_{\text{-}1}$. After attack, we now have $\mu_1\left(P_h(\epsilon_2)\right) = 0$, so a small value of $\mu_{\text{-}1}$ 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 μ_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

[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 $\mathcal{X}$.

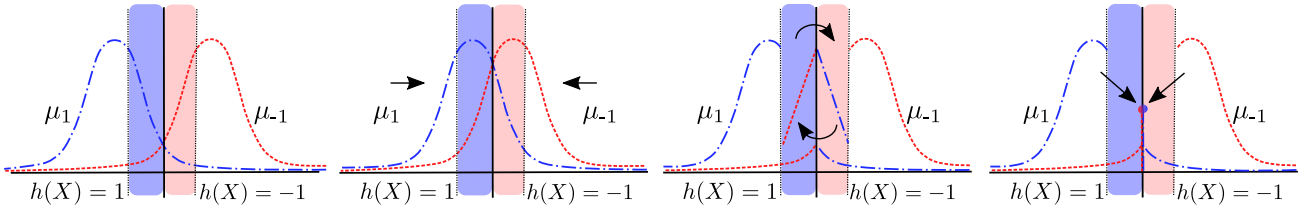

Figure 1. Representation of the $\mu_{\text{-}1}$ (blue dotted line) and μ_1 (red plain line) distributions, without attack (left) and with three different attacks: no penalty (second drawing), with mass penalty (third) and with norm penalty (fourth). On all figures blue area on the left of the axis is $P_h(\epsilon_2)$ and red area on the right is $N_h(\epsilon_2)$.

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 $\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$. If we take $h \in \mathfrak{BR}(\phi)$, 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 $\phi_{\text{-}1} \# \mu_{\text{-}1}$ admit probability density functions, Lemma 3 simply means that h is the Optimal Bayes Classifier for the distribution $(\nu, \phi_1 \# \mu_1, \phi_{\text{-}1} \# \mu_{\text{-}1})$[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 $\Omega \in \{\Omega_{mass}, \Omega_{norm}\}$, 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:

$$\sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \inf_{h \in \mathcal{H}} \mathcal{R}^{\Omega}_{\text{adv}}(h, \phi) \leq \inf_{h \in \mathcal{H}} \sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}^{\Omega}_{\text{adv}}(h, \phi).$$

On the left side problem (sup-inf), the Adversary looks for the best strategy ϕ 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:

$$\sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \inf_{h \in \mathcal{H}} \mathcal{R}^{\Omega}_{\text{adv}}(h, \phi) < \inf_{h \in \mathcal{H}} \sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}^{\Omega}_{\text{adv}}(h, \phi).$$

This means that both problems are not equivalent. In particular, an attack designed to succeed against *any* classifier (*i.e.* a transferable attack) 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 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

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:

$$\inf_{\eta \in \mathcal{P}(\mathcal{H})} \sup_{\varphi \in \mathcal{P}\left(\left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2\right)} \mathop{\mathbb{E}}_{\substack{h \sim \eta \\ \phi \sim \varphi}} \left[\mathcal{R}^{\Omega}_{\text{adv}}(h, \phi)\right]. \tag{7}$$

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 do not hold here.

[2] We prove this result in the supplementary material.

Randomization matters. Even without knowing if an equilibrium exists in the randomized setting, we can prove that *randomization matters*. More precisely we show that 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 $(\mathcal{F}_{\mathcal{X}})^2$ instead of $\mathcal{P}\left((\mathcal{F}_{\mathcal{X}})^2\right)$. 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 classifiers. 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* $n \in \mathbb{N}$, $\mathbf{h} = (h_1, ..., h_n) \in \mathcal{H}^n$ *, and* $\mathbf{q} \in \mathcal{P}(\{1, ..., n\})$*. 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* $y \in \mathcal{Y}$ *as follows:*

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

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 $\mathbf{h}$ 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:

$$\mathop{\mathbb{E}}_{Y \sim \nu}\left[\mathop{\mathbb{E}}_{X \sim \phi_Y \# \mu_Y}\left[\mathop{\mathbb{E}}_{\hat{Y} \sim m_{\mathbf{h}}^{\mathbf{q}}(X)}\left[\mathbb{1}\left\{\hat{Y} \neq Y\right\}\right]\right]\right] - \lambda\,\Omega(\boldsymbol{\phi}). \tag{8}$$

We also write $\mathcal{R}_{\text{adv}}^{\Omega}$ 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 $\mathbf{q}$. 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.

Toy example where a mixture outperforms AT. To better understand how randomization can work, let us look at a simple toy example. Figure 2 illustrates a binary classification setting between two set of points. Attacking the Optimal Bayes Classifier (bold straight line) consists in moving all the points that lie between the dotted lines to the opposite side of the decision boundary (Figure 2, left). The general tactic to defend against an attack is to change the classifier's output for points that are too 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 2 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 dilation and 2 stars that are close to the new boundary) to be smaller than the number of attacks we prevent (the squared points, 2 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 5 attacks at the cost of 4 new ones, so the Adversarial Training improves the total score from 8 to 7.

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 determinsitic outside the gray area, and random inside it[3]. If the first classifier has a weight $\alpha = 0.5$, 6 of the old attacks now succeed only with probability 0.5

[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 doted line. We focus on what happens on the star points for simplicity.

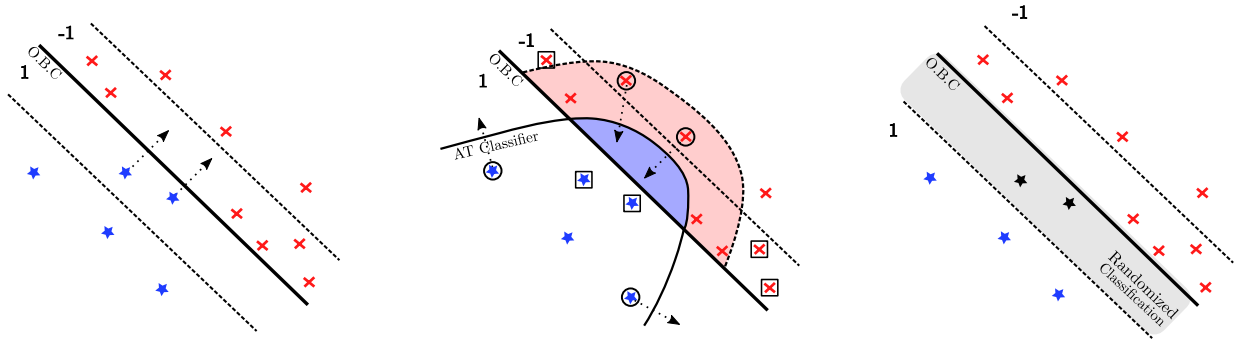

Figure 2. Illustration of adversarial examples (only on class 1 for more readability) crossing the decision boundary (left), adversarially trained classifier for the class 1 (middle), and a randomized classifier that defends class 1. Stars are natural examples for class 1, and crosses are natural examples 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 focus the drawing on the star points. Crosses can be treated symmetrically.

(crosses between the dotted lines), whereas 3 new attacks are created (stars outside of the gray area) that succeed with probability 0.5 also. At the end, the average rate of successful attacks is 6.5, where adversarial training previously achieved 7.

More formally, Theorem 2 shows that whatever penalty we consider, a deterministic classifier can always be outperformed by a randomized algorithm. We now can state our second main result: randomization matters.

Theorem 2. *(Randomization matters) Let us consider* $h_1 \in \mathcal{H}$, $\lambda \in (0,1)$, $\Omega = \Omega_{mass}$, $\phi \in \mathfrak{BR}_\Omega(h_1)$ *and* $h_2 \in \mathfrak{BR}(\phi)$. *Then for any* $\alpha \in (\max(\lambda, 1-\lambda), 1)$ *and for any* $\phi' \in \mathfrak{BR}_\Omega(m^{\mathbf{q}}_{\mathbf{h}})$ *one has*

$$\mathcal{R}^{\Omega}_{\text{adv}}(m^{\mathbf{q}}_{\mathbf{h}}, \phi') < \mathcal{R}^{\Omega}_{\text{adv}}(h_1, \phi).$$

Where $\mathbf{h} = (h_1, h_2)$, $\mathbf{q} = (\alpha, 1-\alpha)$, *and* $m^{\mathbf{q}}_{\mathbf{h}}$ *is the mixture of* $\mathbf{h}$ *by* $\mathbf{q}$. *A similar result holds when* $\Omega = \Omega_{norm}$ *(see supplementary materials).*

Remark Note that depending on the initial hypothesis h_1 and the conditional distributions μ_1 and $\mu_{\text{-}1}$, the gap between $\mathcal{R}^{\Omega}_{\text{adv}}(m^{\mathbf{q}}_{\mathbf{h}}, \phi')$and $\mathcal{R}^{\Omega}_{\text{adv}}(h_1, \phi)$ could vary. Hence, with additional conditions on h_1, μ_1 and $\mu_{\text{-}1}$, we could make the gap appear more explicitly. We keep the formulation general to emphasize that for *any* deterministic classifier, there exists a randomized one that outperforms it in terms of worst-case adversarial score.

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 α. The parameter α 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 α 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 $\tilde{D}$
$\mathbf{q} \leftarrow (1-\alpha, \alpha)$
$\mathbf{h} \leftarrow (h_1, h_2)$

return $m^{\mathbf{q}}_{\mathbf{h}}$

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 the high

[4] We use ℓ_∞**-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.

Dataset	Method	Natural Accuracy	Adaptive-l_∞-PGD $\epsilon_\infty = 0.031$	Adaptive-ℓ_2-C&W $\epsilon_2 = 0.4$	$\epsilon_2 = 0.6$	$\epsilon_2 = 0.8$
CIFAR10	Natural	0.88	0.00	0.00	0.00	0.00
	AT (Madry et al., 2018)	0.83	0.42	**0.60**	0.47	0.35
	Ours	0.80	**0.55**	**0.60**	**0.57**	**0.53**
CIFAR100	Natural	0.62	0.00	0.00	0.00	0.00
	AT (Madry et al., 2018)	0.58	0.26	0.38	0.29	0.22
	Ours	0.56	**0.40**	**0.45**	**0.41**	**0.38**

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-ℓ_∞-PGD** and **Adaptive-ℓ_2-C&W** attacks both computed with 100 iterations. For **Adaptive-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 **Adaptive-ℓ_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.

dimensional datasets we consider, due to its computational costs.

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[6]. 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-ℓ_∞-PGD** and **Adaptive-ℓ_2-C&W**.

Hard constraint parameter. The typical value of ϵ in the hard constraint depends on the norm we consider in the problem setting. In this paper, we use an ℓ_2 norm, however, the constraint parameter for **ℓ_∞-PGD** attack was initially set to be an ℓ_∞ constraint. In order to compare attacks of similar strength, we choose different threshold (ϵ_2 or ϵ_∞) 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 ϵ_2 equal to 0.6 and 0.4 as this values are sometimes used in the literature). Since **Adaptive-ℓ_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 ℓ_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 CIFAR10 and CIFAR100, of our method and classical Adversarial Training under attack with **Adaptive-ℓ_∞-PGD** and **Adaptive-ℓ_2-C&W**, both run for 100 iterations. We used 5 times more iterations 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 t a new classifier trained naturally on the dataset $\tilde{D}$ that contains adversarial examples against the mixture at step $t-1$. Since $\tilde{D}$ has to be constructed from a mixture, one would have to use an adaptive attack as **Adaptive-ℓ_∞-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.).

[6] In order for the attack to succeed, it it more efficient to compute the expected transformation of the logits instead of taking the expectation over the loss. More details on this in the supplementary materials.

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 $\mathcal{P}((\mathcal{F}_{\mathcal{X}})^2)$. 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 $(\mathcal{F}_{\mathcal{X}})^2$ so as to simplify the search in $\mathcal{P}((\mathcal{F}_{\mathcal{X}})^2)$ 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 does 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

Abbasi, M. and Gagné, C. Robustness to adversarial examples through an ensemble of specialists. *arXiv preprint arXiv:1702.06856*, 2017.

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., Azizzadenesheli, K., Bernstein, J. D., Kossaifi, 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.

He, W., Wei, J., Chen, X., Carlini, N., and Song, D. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.

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.

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.

Pang, T., Xu, K., Du, C., Chen, N., and Zhu, J. Improving adversarial robustness via promoting ensemble diversity. *arXiv preprint arXiv:1901.08846*, 2019.

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.

Sen, S., Ravindran, B., and Raghunathan, A. Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks. *arXiv preprint arXiv:2004.10162*, 2020.

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.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

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.

Verma, G. and Swami, A. Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 8643–8653, 2019.

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.

Xu, W., Evans, D., and Qi, Y. Feature squeezing mitigates and detects carlini/wagner adversarial examples. *arXiv preprint arXiv:1705.10686*, 2017.