A Theoretical Approach to Adversarial Robustness

Laurent Meunier

August 2021

Contents

Chapter 1

Introduction

Contents

1.1 Artificial Intelligence foundations

Machine Learning, the computer science subdomain dedicated to building and studying computer systems that automatically improve with experience, is at the very core of the recent advances in Artificial Intelligence. Finding its roots in statistical analysis, it has been widely studied over the past thirty years from algorithmic and mathematical perspectives, giving rise to a new discipline, computational learning theory. With the availability of massive amounts of data and computing power at low price, the last two decades have witnessed a growing interest in real-world applications of the domain. This interest is even stronger since 2012, with the remarkable success of of AlexNet [Krizhevsky et al., 2012] on the ImageNet challenge [Deng et al., 2009], using neural networks with several layers. The era of Deep Learning started then, with unexpected achievements in several domains: generative modeling [Goodfellow et al., 2014], natural language processing [Vaswani et al., 2017], etc. The success of Deep Learning (artificial neural networks with a large number of layers) can be explained by the conjunction of the following factors:

- **Availability of data:** the amount and the cost of data have largely decreased since the emergence of web platforms, and tools for large-scale data management.

- **Computational power:** new specialised hardware architectures such as GPUs and TPUs allow faster and larger training algorithms.
- **Algorithmic scalability:** algorithms are scalable to large models (Distributed Computing, etc.) and large number of data (Stochastic Gradient Descent [Bottou, 2010], etc.)
- **Open Source projects:** Large projects in Machine Learning are nowadays open-sourced (TensorFlow [Abadi et al., 2016], PyTorch [Paszke et al., 2017], Scikit Learn [Pedregosa et al., 2011], etc.) stimulating the emergence of large communities.

It is worth noting here that Artificial Intelligence, as a scientific domain, exists since early 20th century. Protean in nature, it encompasses several notions and fields, beyond Machine Learning, and Deep Learning. Its birth is inseparable from the development of computer science. The first efficient computer was built by Charles Babbage and ran Ada Lovelace's algorithm. Computer Science was formalized and theoretized in the Church-Turing thesis [Turing, 1950], which defines the notion of computability, i.e. functions are computable if they can be out as a list of predefined instructions to be followed. Such instructions are called algorithms. Artificial Intelligence, or at the least the term, was "officially founded" as a research field in 1956 at the Dartmouth Workshop [McCarthy et al., 1955], organized by Marvin Minsky, John McCarthy, Claude Shannon and Nathan Rochester. During this conference, the term "Artificial intelligence" was proposed and adopted by the community of researchers. Since then, the field has oscillated between hype and disappointment, with no less than two major period of disinterest as the AI winters. This thesis is clearly developed during the third hype's period, but we keep in mind the very enlightening history of the discipline.

1.2 Risks with Learning Systems

1.2.1 Common Threats

Cybersecurity is at the core of computer science. Cryptography has been one of the hottest topics during the last thirty years. Despite their performances, learning systems are subject to many types of vulnerabilities and, by their popularity, are then prone to malicious attacks. Probably, the most known vulnerability that got public attention is privacy. While the amount of available data is exponentially growing, recovering identities by crossing datasets is easier when data are not protected. As it was exhibited in the de-anonymization of the Netflix 1M$ prize dataset [Narayanan and Shmatikov, 2008], hiding identities in datasets is not sufficient to protect the privacy data. Computer scientists have then intensified their effort so as to propose ways to protect data, leading to the emergence to what is considered as a gold standard for data protection: Differential Privacy [Dwork, 2008]. It barely consists in adding noise to data to make them unrecoverable without too much deteriorating the their utility. It is appealing because it comes with strong theoretical guarantees, while being simple to manipulate, allowing to tradeoff between the degree of privacy through noise injection and the quality of the information one can infer from the data. Common privacy attacks are:

- **Model stealing [Tramèr et al., 2016]:** An attacker aims at stealing the parameters of a given model.
- **Membership inference [Shokri et al., 2017]:** Inferring whether a data sample was present or not in a training set.

Consequently to privacy threats, European authorities conceived the GDPR (General Data Protection Regulation)[1], adopted in 2016, which defines new rules on the use of data and on

[1] `https://eur-lex.europa.eu/eli/reg/2016/679/oj`

privacy. Today, GDPR is part of any data management plan of private companies. As an update of the GDPR, a second law proposition regarding data sharing from public and private companies has been introduced by the European Commission on The Governance of Data[2] in 2020.

Another type of vulnerability in Machine Learning is model failure. A malicious user, by modifying either the model or the data, can make it performs very poorly. The most known attacks aiming at model failures are:

- **Data poisoning attacks [Kearns and Li, 1993]:** changing some data in the training set so that the model performs very poorly on the hold-out set.
- **Evasion attacks [Biggio et al., 2013, Szegedy et al., 2014]**: small imperceptible perturbations at inference time. We will refer them to *"adversarial attacks"*.

Known and gaining interest in academia, these threats are not very known by most of the companies Kumar et al. [2020b]. More importantly, such vulnerabilities hinder the use of state of the art models in critical systems (autonomous vehicles, healthcare, etc.). In the manuscript we will focus on adversarial attacks. We introduce this threat more in details in the next paragraph.

References to adversarial examples in European Commission in law proposal on Artificial Intelligence systems

As part of the introduction: *"Cybersecurity plays a crucial role in ensuring that AI systems are resilient against attempts to alter their use, behaviour, performance or compromise their security properties by malicious third parties exploiting the system's vulnerabilities. Cyberattacks against AI systems can leverage AI specific assets, such as training data sets (e.g. data poisoning) or trained models (e.g. adversarial attacks), or exploit vulnerabilities in the AI system's digital assets or the underlying ICT infrastructure. To ensure a level of cybersecurity appropriate to the risks, suitable measures should therefore be taken by the providers of high-risk AI systems, also taking into account as appropriate the underlying ICT infrastructure."*

Title III (High risk AI systems), Chapter II (Requirements for high risk AI system), Article 14.52 (Human oversight): *"High-risk AI systems shall be resilient as regards attempts by unauthorised third parties to alter their use or performance by exploiting the system vulnerabilities. The technical solutions aimed at ensuring the cybersecurity of high-risk AI systems shall be appropriate to the relevant circumstances and the risks. The technical solutions to address AI specific vulnerabilities shall include, where appropriate, measures to prevent and control for attacks trying to manipulate the training dataset ('data poisoning'), inputs designed to cause the model to make a mistake ('adversarial examples'), or model flaws."*

A first regulation text on Artificial Intelligence[3] systems was proposed by the European commission in April 2021. This text includes a large section dedicated to "High Risk AI". High risk AI is referred to any autonomous system than can endanger human lives. This text aims at dealing with many threats in Learning Systems. Two direct references are made to adversarial attacks, underlying the need for companies to deal with them. The difficulty is to unify and create precise rules in a domain where results and certificates are mostly empirical. As mentioned earlier, it is known that robust models are often less performing and can make autonomous systems

[2] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020PC0767
[3] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206

unusable in real world scenarii. Thus, this text is a first step towards a unified regulation on autonomous systems but might miss precise requirements for models to be used in production.

1.2.2 Adversarial attacks against Machine Learning Systems

Despite the recent gain of interest in studying adversarial attacks in Machine Learning, the problematic exists however for a while and takes its source in SPAM classification where adversaries were spammers whose goal was to evade from the taken decision[4].

With the recent success of Deep Learning algorithms, in particular in computer vision, several authors [Biggio et al., 2013, Szegedy et al., 2014] have highlighted their vulnerability to adversarial attacks. Adversarial attacks in this case are widely understood as "imperceptible" perturbations of an image, i.e. slight changes in the pixels, so that this image remains unchanged from human sights. This characteristic might be surprising but is actually a severe curb in applying state-of-the-art deep learning methods in critical systems. There are number of issues that makes difficult building and evaluating robust models for real life applications:

1. The notion of imperceptibility is not well understood: numerically measuring human perception is still an open problem. Hence, detecting the change of perception due to adversarial attacks is an ill-posed problem. Most of the research in the domain focused on pixel-wise perturbations (e.g. ℓ_p norms), while real world threats would be crafted by inserting some misleading objects in the environment (e.g. patches [Brown et al., 2017], T-shirts [Xu et al., 2020], textures [Wiyatno and Xu, 2019],etc.).
2. Robustness is often empirically measured: there exist only a few methods with formal guarantees on the robustness and these guarantees are often loose. Robustness is usually measured on a set of possible attacks and not all possible perturbations are spanned by these attacks, leaving rooms for potential blind spots.
3. There exists a trade-off between robustness and accuracy. Most models that are robust suffer from a performance drop on natural data. For instance, a robustly trained robot will perform much lower on natural tasks than an accurate non-robust robot. That makes robust models unusable in real world applications [Lechner et al., 2021].

1.3 Adversarial Classification in Machine Learning

In this manuscript, we will focus on the task of classification in Machine Learning. The purpose of this task is to "learn" how to classify some input x into some label(s). The input can be an image, a text, an audio, etc. For instance, in computer vision, a known dataset is ImageNet where the goal is to learn how to classify high quality images into 1000 labels [Deng et al., 2009]. In natural language processing, the IMDB Movie Review Sentiment Classification dataset [Maas et al., 2011] aims at classifying positive or negative sentiments from movie reviews. To learn a classifier, the task is often supervised, i.e, we have access to labeled inputs, which constitutes the so-called training set. To assess the quality of the learnt model, we evaluate it on other images that constitute the test set.

1.3.1 A Learning Approach for Classification

From now, we will assume that the inputs are in some space $\mathcal{X}$ and the labels form a set $\mathcal{Y} := \{1, \dots, K\}$. To learn an adequate classification model, we denote $\{(x_1, y_1), \dots, (x_N, y_N)\}$

[4]Dalvi et al. [2004] showed that linear classifiers used in spam classification could be fooled by simple "evasion attacks" as spammers inserted "good words" into their spam emails.

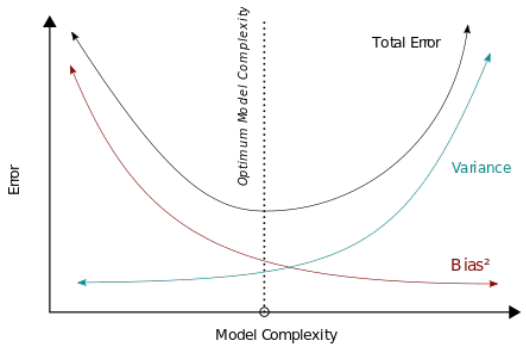


Figure 1.1: Bias-Variance tradeoff. A model with low complexity will have a low variance but an high bias. A model with high complexity will have a low bias but an high variance.

the N elements of $\mathcal{X} \times \mathcal{Y}$ forming the training set. We furthermore assume that these inputs are independent and identically distributed (i.i.d.) from some distribution $\mathbb{P}$ on $\mathcal{X} \times \mathcal{Y}$. The aim is now to learn a function/hypothesis from these samples $h : \mathcal{X} \to \mathcal{Y}$ to classify an input x with a label y. To assess the quality of a classifier, the metric of interest is often the misclassification rate of the model, or the 0/1 loss risk, and it is defined as:

$$\mathcal{R}_{0/1}(h) := \mathbb{P}(h(x) \neq y) = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\mathbf{1}_{h(x)\neq y}\right]$$

The optimal classifier, minimizing the standard risk is called the Bayes optimal classifier and is defined as $h(x) = \operatorname{argmax}_k \mathbb{P}(y = k \mid x)$. As the sampling distribution $\mathbb{P}$ is usually unknown, the optimal Bayes classifier is also unknown. The accuracy is often empirically evaluated on a test set $\{(x'_1, y'_1), \ldots, (x'_M, y'_M)\}$ independent from the training set and i.i.d. sampled from $\mathbb{P}$. To find this classifier h, we learn a function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^K$ returning scores, or logits, $(f_1(x), \ldots, f_K(x))$ corresponding to each label. Then h is set to $h(x) = \operatorname{argmax}_k f_k(x)$. The function $\mathbf{f}$ is usually learned by minimizing the empirical risk for a certain convenient loss function L over some class of functions $\mathcal{H}$.

$$\inf_{\mathbf{f}\in\mathcal{H}} \widehat{\mathcal{R}}_N(\mathbf{f}) := \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{f}(x_i), y_i).$$

This problem is called Empirical Risk Minimization (ERM). The theory of this problem has been widely studied and is well understood. It is often argued that there is a tradeoff on the "size" of $\mathcal{H}$: having a too small $\mathcal{H}$ may lead to underfitting, i.e. not enough parameters to describe the optimal possible function while a too large $\mathcal{H}$ may lead to overfitting, i.e. fitting too much training data. We often talk about bias-variance tradeoff (see Figure 1.1). A penalty term $\Omega_{\mathcal{H}}(\mathbf{f})$ can also be added to the ERM objective to prevent from overfitting. This tradeoff was recently questioned by the double descent [Belkin et al., 2019] phenomenon where overparametrized (i.e. number of parameters largely over the number of training samples) regimes lower the risk.

The presence of adversaries in classification questions the knowledge we have in standard statistical learning. Indeed most standard results do not hold in presence of adversaries, hence, opening a new research area dedicated to studying and understanding the classification problem in presence of adversarial attacks, and more importantly, deepens our understanding of machine learning/deep learning in high dimensional regimes.

1.3.2 Classification in Presence of Adversarial Attacks

Though a model can be very well performing on natural samples, small perturbations of these natural samples can lead to unexpected and critical behaviours of classification models [Biggio et al., 2013, Szegedy et al., 2014]. To formalize that, we will assume the existence of a "perception" distance $d : \mathcal{X}^2 \to \mathbb{R}$ such that a perturbation x' of an input x remains imperceptible if $d(x, x') \leq \varepsilon$ for some constant $\varepsilon \geq 0$. This "perception" distance is difficult to define in practice. For images, the $\|\cdot\|_\infty$ distance over pixels is often used, but is not able to capture all imperceptible perturbations. This choice is purely arbitrary: for instance, we will highlight in the manuscript that $\|\cdot\|_2$ perturbations can also be imperceptible while having a large $\|\cdot\|_\infty$. Image classification algorithms are also vulnerable to geometric perturbations, i.e. rotations and translations [Engstrom et al., 2019, Kanbak et al., 2018].

Therefore, the goal of an attacker is to craft an adversarial input x' from an input x that is imperceptible , i.e. $d(x, x') \leq \varepsilon$ and misclassifies the input, i.e. $h(x') \neq y$. Such a sample x' is called an adversarial attack. The used criterion cannot be the misclassification rate anymore, we need to take into account the possible presence of an adversary that maliciously perturbs the input. We then define the robust/adversarial misclassification rate or robust/adversarial 0/1 loss risk:

$$\begin{aligned}\mathcal{R}^{\varepsilon}_{0/1}(h) &:= \mathbb{P}_{(x,y)}(\exists x' \in \mathcal{X} \text{ s.t. } d(x, x') \leq \varepsilon \text{ and } h(x') \neq y) \\ &= \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{x'\in\mathcal{X} \text{ s.t. } d(x,x')\leq\varepsilon} \mathbf{1}_{h(x')\neq y}\right]\end{aligned}$$

Akin standard risk minimization, we aim to learn a function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^K$ such that $h(x) = \operatorname{argmax}_k f_k(x)$. Usually in adversarial classification we aim at solving the following optimization problem, that we will call adversarial empirical risk minimization:

$$\inf_{\mathbf{f}\in\mathcal{H}} \widehat{\mathcal{R}}^{\varepsilon}_N(\mathbf{f}) := \frac{1}{N}\sum_{i=1}^{N} \sup_{x'\in\mathcal{X} \text{ s.t. } d(x,x')\leq\varepsilon} L(\mathbf{f}(x_i), y_i).$$

This problem is more challenging to tackle than the standard risk minimization since it involves a hard inner supremum problem [Madry et al., 2018]. Guarantees in the adversarial setting are therefore difficult to obtain both in terms of convergence and statistical guarantees. The usual technique to solve this problem is called Adversarial Training [Goodfellow et al., 2015, Madry et al., 2018]. It consists in alternating inner and outer optimization problems. Such a technique improves in practice adversarial robustness but lacks theoretical guarantees. So far, most results and advances in understanding and harnessing adversarial attacks are empirical [Ilyas et al., 2019, Rice et al., 2020], leaving many theoretical and practical questions open. Moreover, robust models suffer from a performance drop and vulnerablity of models is currently still very high (see Table 1.1), which leaves room for substantial improvements.

Attacker	**Paper reference**	**Standard Acc.**	**Robust Acc.**
None	[Zagoruyko and Komodakis, 2016]	94.78%	0%
$\ell_\infty(\varepsilon = 8/255)$	[Rebuffi et al., 2021]	89.48%	62.76%
$\ell_2(\varepsilon = 0.5)$	[Rebuffi et al., 2021]	91.79%	78.80%

Table 1.1: State of the art accuracies on adversarial tasks on a WideResNet 28x10 [Zagoruyko and Komodakis, 2016]. Results are reported from [Croce et al., 2020a]

1.4 Outline and Contributions

We will first introduce in Chapter 2 the necessary background regarding Machine Learning and Adversarial Examples. We will then analyze adversarial attacks from three complementary points of view outlined as follows.

1.4.1 A Game Theoretic Approach to Adversarial Attacks

A line of research, following Pinot et al. [2020], to understand adversarial classification is to rely on game theory. In Chapter 4, we will build on this approach and define precisely the motivations for both the attacker and the classifier. We will cast it naturally as a zero sum game. We will in particular, study the problem of the existence of equilibria. More precisely, we will answer the following open question.

Question 1

What is the nature of equilibria in the adversarial examples game?

In game theory, there are many types of equilibria. In this manuscript, we will focus on Stackelberg and Nash equilibria. We will show the existence of both when both the classifier and the attacker play randomized strategies. To reach such equilibria, the classifier will be random, and the attacker will move randomly the samples at a maximum distance of ε. Then, we will propose two different algorithms to compute the optimal randomized classifier in the case of a finite number of possible classifiers. We will finally propose a heuristic algorithm to train a mixture of neural networks and show experimentally the improvements we achieve over standard methods.

This work **Mixed Nash Equilibria in the Adversarial Examples Game** was published at ICML2021.

1.4.2 Loss Consistency in Classification in Presence of an Adversary

In standard classification, consistency with regards to $0/1$ loss is a desired property for the surrogate loss L used to train the model. In short, a loss L is said to be consistent if for every probability distribution, a sequence of classifiers (f_n) that minimizes the risk associated with the loss L, it also minimizes the $0/1$ loss risk. Usually, in standard classification, the problem is simplified thanks to the notion of calibration. We will see that the question of consistency in the adversarial problem is much harder.

Question 2

Which losses are consistent with regards to the $0/1$ loss in the adversarial classification setting?

We tackle this question by showing that usual convex losses are not calibrated for the adversarial classification loss. Hence this negative result emphasizes the difficulty of understanding the adversarial attack problem, and building provable defense mechanisms.

1.4.3 Building Certifiable Models

The last problem we deal with in this manuscript is the implementation of robust certifiable models. In short, a classifier is said to be certifiable at an input x at level ε if one can ensure there exist no adversarial examples in the ball of radius ε. This problem is challenging since it is far from trivial to come up with non vacuous bounds that are exploitable in practice.

Question 3

How to efficiently implement certifiable models with non-vacuous guarantees?

To this end, we propose two methods that enforce Lipschitzness on the predictions of neural networks:

1. The first one consists in noise injection. We show that by adding a noise on an input of a classifier, we are able to get guarantees on the decision up to some level ε. This work **Theoretical evidence for adversarial robustness through randomization** was published at NeurIPS2019.
2. A second one consists in building contractive blocks in a ResNet architecture. This method draws its inspiration from the continuous flow interpretation of residual networks. More precisely, we show that using a gradient flow of a convex function, our network is 1-Lipschitz. We then design such a function, showing empirically and theoretically the robustness benefits of this approach.

1.4.4 Additional Works

Additionally to the works we present in the main document, we also present some other contributions we made during the thesis. These are deferred to the appendices.

Regarding adversarial examples, we will present:

- **Adversarial Attacks on Linear Contextual Bandits (see Appendix C):** we build provable attacks against online recommendation systems, namely Linear Contextual Bandits. This work was published at NeurIPS2020.
- **ROPUST: Improving Robustness through Fine-tuning with Photonic Processors and Synthetic Gradients (see Appendix D):** we use an Optical Processor Unit over existing defenses to improve adversarial robustness. This work was published at a workshop on Adversarial Attacks at ICML2021.

We published a paper in optimal transport named **Equitable and Optimal Transport with Multiple Agents (see Appendix ??)** where we introduce a way to deal with multiple costs in optimal transport by equitably partitioning transport among costs. We also published many works in the field of evolutionary algorithms:

- **Variance Reduction for Better Sampling in Continuous Domains (see Appendix ??)**: we show that, in one shot optimization, the optimal search distribution, used for the sampling, might be more peaked around the center of the distribution than the prior distribution modelling our uncertainty about the location of the optimum. This work was published at PPSN2020.
- **On averaging the best samples in evolutionary computation (see Appendix ??):** we prove mathematically that a single parent leads to a sub-optimal simple regret in the case of the sphere function. We provide a theoretically-based selection rate that leads to

better progress rates. This work was published at PPSN2020.

- **Asymptotic convergence rates for averaging strategies (see Appendix ??):** we extend the results from the previous papers to a wide class of functions including C^3 functions with unique optima. This work was published at FOGA2021.
- **Black-Box Optimization Revisited: Improving Algorithm Selection Wizards through Massive Benchmarking (see Appendix ??):** We propose a wide range of benchmarks integrated in Nevergrad [Rapin and Teytaud, 2018] platform. This work was published in the journal TEVC.

Chapter 2

Background

This chapter introduces the necessary background on classification on adversarial examples.

Contents

2.1 Supervised Classification

A classification task aims at learning a function that assigns a label to a given input. Along with regression, classification is one of the supervised learning tasks. One can find classification tasks in Computer Vision [Deng et al., 2009, Krizhevsky et al., LeCun and Cortes, 2010], Natural Language Processing [Vaswani et al., 2017], Speech Recognition [Dong et al., 2018], etc. In this thesis, most examples will be from Computer Vision and Image Recognition.

2.1.1 Notations

In this section, we formalize the task of classification. First, we define the notions of inputs and labels:

- Consider an input space $\mathcal{X}$, typically images. We assume this space is endowed with an arbitrary metric d, possibly a perception distance or any ℓ_p norm. In the following of the manuscript, unless it is specified, $(\mathcal{X}, d)$ will be a *proper* (i.e. closed balls are compact) *Polish* (i.e. completely separable) metric space. Note that for any norm $\|\cdot\|$, $(\mathbb{R}^d, \|\cdot\|)$ is a proper Polish metric space.
- Each input $x \in \mathcal{X}$ has to be associated with a label y. A label is a descriptor of the input. The set of labels is discrete and we designate it by $\mathcal{Y} := \{1, \dots, K\}$. $\mathcal{Y}$ is endowed with the trivial metric $d'(y, y') = \mathbf{1}_{y \neq y'}$. Note that $(\mathcal{X} \times \mathcal{Y}, d \oplus d')$ is also a proper Polish space.

The purpose of classification is to learn a classifier $h : \mathcal{X} \to \mathcal{Y}$. It is usual to learn a function $f : \mathcal{X} \to \mathbb{R}^K$ such that: $h(x) = \operatorname{argmax}_{k \in \mathcal{Y}} f_k(x)$. In a classification problem in machine learning, the data is assumed to be sampled from an unknown probability distribution $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$. We will assume from now that all the probability distributions we consider are Borel distributions. For any Polish Space $\mathcal{Z}$, we will denote $\mathcal{B}(\mathcal{Z})$ the Borel σ-algebra and the set of Borel distributions over $\mathcal{Z}$ will be denoted $\mathcal{M}_+^1(\mathcal{Z})$. We recall that on Polish space, all Borel probability distributions are Radon measures. We also recall the notion of *universal measurability*: a set $A \subset \mathcal{Z}$ is said to be universally measurable if it is measurable for every *complete* Borel probability measure.

When $\mathcal{Z}$ and $\mathcal{Z}'$ are two measurable spaces endowed with their Borel σ-algebra (unless specified), we will denote $\mathcal{F}(\mathcal{Z}, \mathcal{Z}')$ the space of measurable functions from $\mathcal{Z}$ to $\mathcal{Z}'$. Without loss of generality, when $\mathcal{Z}' = \mathbb{R}$, we will simply denote: $\mathcal{F}(\mathcal{Z}) := \mathcal{F}(\mathcal{Z}, \mathcal{Z}')$.

2.1.2 Classification Task in Supervised Learning

In standard classification, we usually aim at maximizing the accuracy of the classifier, or equivalently, at minimizing the risk associated with the $0/1$ loss defined as follows.

Definition 1. *Let $\mathbb{P}$ be a Borel probability distribution over $\mathcal{X} \times \mathcal{Y}$. Let $h : \mathcal{X} \to \mathcal{Y}$ be a Borel measurable classifier. Then, the risk of h associated with $0/1$ loss (or error of h) is defined as:*

$$\mathcal{R}_{\mathbb{P}}(h) := \mathbb{P}(h(x) \neq y) = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\mathbf{1}_{h(x)\neq y}\right] \tag{2.1}$$

The Optimal Bayes risk is defined as the optimal risk over measurable classifiers $\mathcal{F}(\mathcal{X}, \mathcal{Y})$:

$$\mathcal{R}^\star_{\mathbb{P}} := \inf_{h \in \mathcal{F}(\mathcal{X},\mathcal{Y})} \mathcal{R}_{\mathbb{P}}(h) \tag{2.2}$$

If $f : \mathcal{X} \to \mathbb{R}^K$, then the risk of f is defined as $\mathcal{R}_{\mathbb{P}}(f) := \mathbb{P}(\operatorname{argmax}_{k\in\mathcal{Y}} f_k(x) \neq y)$

TODO: note that the loss tis not indexed when talking about bayes risk Note that this quantity is well defined when h or f is Borel or universally measurable. The optimal classifier is called the /emphOptimal Bayes classifier and is defined as $h^\star(x) = \operatorname{argmax}_k \mathbb{P}(y = k \mid x)$. We remark that the disintegration theorem ensures that $x \mapsto \mathbb{P}(y = k \mid x)$ is indeed Borel measurable.

In practice, the access to the Optimal Bayes classifier is not possible because it requires full knowledge of the probability distribution $\mathbb{P}$ which is not the case in general. Instead, in the supervised learning setting, the learner has access to data points $\{(x_1, y_1), \dots, (x_N, y_N)\}$, that constitutes the *training set*. Knowing the Optimal Bayes classifier on training points is not sufficient to generalize on points out of the training set. Hence one needs to reduce the search space of measurable functions to a much smaller one, denoted $\mathcal{H}$ in the sequel. The $0/1$ loss is not convex neither continuous, and minimizing directly the $0/1$ loss risk on $\mathcal{H}$ might be NP-hard even for simple set of hypotheses as linear classifiers. We usually minimize a well-chosen surrogate loss function L. A *loss function* $L : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ is a non negative Borel measurable function.

An example of such a loss is the cross entropy loss defined as $L(f(x), y) = -\sum_{i=1}^{K} \mathbf{1}_{y=i} \log f_i(x)$ where $f_i(x)$ is the probability learnt by the model with input x belonging to the class i. Hence the objective is to minimize the empirical risk associated with $\mathcal{H}$ using the loss L defined as:

$$\widehat{\mathcal{R}}_L(f) := \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), y_i).$$

Neural Networks A popular set of classifiers are Neural Networks. They gained in popularity due to their exceptional performances in Image Recognition He et al. [2016], Krizhevsky et al. [2012] or Natural Language Processing for instance Vaswani et al. [2017]. In its simpler form, a neural network is a concatenation of linear operators and non-linear functions (called *activations*). This concatenation are called *layers*. Formally a neural network with L layers writes:

$$f(x) = (W_L \sigma (W_{L-1} \dots \sigma (A_1 x + b_1) \dots) + b_L)$$

where W_i are called the weight matrices and b_i the biases. In the case of image recognition, the weights may have a special structure of convolution: such networks are called *Convolutional Networks*. We illustrate a convolutional layer in Figure **??**.

To train neural networks, the backpropagation is a standard algorithm based on the chain rule. This algorithm is subject to gradient vanishing, or gradient explosion issues. To circumvent these problems, many tricks were proposed as using ReLU-like activation functions [Ramachandran et al., 2017, Xu et al., 2015], Dropout [Srivastava et al., 2014], Batch Normalization [Ioffe and Szegedy, 2015] or the use of Residual Layers [He et al., 2016]. More, despite their popularity, it is difficult to understand the outstanding performances of neural networks.

2.1.3 Surrogate losses, consistency and calibration

Binary Classification. In this section, we recall the main results about surrogate losses in binary classification. We assume that $\mathcal{Y} = \{-1, +1\}$. In this case, a classifier is a measurable function $f : \mathcal{X} \to \mathbb{R}$ such that an input x is classified as 1 if $f(x) > 0$ and as -1 if $f(x) \leq 0$. Then the 0/1 loss is defined as $\mathbf{1}_{y \times sign(f(x)) \leq 0}$. As mentioned earlier, optimizing the risk associated with the 0/1 loss is a difficult task. We need to properly introduce notions of surrogate losses.

A margin loss is a loss L such that there exist a measurable function $\phi : \mathbb{R} \to \mathbb{R}_+$, satisfying, $L(x, y, f) = \phi(yf(x))$. The risk associated with a margin loss ϕ is then $\mathcal{R}_{\phi,\mathbb{P}}(f) := \mathbb{E}_{\mathbb{P}}[\phi(yf(x))]$. A loss ϕ is said to be *classification-consistent* if every minimizing sequence for the risk associated with the ϕ loss is also a minimizing sequence for the risk associated with the 0/1-loss. In other words, for a given $\mathbb{P} \in \mathcal{M}_1^+(\mathcal{X} \times \mathcal{Y})$, ϕ is classification-consistent for $\mathbb{P}$ if for all sequences $(f_n)_{n \in \mathbb{N}}$ of measurable functions:

$$\mathcal{R}_{\phi,\mathbb{P}}(f_n) \to \mathcal{R}^\star_{\phi,\mathbb{P}} := \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{\phi,\mathbb{P}}(f) \implies \mathcal{R}_{\mathbb{P}}(f_n) \to \mathcal{R}^\star_{\mathbb{P}} \tag{2.3}$$

While this notion seems complicated to study, Bartlett et al. [2006], Steinwart [2007], Zhang [2004b] have focused on a relaxation named *calibration*. A loss is said to be *classification-calibrated* if for every $\varepsilon > 0$, there exists $\delta > 0$ such that for every $\alpha \in \mathbb{R}$ and $\eta \in [0, 1]$:

$$\eta\phi(\alpha) + (1-\eta)\phi(-\alpha) - \min_{\beta \in \mathbb{R}} [\eta\phi(\beta) + (1-\eta)\phi(-\beta)] \leq \delta \implies sign\left((\eta - \frac{1}{2})\alpha\right) = 1$$

We remark the notion of calibration is basically a pointwise notion of consistency with η corresponding to $\mathbb{P}(y = 1|x)$. Bartlett et al. [2006], Steinwart [2007], Zhang [2004b] proved the

equivalence of the two notions in the case of standard-binary classification. In particular they show that a wide range of convex margin losses are actually classification-consistent: if ϕ is convex and differentiable at 0, then ϕ is calibrated if and only if $\phi'(0) < 0$.

The problem of consistency have been investagated in the case of multi-label classification by Zhang [2004a]. The results can be similarily derived and it was show that large range of convex functions are actually consistent for classification problems.

2.1.4 Empirical Risk Minimization and Generalization

As mentioned earlier, the learner has access to training points $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ and not to the whole distribution. We aim at learning the classifier on a set of functions $\mathcal{H}$. The classifier $\hat{f}_N$ is then chosen to minimize the empirical risk given a loss L:

$$\hat{f}_N = \operatorname*{argmin}_{f \in \mathcal{H}} \widehat{\mathcal{R}}_L(f) = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), y_i).$$

Since the learning procedure takes into account a finite number of samples and a set $\mathcal{H}$ of hypotheses, one need to control the risk of the classifier $\hat{f}_N$.

Risk Decomposition and bias-complexity tradeoff. The excess risk of a classifier is defined as the difference between the risk and the optimal risk: $\mathcal{R}_L(f_n) - \mathcal{R}_L^\star$. The excess risk can be decomposed as follows:

$$\mathcal{R}_L(\hat{f}_N) - \mathcal{R}_L^\star = \left(\mathcal{R}_L(\hat{f}_N) - \mathcal{R}_{L,\mathcal{H}}^\star\right) + \left(\mathcal{R}_{L,\mathcal{H}}^\star - \mathcal{R}_L^\star\right)$$

with $\mathcal{R}_{L,\mathcal{H}}^\star = \inf_{f \in \mathcal{H}} \mathcal{R}_L(f)$. The two terms in the previous decomposition corresponds respectively to:

- **The estimation risk**: the empirical risk $\mathcal{R}(\hat{f}_N)$ (i.e., training error) is only an estimate of the optimal risk, and so $\hat{f}_N$ is only an estimate of the predictor minimizing the true risk. The estimation risk depends on the training set size N and on the size, or complexity, of $\mathcal{H}$. The more samples we have the smaller will be the estimation risk and more complex $\mathcal{H}$ is the larger the estimation error will be.
- **The approximation risk**: the approximation risk is the error made by optimizing over $\mathcal{H}$ instead of minimization over the whole space of measurable functions. As the function space $\mathcal{H}$ grows, the approximation naturally decreases.

This decomposition induces a tradeoff on the complexity of $\mathcal{H}$ named *bias-complexity tradeoff* or *bias-variance tradeoff*. On one hand, if $\mathcal{H}$ is not enough rich, then the estimation risk would be small but the approximation error can be large, it is called *underfitting*. On the other hand, if $\mathcal{H}$ is is too rich, then the approximation risk would be small but the estimation error large, it is called *overfitting*. To overcome these issues in practice, it is usual to add a regularization parameter to the empirical risk depending on the set $\mathcal{H}$:

$$\hat{f}_N = \operatorname*{argmin}_{f \in \mathcal{H}} \widehat{\mathcal{R}}_L(f) + \lambda \times \Omega_{\mathcal{H}}(f) = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), y_i) + \lambda \times \Omega_{\mathcal{H}}(f).$$

The convergence of regularized least squares regression has been largely studied on Reproducing Kernel Hilbert Space (RKHS). A RKHS $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ is characterized by a symmetric, positive definite function called a kernel over $\mathcal{X}$ such that for all $f \in \mathcal{H}$ and $x \in \mathcal{X}$, $f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$. In this case, the regularization parameter $\Omega_{\mathcal{H}}(f)$ is the square norm of f: $\|f\|_{\mathcal{H}}^2$.

Uniform Convergence. Since $\hat{f}_N$ is dependent on the training samples, it is usually difficult to estimate $\mathcal{R}(\hat{f}_N)$ from training samples. A natural thing to do is to upperbound this quantity using:

$$|\widehat{\mathcal{R}}(\hat{f}_N) - \mathcal{R}(\hat{f}_N)| \leq \sup_{f \in \mathcal{H}} |\widehat{\mathcal{R}}(f) - \mathcal{R}(f)|$$

The convergence of the right-end term is referred as uniform convergence or PAC-learning [Valiant, 1984]. It can be bounded either with high probability or in expectation (i.e. L^1 convergence). We remark the speed of convergence depends on the complexity of $\mathcal{H}$: more complex $\mathcal{H}$ is, the slower the convergence will be, hence exhibiting again a tradeoff on the expressivity of $\mathcal{H}$. There have been a lot of research that proposed tools to study this convergence. Now, we recall a fundamental tool, namely the Rademacher complexity.

The Rademacher complexity was introduced by Bartlett and Mendelson [2002] to study the problem of uniform convergence. Given a set of functions $\mathcal{H}$, and observations $S = \{z_1, \ldots, z_N\}$ from a distribution $\mathbb{P}$, the empirical Rademacher complexity is defined as:

$$\widehat{Rad}_S(\mathcal{H}) := \frac{2}{N} \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} |\sum_{i=1}^{N} \sigma_i h(z_i)| \right]$$

where σ_i are independent samples from Rademacher law: $P[\sigma_i = +1] = P[\sigma_i = -1] = \frac{1}{2}$. When $\mathcal{H}$ is not too complex (for instance, finite set or linear classifiers), one can bound the Rademacher complexity by $O(n^{-1/2})$. The Rademacher complexity upperbounds the uniform risk error as follows:

$$\mathbb{E}_{S \sim \mathbb{P}^N} \left[\sup_{h \in \mathcal{H}} |e_S(h) - e_{\mathbb{P}}(h)| \right] \leq 2 \mathbb{E}_{S \sim \mathbb{P}^N} \left[\widehat{Rad}_S(\mathcal{H}) \right]$$

where $e_{\mathbb{P}}(h) = \mathbb{E}_{z \sim \mathbb{P}}[h(z)]$ and $e_S(h) = \frac{1}{N} \sum_{i=1}^{N} h(z_i)$. This property leads to the following generalization error bound derived from classical concentration bounds: with probability $1 - \delta$ (over the sampling S), for every $h \in \mathcal{H}$:

$$e_S(h) - e_{\mathbb{P}}(h) \leq 2 \widehat{Rad}_S(\mathcal{H}) + 4 \sqrt{\frac{2 \log(4/\delta)}{n}} \quad .$$

Rademacher complexity along with VC-dimension [Vapnik, 1998] are the main tools for deriving generalization bounds. The two concepts are linked and one can upperbound the Rademacher complexity with the VC dimension. Another tool is TODO

2.2 Introduction to Adversarial Classification

In this section, we present the required background about adversarial classification. In the first part, we present formally what is an adversarial attack, then how to craft them in practice. After, we present ways of defending against adversarial examples. Finally, we state the main results about theoretical understanding of adversarial examples.

2.2.1 What is an adversarial example?

In classification tasks, an adversarial example is a perturbation of an input that is imperceptible to humans, but that state-of-the-art classifiers are unable to classify accurately. In the following of the manuscript we define adversarial attacks as follows.

Definition 2. *Let $h : \mathcal{X} \to \mathcal{Y}$ be a classifier. An adversarial attack of level ε on the input x with label y against the classifier h is a perturbation x' such that:*

$$h(x') \neq y \quad \textit{and} \quad d(x, x') \leq \varepsilon \ .$$

This definition is very simple and general. The distance d can refer to an ℓ^p distance, taken as a surrogate to a perception distance. We can associate to adversarial examples a notion of adversarial risk. The adversarial risk is the worst case risk if each point is optimally attacked at level ε.

Definition 3. *Let $\mathbb{P}$ be a Borel distribution over $\mathcal{X} \times \mathcal{Y}$. Let $h : \mathcal{X} \to \mathcal{Y}$ be a classifier. We define the* adversarial risk *of h at level ε as:*

$$\mathcal{R}_\varepsilon(h) := \mathbb{P}\left[\exists x' \in B_\varepsilon(x),\ h(x') \neq y\right] = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{x' \in B_\varepsilon(x)} \mathbf{1}_{h(x')\neq y}\right]$$

where $B_\varepsilon(x) = \{x' \in \mathcal{X} \mid\ d(x, x') \leq \varepsilon\}$. If $f : \mathcal{X} \to \mathbb{R}^K$, then the adversarial risk of f at level ε is defined as

$$\mathcal{R}_\mathbb{P}(f) := \mathbb{P}\left[\exists x' \in B_\varepsilon(x),\ \operatorname*{argmax}_{k\in\mathcal{Y}} f_k(x') \neq y\right]$$

A first property is that the adversarial risk is well defined. While this result seems trivial, it requires advanced arguments from measure theory.

Proposition 1. *Let $\mathbb{P}$ be a Borel distribution over $\mathcal{X} \times \mathcal{Y}$. Let $h : \mathcal{X} \to \mathcal{Y}$ be a classifier. If h is Borel measurable then $(x, y) \mapsto \sup_{x' \in B_\varepsilon(x)} \mathbf{1}_{h(x')\neq y}$ is universally measurable.*

Proof. We define $\phi_\varepsilon(x, y, f) = \sup_{x' \in B_\varepsilon(x)} \mathbf{1}_{h(x)\neq y}$. We have :

$$\phi_\varepsilon(x, y, f) = \sup_{(x',y')\in\mathcal{X}\times\mathcal{Y}} \mathbf{1}_{h(x')\neq y'} - \infty \times \mathbf{1}\{d(x', x) \geq \varepsilon \text{ or } y' \neq y\}$$

Then,

$$((x, y), (x', y')) \mapsto \mathbf{1}_{h(x')\neq y'} - \infty \times \mathbf{1}\{d(x', x) \geq \varepsilon \text{ or } y' \neq y\}$$

defines a measurable, hence upper semi-analytic function. Using [Bertsekas and Shreve, 2004, Proposition 7.39, Corollary 7.42], we get that for all $f \in \mathcal{F}(\mathcal{X})$, $(x, y) \mapsto \phi_\varepsilon(x, y, f)$ is a universally measurable function. □

Similarly to the standard classification setting, we define the optimal bayes risk for adversarial classification.

Definition 4. *Let $\mathbb{P}$ be a Borel distribution over $\mathcal{X} \times \mathcal{Y}$. We call* adversarial Optimal Bayes risk *of level ε , the infimum of adversarial risk of level ε over the set of Borel measurable classifiers $\mathcal{F}(\mathcal{X}, \mathcal{Y})$:*

$$\mathcal{R}^\star_\varepsilon := \inf_{h\in\mathcal{F}(\mathcal{X},\mathcal{Y})} \mathcal{R}_\varepsilon(h)$$

Contrarily to the the standard case, the existence of optimal Bayes classifiers for the adversarial risk is a difficult question.

2.2.2 Casting Adversarial examples

The probably most puzzling about adversarial examples is the facility to craft them. Let us consider an attacker that aim at finding an adversarial perturbation x' of an input x for a given classifier $\mathbf{f}$. In order to craft an adversarial example, typically the cross-entropy, the attacker maximizes the following objective given a differentiable loss L:

$$\max_{x' \in \mathcal{X} \text{ s.t. } d(x,x') \leq \varepsilon} L\left(\mathbf{f}(x'), y\right). \tag{2.4}$$

In this case the attack is said to be *untargeted*, i.e. the classifier aims at evading the label y. On the other side, a *targeted attack* aims at perturbing a label x to make it classify to a target label y. In this case, the attacker objective writes: $\min_{x' \in \mathcal{X} \text{ s.t. } d(x,x') \leq \varepsilon} L\left(\mathbf{f}(x'), y\right)$. An attacker may also target at finding the smallest perturbation problem [Carlini and Wagner, 2017, Moosavi-Dezfooli et al., 2016]. Many attacks were proposed that we will categorize into two parts: white-box attacks and black-box attacks.

White box attacks: In this setting, the attacker has full knowledge of the function $\mathbf{f}$ and its parameters. Hence, these attacks often takes advantages of the differentiablity of { and the loss function L. Then, such attacks usually takes the gradient $\nabla_x L\left(f\left(x^t\right), y\right)$ as ascent direction for crafting adversarial examples. These attacks are called *gradient based attacks*. The most popular white box attacks are PGD attack Kurakin et al. [2016], Madry et al. [2018], FGSM attack [Goodfellow et al., 2015], Carlini&Wagner attack [Carlini and Wagner, 2017], AutoPGD [Croce and Hein, 2020], FAB [Croce and Hein, 2020], etc. As an illustration of the simplicity of crafting adversarial examples, we show hereafter how the desingn of a PGD attack in an ℓ_p case.

Example (PGD attack)**.** *Let $x_0 \in \mathbb{R}^d$ be an input. The projected gradient descent (PGD) Kurakin et al. [2016], Madry et al. [2018] of radius ε, recursively computes*

$$x^{t+1} = \prod_{B_p(x,\varepsilon)} \left(x^t + \alpha \underset{\delta \ s.t. \ ||\delta||_p \leq 1}{\operatorname{argmax}} \langle \Delta^t, \delta \rangle \right)$$

where $B_p(x, \varepsilon) = \{x + \tau \ s.t. \ \|\tau\|_p \leq \epsilon\}$, $\Delta^t = \nabla_x L\left(f\left(x^t\right), y\right)$, α is a gradient step size, and $\prod_S$ is the orthogonal projection operator on S. Many attacks are extensions of this one as AutoPGD [Croce et al., 2020b] and SparsePGD [Tramèr and Boneh, 2019]

Black box attacks: In this setting, the attacker has limited knowledge of the classifier. The attacker does not have have access to the parameters of the classifier, but can query either the predicted logits or the predicted label for a given input x. To craft adversarial examples, it was proposed to mimic gradient-based attacks using gradient estimation as the ZOO attack [Chen et al., 2017] and NES attack [Ilyas et al., 2018a, 2019]. Attacks might also be based on other optimization methods such as combinatorial methods [Moon et al., 2019] or evolutionary computation [Andriushchenko et al., 2019].

Adversarial Examples beyond Image Classification. Adversarial examples do not only exist in Image Classification, although it is the most spectacular example as images are perceptually unchanged. We can enumerate, non exhaustively, the following examples of adversarial classification:

- **Image Segmentation and Object Detection:** Xie et al. [2017] proposed to attack image segmentation and object detection. The goal of such attack is enforce a undesirable detection or segmentation in an image.

- **Video classification:** Videos are series of images. Adversarial attacks against video classification systems are closed to adversarial examples in standard Image Classification. Adversarial attacks might aim at changing either a bit many frames [Jiang et al., 2019] or a lot only a few frames [Mu et al., 2021].
- **Audio systems:** Audio systems can be fooled by adding inaudible adversarial noise to an audio file [Carlini and Wagner, 2018]. These attacks raise issues in the massive use of personal vocal assistants [Zhang et al., 2019b].
- **NLP classification tasks:** Adversaries change some words in a text to make it misclassified. However such examples can also change the meaning of the text and consequently change its classification also humans. Examples of attempts for adversarial examples against NLP systems can be either black box [Jin et al., 2019, Li et al., 2020a] or gradient-based [Guo et al., 2021]
- **Recommender Systems** A recent line of work Garcelon et al. [2020], Jun et al. [2018], Liu and Shroff [2019] aimed at crafting adversarial attacks against bandit algorithms [Lattimore and Szepesvári, 2018]. The goal of these attacks are to force the learner to chose the wrong arms a linear number of times. While these works are mostly theoretical, their potential use in practical settings might raise issues for businesses in a close future.

2.2.3 Defending against adversarial examples

Defending against adversarial examples is still an open research questions with few answers to it. One can derive the methods in two categories: empirical defenses and provable defenses.

Provable defenses. A defense is said to be provable if there is a theoretical guarantee to ensure a level of robustness. Formally, a classifier h is said to *certifiably robust at level* ε at input x with label y if there exist no adversarial example of level ε on h at the point (x, y), i.e. for all x' such that $d(x, x') \leq \varepsilon$, $h(x') = y$. Researchers have focused on finding ways to certify robustness. The first categories of defenses relies on convex relaxation of layers [Wong and Kolter, 2018, Wong et al., 2018]. It consists to consider a convex outer approximation of the set of activations reachable through a norm-bounded perturbation of an input. In the case of ReLU activation, the robust optimization problem that minimizes the worst case loss over this outer region writes as a linear program. Another developed method is noise injection to the input [Cohen et al., 2019, Lecuyer et al., 2019, Pinot et al., 2019, Salman et al., 2019]. By adding a noise, the inputs can be seen as distributions. The certificates are derived by determining which classifier would be the most powerful to distinguish two inputs. This idea is closely related to the notions of statistical tests [Cohen et al., 2019], information theory [Pinot et al., 2019] and differential privacy [Lecuyer et al., 2018]. Finally, a last trend to develop provably robust neural networks is to enforce Lipschitzness property [Tsuzuku et al., 2018]. Many papers hav worked on designing Lipschitz layers [Li et al., 2019a, Singla and Feizi, 2021, Trockman et al., 2021] and activations [Anil et al., 2019, Huang et al., 2021b, Singla et al., 2021a].

Empirical defenses. Defenses against adversarial examples often have no theoretical guarantees and based on training heuristics. The first defense that was proposed is *Adversarial Training* [Goodfellow et al., 2015, Madry et al., 2018]. This defense is an heuristic to minimize the adversarial risk. We describe the adversarial training defense in Algorithm 1 to training a classifier f_θ parametrized by θ. It consists minimization steps and attacks on the classifier to make it more robust. To our knowledge there exists no proof of convergence for this defense. Many other empirical defenses are variants of Adversarial Training as TRADES [Zhang et al.,

Algorithm 1: Adversarial Training algorithm

T: number of iterations, Level of attack ε
for $t = 1, \dots, T$ **do**
  Let B_t be a batch of data.
  $\tilde{B}_t \leftarrow$ Attack of level ε on images in B_t for the model f_{θ_t} (using PGD for instance)
  $\theta_k^t \leftarrow$ Update θ_k^{t-1} with $\tilde{B}_t$ with a SGD or Adam step
end

2019a] or MART [Wang et al., 2019]. For instance, TRADES aims at minimizing the following objective:

$$f \mapsto \mathbb{E}\left[L(f(x), y) + \lambda \times \max_{x' \in B_\varepsilon(x)} L(f(x'), f(x))\right] \quad .$$

The first term aims at optimizing standard robustness and the second term is a regularization for adversarial robustness. The objective is to better balance the tradeoff between robustness and standard accuracy. Similarly to Adversarial Training, the inner supremum is optimized using PGD algorithm.

Another promising way to defend against adversarial examples is to augment the dataset. For instance, Carmon et al. [2019], Rebuffi et al. [2021] proposed to use unlabeled data to improve Adversarial Training strategies. Other works as [Wang et al., 2019] proposed to use artificially generated inputs to improve adversarial robustness. We do not enter into details of these but most powerful defenses uses one of these techniques [Croce et al., 2020a].

Evaluation Protocol. Unless the used defense mechanisms are provable and provide guarantees, evaluating and assessing adversarial robustness is rigorous and meticulous task for empirical defenses. For instance, many papers introduced "defenses" that were actually proven to be "false" [Athalye et al., 2018a, Carlini et al., 2019]. Indeed, when proposing a defense, one needs to adapt the attack model to the defense. We describe the following common issues. For instance, when evaluating against randomized classifiers in either white-box or black-box setting, the return output is a random variable, hence the computation of an attack against it needs to be adapted to the non-deterministic nature of the classifier. To do so, Athalye et al. [2018a] proposed to average either the logits or the gradient of the classifier to build a suitable attack against a randomized classifier. This procedure was called Expectation Over Transformation (EOT). A second example is defenses that aims at using non differentiable activation functions as Heaviside functions. Athalye et al. [2018b] proposed to use BPDA (xxx), i.e. differentiable approximations to circumvent the "defense". Black-box attacks are also a way to build efficient attacks in this case.

To answer the need of adversarial examples research community to evaluate accurately their models against adversarial examples, Croce et al. [2020a] proposed RobustBench as a unified platform for benchmarking adversarial defenses. The platform evaluates models on different black-box and white-box, targeted and untargeted attacks (AutoPGD [Croce et al., 2020b], FAB [Croce and Hein, 2020], SquareAttack [Andriushchenko et al., 2019]). However, this platform has its limitations: for instance, it does not propose to evaluate the robustness of randomized classifiers.

State-of-the-art in Image Classification To evaluate the perfomance of an attack of a classification algorithm, one needs to train and evaluate on datasets. In image classification

evaluation, three datasets are mainly used:

- **MNIST [LeCun]:** A dataset of black and white low-quality images representing the 10 digits. The training set contains 50000 images and test set 10000 images. These images are of dimension $28 \times 28 \times 1$ (784 in total). This dataset is known to be easy ($> 99\%$ can be obtained using simple classifiers). In adversarial classification, the problem is also easy to be solved. Evaluation on MNIST is not sufficient to assess the performance of a classifier or even a defense against adversarial examples.
- **CIFAR10 and CIFAR100 [Krizhevsky and Hinton, 2009]:** Datasets of colored low-quality images representing the 10 labels and 100 labels for respectively CIFAR10 and CIFAR100. Each training set contains 50000 images and test set 10000 images. These images are of dimension $32 \times 32 \times 1$ (3072 in total). The current state-of-the-art on CIFAR10 in standard classification is $> 99\%$ of accuracy, but asks advanced methods to reach such a score. On CIFAR100, the current state-of-the-art is around 94%. In adversarial classification both datasets are challenging and difficult. The evolution of state-of-the-art in adversarial classification is available in RobustBench[1]. Benchmark in adversarial classification are often made on these datasets.
- **ImageNet [Deng et al., 2009]:** ImageNet refers to a dataset containing 1.2 million of images labeled into 1000 classes. Images are of diverse qualities, but models often takes input of dimension $224 \times 224 \times 3$ (dimension 150528 in total). The current state-of-the-art on ImageNet is about 87%. There is no need to say that adversarial classification on ImageNet is still a very-challenging task. Further than the standard dataset, ImageNet project is still in development: the project gathers 14197122 images and 21841 labels on August 31th, 2021.

2.2.4 Theoretical knowledge in Adversarial classification

Curse of dimensionality. From the seminal paper on adversarial examples on deep learning systems [Szegedy et al., 2014], the input dimension has been considered as an argument for inevitability of adversarial attacks. To assess this intuition, Gilmer et al. [2018], Shafahi et al. [2018] proved that for a wide range of distribution $\mathbb{P}$ on the unit sphere of dimension d, and any classifier h it is possible to find an attack on examples x with high probability, exponentially depending on the dimmension d on $\mathbb{P}$. The arguments relies on isoperimetric inequalities and was extended to log-concave distributions on Riemannian manifolds and uniform distribution over positively curved Riemannian manifolds [Dohmatob, 2019].

[Simon-Gabriel et al., 2019] also tried to explain the exitence of adversarial examples for neural networks under the light of the high dimensionality of inputs. The authors assumed that neetworks have ReLU activations and that the distributions of weight are Gaussian. Under such hypothesis, they proved that the gradient norm with regards the input is highly dependent on the dimension of the input, then justifying again that the dimensionality of the input is a reason for existence of adversarial examples.

Generalization Bounds in Adversarial Learning. Similarly to the standard classification case, research have focused on computing uniform bounds for adversarial classification. These works are often inspired from generalizations of standard tools as VC-dimension [Cullina et al., 2018] or Rademacher complexity [Awasthi et al., 2020, Khim and Loh, 2018, Yin et al., 2019] is the adversarial case. They exhibit generalization bounds that are highly dependent on the dimension of the input. Indeed the Rademacher complexity for classes adapted to the adversarial

[1] `https://robustbench.github.io/`

case add a polynomial term in the dimension d of the input. However, for randomized classifiers, it is difficult to adapt PAC-Bayes bounds to the adversarial setting [Viallard et al., 2021]. Indeed, the proof schemes cannot be used in the adversarial setting. Moreover, there is still misunderstanding in the bias-complexity tradeoff in the adversarial case [Wang et al., 2018].

Adversarial Bayes Risk. The adversarial bayes risk has been studied only very recently by researchers. Bhagoji et al. [2019], Pydi and Jog [2021a], Trillos and Murray [2020] expressed the adversarial risk as an optimal transport problem for a suitable cost. Another approach was to study the adversarial risk from a game theoretic perspective. We will explain in details these contributions in Section 3.1.1.

One of the recent contributions is the existence of optimal classifiers for the adversarial setting. The problem is not trivial because of the inner supremum and the difficulty to define a suitable topology on the space of measurable functions. The two papers [Awasthi et al., 2021b, Bungert et al., 2021] propose two different approaches for proving the existence of Bayes classifiers. Bungert et al. [2021] proposed a $L^1 + TV$ decomposition [Chan and Esedoglu, 2005] of the adversarial risk. To this end, the authors introduced a non-local perimeter satisfying the submodularity property. They got interested in a suitable relaxation of the adversarial with ν essential supremum where ν is a well-chosen distribution. This allows to study the problem in $L^\infty(\mathcal{X}, \nu)$. The properties of this relaxation are nice (i.e. compactness and semi-continuity) which allows the authors to prove the existence of a minimizer for the relaxed problem. From this solution, the authors build a solution to the the adversarial problem that is Borel-measurable. The authors studied the regularity properties of these minimizers.

TODO: explain other paper

2.3 Game Theory in a Nutshell

Game theory studies strategic interactions among agents assuming their actions are rational. It has many applications in social science [Moulin, 1986] and more recently in machine learning [Goodfellow et al., 2014] for instance. In this section, we recall main concepts in game theory that will help us better understanding the problem of adversarial examples.

2.3.1 Two-player zero-sum games

An important subclass of game theoretic problems are two-person zero-sum games. In such a game there are two players namely Player 1 and Player 2 with opposite objectives. When Player 1 plays an action x in some space $\mathcal{A}_1$ and Player 2 players an action y in some space $\mathcal{A}_2$, Player 1 receives a reward $u_1(x, y)$ (also named utility) and Player 2 receives a reward $u_2(x, y) = -u_1(x, y)$. The objective for each player is to find what is the best strategy to play against the other player to maximize their utility. These strategies are of two types:

- **deterministic strategies:** the player plays a strategy x (for Player 1) or y (for Player 2).
- **mixed strategies:** the player pick up x (for Player 1) or y (for Player 2) randomly according to some probability distribution μ. In this case, the utility functions are averaged according to the strategies μ and ν for respectively Player 1 and Player 2. The average reward of the Player 1 is then $\mathbb{E}_{x\sim\mu, y\sim\nu}[u_1(x, y)]$.

An important matter is the order of play in the game: the strategies might be different if the player know what was the action of the player before him. This leads us to the notion of best response. Assume that a mixed strategy μ was played by Player 1, then the set

of best responses for Player 2 to Player 1 strategy is a strategy that maximizes the utility: $\arg\max_\nu \mathbb{E}_{x\sim\mu, y\sim\nu}[u_1(x,y)]$. We denote this set $BR_2(\mu)$. Game theory aims at studying and computing the nature of strategies in response to other players strategies.

2.3.2 Equilibria in two-player zero-sum games

In game theory, optimal strategies for players are studied under the name of equilibria. Depending on the game, we might have interest in two types of equilibria: Nash equilibria where players do not cooperate and have to choose a strategy simultaneously, and Stackelberg equilibria where a player defines its strategy before the other one. We only focus on two-player zero-sum game.

Nash Equilbria. In a Nash equilibrium, each player is assumed to know the equilibrium strategies of the othe player, and no player has anything to gain by changing only their own strategy. In other words, it is the strategy a rational player should adopt without any cooperation with the other. Note that the existence of Nash equilibrium is not always guaranteed. Formally, a Nash equilibrium is a tuple of actions $(x^\star, y^\star)$ for Players 1 and 2 such that for all other actions x for Player 1 and y for Player 2 we have:

$$u_1(x^\star, y^\star) \geq u_1(x, y^\star) \text{ and } u_2(x^\star, y^\star) \geq u_2(x^\star, y)$$

Note that here the strategies can be either mixed or deterministic. In a two-player zero-sum game we can restate the previous condition as

$$u_1(x, y^\star) \leq u_1(x^\star, y^\star) \leq u_1(x^\star, y)$$

We remark that a Nash equilbrium is defined as a best response to each other strategy, i.e. $(x^\star, y^\star)$ is a Nash equilibrium if and only if $x^\star \in BR_1(y^\star)$ and $y^\star \in BR_2(x^\star)$. We can then come to a necessary and sufficient condition for the existence of Nash equilibria in the case of a two-player zero-sum game:

$$\max_x \min_y u_1(x,y) = \min_y \max_x u_1(x,y)$$

It is a strong duality condition on the function u_1, with the additional property that the optima are attained. If there is duality but the optima are not attained, we can state the existence of δ-approximate Nash equilbria for every $\delta > 0$, i.e. (x^δ, y^δ) such that:

$$u_1(x^\delta, y^\delta) \geq u_1(x, y^\delta) - \delta \text{ and } u_2(x^\delta, y^\delta) \geq u_2(x^\delta, y) - \delta$$

Stackelberg Equilibria. A Stackelberg game is a game where Player 1 defines its strategy before Player 2. Stackelberg equilibria are a tuple of optimal strategies for each player. As Player 1 needs to define its strategy before Player 2, the strategy $x^\star$ of Player 1 has to maximize $\min_y u_1(x,y)$. The strategy for Player 2 is then just to play an action that maximizes its utility given that Player 1 played $x^\star$. In other words, he has to choose a best response to $x^\star$. Note that if $(x^\star, y^\star)$ is a Nash equilibrium then it is also a Stackelberg equilibrium.

2.3.3 Strong Duality Theorems

Finite action sets. In a two-player zero-sum game where the actions space is finite for both players, the rewards can be casted in a matrix $A \in R^{n\times m}$ where $A_{ij} = u_1(x_i, y_j)$. In this case,

Von Neumann [Von Neumann, 1937] proved that there always exists a mixed equilibrium. A mixed strategy of n actions can be embedded in the probability simplex:

$$\Delta_n := \left\{ (p_1, \dots, p_n) \in \mathbb{R}_+^n | \sum_{i=1}^n p_i = 1 \right\}$$

Theorem 1 (Von Neumann's Theorem [Von Neumann, 1937])**.** *Let $A \in R^{n \times m}$ then:*

$$\max_{x \in \Delta_n} \min_{y \in \Delta_m} x^T A y = \min_{y \in \Delta_m} \max_{x \in \Delta_n} x^T A y$$

Infinite action sets. For infinite action sets, Von Neumann's Theorem is usually not sufficient. There are two main extensions with different hypotheses, namely Sion's Theorem [Sion, 1958] and Fan's Theorem [Fan, 1953].

Theorem 2 (Sion's Theorem [Sion, 1958])**.** *Let X be a compact convex set and Y be a convex set of a linear topological space. Let $u : X \times Y \to \mathbb{R}$ be a function such that for all $y \in Y$, $u(\cdot, y)$ is quasi-concave and upper semi-continuous; and for all $x \in X$, $u(x, \cdot)$ is quasi-convex and lower semi-continuous, then:*

$$\max_{x \in X} \inf_{y \in Y} u(x, y) = \inf_{y \in Y} \max_{x \in X} u(x, y)$$

Moreover, if Y is compact, then the infimum is attained.

Note that a function is said to be *quasi-convex* if it lower level sets are convex sets. In particular, convex functions are quasi-convex.

Theorem 3 (Fan's Theorem [Fan, 1953])**.** *Let X be a compact convex set and Y be a convex set (not necessarily topological). Let $u : X \times Y \to \mathbb{R}$ be a function such that for all $y \in Y$, $u(\cdot, y)$ is concave and upper semi-continuous; and for all $x \in X$, $u(x, \cdot)$ is convex, then:*

$$\max_{x \in X} \inf_{y \in Y} u(x, y) = \inf_{y \in Y} \max_{x \in X} u(x, y)$$

Moreover, if Y is compact and for all $x \in X$, $u(x, \cdot)$ is lower semi-continuous, the infimum is attained.

The hypotheses are close since both concerns convexity or quasi convexity of the reward function and the semi-continuity of the partial reward. The differences are subtle and there are cases where one may use either Sion's or Fan's Theorem. For infinite action sets, it is usual to consider mixed strategies as probability distributions on X or Y. In this case, we often endow $\mathcal{M}_+^1(\mathcal{X})$ and $\mathcal{M}_+^1(\mathcal{Y})$ with the weak-$*$ (or narrow) topology of measures and use Sion's or Fan's Theorem directly on these probability spaces.

2.4 Optimal Transport concepts

Optimal Transport have gained interest in Machine Learning applications during the past years. Indeed, Optimal Transport has the ability to model many problems as Generative Adversarial Networks [Arjovsky et al., 2017], and in Adversarial Learning [Bhagoji et al., 2019, Pydi and Jog, 2021a, Sinha et al., 2017]. In particular, it will be a central tool in this thesis with the notion of distributionally robust optimisation introduced in Section 3.1.2. The computation methods for optimal transport problems have also been considerably improved recently. Originally introduced by Monge, this Optimal Transport was a problem where the aim was to move some quantity x

to some places y while minimizing the total cost of transport. Let $\mathcal{Z}$ be a Polish space. Let $\mathbb{P}$ and $\mathbb{Q}$ be two Borel probability distributions over $\mathcal{Z}$ and $c : \mathcal{Z}^2 \to \bar{\mathbb{R}}_+$ be a non-negative function. Formally, the problem was posed as follows:

$$\inf_{T|\ T_\sharp \mathbb{P}=\mathbb{Q}} \mathbb{E}_{z\sim\mathbb{P}}\left[c(z, T(z))\right]$$

where T is a measurable mapping. The main problem with the previous problem, is that there may exist no mapping from $\mathbb{P}$ to $\mathbb{Q}$, for instance when $\mathbb{P}$ is a single dirac and $\mathbb{Q}$ support contains more than two points. To overcome this issue, Kantorovich proposed to interest in couplings in mappings. Formally couplings between distributions are defined as follows.

Definition 5 (Couplings between distributions)**.** *Let $\mathcal{Z}$ be a Polish space. Let $\mathbb{P}$ and $\mathbb{Q}$ be two Borel probability distributions over $\mathcal{Z}$. The set of coupling distributions between $\mathbb{P}$ and $\mathbb{Q}$ is defined as:*

$$\Gamma_{\mathbb{P},\mathbb{Q}} := \left\{\gamma \in \mathcal{M}_+^1(\mathcal{Z}^2) \mid\ \Pi_{1,\sharp}\gamma = \mathbb{P},\ \Pi_{2,\sharp}\gamma = \mathbb{Q}\right\}$$

where $\Pi_{i,\sharp}$ represents the push-forward of the projection on the i-th component.

Setting this definition, one can define a well-posed version of the Monge problem, often referred to Kantorovich problem.

Definition 6 (Optimal Transport)**.** *Let $\mathcal{Z}$ be a Polish space. Let $c : \mathcal{Z}^2 \to \bar{\mathbb{R}}_+$ be a lower semi-continuous non-negative function. Let $\mathbb{P}$ and $\mathbb{Q}$ be two Borel probability distributions over $\mathcal{Z}$. The Optimal Transport problem or Wasserstein problem between $\mathbb{P}$ and $\mathbb{Q}$ associated with cost function c is defined as:*

$$W_c(\mathbb{P},\mathbb{Q}) := \inf_{\gamma\in\Gamma_{\mathbb{P},\mathbb{Q}}} \int c(x,y)d\gamma(x,y) = \inf_{\gamma\in\Gamma_{\mathbb{P},\mathbb{Q}}} \mathbb{E}_{(x,y)\sim\gamma}\left[c(x,y)\right]$$

A clear introduction to this problem can be found in Villani [2003]. In particular, it was proved that the infimum is attained. When $\mathcal{X}$ is endowed with ground metric d, one can endow the space of probability distributions with bounded p-moments with a metric named the Wasserstein-p metric defined as:

$$D_p(\mathbb{P},\mathbb{Q}) := \inf_{\gamma\in\Gamma_{\mathbb{P},\mathbb{Q}}} \mathbb{E}_{(x,y)\sim\gamma}\left[d^p(x,y)\right]^{1/p}$$

With this metric, the space of probability distributions with bounded p-moments metrizes the weak topology of measures. When $p = \infty$, the D_∞ be be defined in the limit as:

$$D_\infty(\mathbb{P},\mathbb{Q}) := \inf_{\gamma\in\Gamma_{\mathbb{P},\mathbb{Q}}} \gamma - \operatorname*{ess\,sup}_{(x,y)} d(x,y)$$

The Wasserstein-∞ metric can be extended to other costs and will be denoted $W_{\infty,c}$.

Entropic Regularized Optimal Transport. The computation time of the exact Optimal Transport solution is often prohibitive: the complexity is supercubic in the number of samples in the empirical distributions. Cuturi [2013], Peyré et al. [2019] proposed an entropic regularization of Optimal Transport to accelerate the computation, which writes

$$W_c^\varepsilon(\mathbb{P},\mathbb{Q}) := \inf_{\gamma\in\Gamma_{\mathbb{P},\mathbb{Q}}} \int c(x,y)d\gamma(x,y) = \inf_{\gamma\in\Gamma_{\mathbb{P},\mathbb{Q}}} \mathbb{E}_{(x,y)\sim\gamma}\left[c(x,y)\right] + \varepsilon \times KL(\gamma||\mathbb{P}\otimes\mathbb{Q})$$

where KL is the Kullback Leibler Leibler divergence defined as $\mathrm{KL}(\mu||\nu) = \int \log\frac{d\mu}{d\nu}d\mu + \int d\nu - \int d\mu$ if $\mu \ll \nu$, and $+\infty$ otherwise. To solves this problem, Cuturi [2013] proposed to use Sinkhorn iterations which considerably accelerate the computation of an approximate solution to the optimal transport problem.

Kantorivch Duality. A fundammental theorem in Optimal Transportation is the Kantorovich duality theorem as follows.

Theorem 4 (Kantorovich duality)**.** *Let $\mathcal{Z}$ be a Polish space. Let $c : \mathcal{Z}^2 \to \bar{\mathbb{R}}_+$ be a lower semi-continuous non-negative function. Let $\mathbb{P}$ and $\mathbb{Q}$ be two Borel probability distributions over $\mathcal{Z}$. Then the following strong duality theorem holds:*

$$W_c(\mathbb{P}, \mathbb{Q}) = \sup_{f,g \in C(\mathcal{Z}),\ f \oplus g \leq c} \int f d\mathbb{P} + \int f d\mathbb{Q}$$

where for all $x, y \in \mathcal{Z}$, $f \oplus g(x, y) := f(x) + g(y)$.

One can find a proof of this result in [Villani, 2003]. The main arguments are that the dual of continuous functions on a compact space is the space of Radon measures, and the Rockafellar duality theorem. We can also mention its entropic regularized version.

Theorem 5 (Kantorovich duality)**.** *Let $\mathcal{Z}$ be a Polish space . Let $c : \mathcal{Z}^2 \to \bar{\mathbb{R}}_+$ be a lower semi-continuous non-negative function. Let $\mathbb{P}$ and $\mathbb{Q}$ be two Borel probability distributions over $\mathcal{Z}$. Then the following strong duality theorem holds:*

$$W_c(\mathbb{P}, \mathbb{Q}) = \sup_{f,g \in C(\mathcal{Z})} \int f d\mathbb{P} + \int f d\mathbb{Q} - \varepsilon \left(\int e^{\frac{f(x)+g(y)-c(x,y)}{\varepsilon}} d\mu(x) d\nu(y) - 1 \right)$$

where for all $x, y \in \mathcal{Z}$, $f \oplus g(x, y) := f(x) + g(y)$.

Chapter 3

Related Work

Contents

3.1 A game theoretic approach to adversarial classification

While adversarial classification can be naturally understood as a game between the attacker and the classifier, it has only been very recent that the problem has been studied from a game theoretic perspective. Adversarial examples have been studied under the notions of Stackelberg game in Brückner and Scheffer [2011], and zero-sum game in Bose et al. [2021], Perdomo and Singer [2019], Rota Bulò et al. [2017].

In [Bose et al., 2021], the authors consider a setting with a convex loss function $L : \mathbb{R}^k \times \mathcal{Y} \to \mathbb{R}$, a convex set of deterministic classifiers $\mathcal{H}$ and a generative attacker $g : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ (i.e. a measurable function) such that:

$$d(g(x, y, z), x) \leq \varepsilon$$

for all x, y, z and z is sampled from a latent distribution p_z. The sets of such functions g is denoted G_ε. In this setting the authors show there is no duality gap for the game between the attacker and the learner:

$$\min_{f\in\mathcal{H}} \max_{g\in G_\varepsilon} \mathbb{E}_{(x,y)\sim\mathbb{P},z\sim p_z} \left[L\left(f(g(x, y, z), y)\right] = \max_{g\in G_\varepsilon} \min_{f\in\mathcal{H}} \mathbb{E}_{(x,y)\sim\mathbb{P},z\sim p_z} \left[L\left(f(g(x, y, z), y)\right]\right.\right.$$

However, this setting is limited due to the convexityassumptions. As we will see in Chapter 5, one can prove that no convex loss can be a good surrogate for the 0/1 loss in the adversarial setting. The goal of the paper is to build a framework to design new zero-shot black-box adversarial attacks from generative attackers. Such an attack is called a *No Box attack*.

Pinot et al. [2020] proposed to study the adversarial attacks problem from a game theoretic point of view. The authors proposed to treat the case of binary classification with 0/1 loss where the classifier can be either allow to deterministically play a continuous function or randomly chose a continuous function. In game theoretic terminology, the classifier can play mixed strategies of continuous functions. On the other side, the attacker is deterministic. Formally, its set of actions is:

$$\mathcal{F}_\varepsilon = \{f \in \mathcal{F}(\mathcal{X}\times\mathcal{Y}, \mathcal{X}\times\mathcal{Y}) \mid \forall (x,y) \in \mathcal{X}\times\mathcal{Y},\ \|f_1(x,y) - x\| \leq \varepsilon \text{ and } f_2(x,y) = y\}$$

In their work, the authors also assume that the attacker suffers a regularization. the first considered regularization penalizes the average perturbation for the attacker:

$$\Omega(f) = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\|x - f_1(x,y)\|\right]$$

The second one penalizes the attacker if he attacks "too many points":

$$\Omega(f) = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\mathbf{1}_{x\neq f_1(x,y)}\right]$$

Given one of these regularization, the score function for the classifier h and an attacker f, is defined as:

$$\mathbb{E}_{\mathbb{P}}\left[L(h(f(x)), y)\right] - \lambda \times \Omega(f)$$

where λ is a non negative constant. In this setting, the authors show that there do not exist a pure Nash Equilibrium. In particular, the risk for randomized classifiers is stricly smaller than the risk for deterministic classifiers. The question of the nature of equilibria was remained open.

3.1.1 Adversarial Risk Minimization and Optimal Transport

Optimal Transport is a key element when studying Adversarial Classification problems. Let $\mathbb{P}$ be a distribution on the input-label space $\mathcal{X}\times\mathcal{Y}$. We recall that the problem of adversarial risk minimization is defined as

$$\mathcal{R}^\star_{\varepsilon,\mathbb{P}} = \inf_h \mathbb{P}_{(x,y)}\left[\exists x' \in B_\varepsilon(x),\ h(x') \neq y\right]$$

A recent line of work [Bhagoji et al., 2019, Pydi and Jog, 2021a, Trillos and Murray, 2020] draw important links between $\mathcal{R}^\star_{\varepsilon,\mathbb{P}}$ and Optimal Transport problems in the case of binary classification ($\mathcal{Y} = \{-1, +1\}$) the space $\mathcal{X}$ satisfy a midpoint property, i.e. for all $x_1, x_2 \in \mathcal{X}$ there exist $x \in \mathcal{X}$ such that $d(x, x_1) = d(x, x_2) = \frac{d(x_1,x_2)}{2}$. It was shown that in this case:

$$\mathcal{R}^\star_{\varepsilon,\mathbb{P}} = \frac{1}{2} - \frac{1}{2} W_{c_\varepsilon}(\mathbb{P}, \mathbb{P}^S)$$

where $\mathbb{P}^S := T^S_\sharp \mathbb{P}$ with $T^S(x,y) = (x,-y)$ and

$$c_\varepsilon\left((x,y),(x',y')\right) = \mathbf{1}_{d(x,x')>2\varepsilon, y\neq y'}$$

Note that T^S only switches the label of pair (x,y). When $\varepsilon = 0$, $W_{c_\varepsilon}(\mathbb{P}, \mathbb{P}^S)$ equals the total variation distance between $\mathbb{P}$ and $\mathbb{P}^S$, which was a result proved in [Trillos and Murray, 2020]. While this property does not have practical properties yet, there is a hope that this relation might help at building more robust classifiers to adversarial examples.

3.1.2 Distributionally Robust Optimization

Another close link between adversarial attacks and Optimal Transport can be made under the light of distributionally robust optimization problems. Let $\mathcal{Z}$ and Θ be Polish spaces. Let $\mathbb{P}$ be a Borel probability distribution over $\mathcal{Z}$. Let $f : \Theta \times \mathcal{Z} \to \mathbb{R}$ be an upper semi continuous function in its second variable. Let us consider the following problem:

$$\min_{\theta \in \Theta} \mathbb{E}_{z \sim \mathbb{P}} [f(\theta, z)] = \min_{\theta \in \Theta} \int f(\theta, z) d\mathbb{P}(z) \tag{3.1}$$

This problem can typically be a risk minimization problem in Machine Learning when $\mathbb{P}$ is a distribution over input-label pairs and Θ is a parameter space for the classifier. A distributionally robust optimization (DRO) problem is a problem similar to Equation (3.1), but the learner aims at being robust to a change in the distribution $\mathbb{P}$. Typically if D is an uncertainty metric for distribubtions. Formally, the DRO problem is casted as follows:

$$\min_{\theta \in \Theta} \sup_{\mathbb{Q} \in \mathcal{M}_+^1(\mathcal{Z}) | \ D(\mathbb{P}, \mathbb{Q}) \leq \varepsilon} \mathbb{E}_{z \sim \mathbb{Q}} [f(z)]$$

For instance, D be a Kullback-Leibler divergence or other f-divergences [Duchi et al., 2016, Namkoong and Duchi, 2016], total variation distances [Jiang and Guan, 2018, Rahimian et al., 2019] or optimal transport distances [Blanchet and Murthy, 2019, Raghunathan et al., 2018, Shafieezadeh Abadeh et al., 2015].

In the case of Wasserstein uncertainty sets, let $c : \mathcal{Z} \to \bar{\mathbb{R}}_+$ be a lower semi-continuous non-negative function. Then a Wasserstein distributionally robust optimization (DRO) problem is defined as follows:

$$\min_{\theta \in \Theta} \sup_{\mathbb{Q} \in \mathcal{M}_+^1(\mathcal{Z}) | \ W_c(\mathbb{P}, \mathbb{Q}) \leq \varepsilon} \mathbb{E}_{z \sim \mathbb{Q}} [f(z)]$$

Then we can define the Wasserstein balls as

$$\mathcal{B}_c(\mathbb{P}, \varepsilon) := \left\{ \mathbb{Q} \in \mathcal{M}_1^+(\mathcal{Z}) \mid W_c(\mathbb{P}, \mathbb{Q}) \leq \varepsilon \right\}$$

This problem induces an attack on the distribution $\mathbb{P}$. Informally, one can interpret a Wassertein ball as an attacker moving each point x of the distribution $\mathbb{P}$ to a distribution $\mathbb{Q}_x$ so that the average "distance" $\mathbb{E}_{x \sim \mathbb{P}} [\mathbb{E}_{y \sim \mathbb{Q}_x} [c(x, y)]]$ at most equal to ε. With this interpretation, we can start linking the Wasserstein DRO problem to the adversarial learning problem. Indeed in the adversarial attack problem, the attacker is authorized to move each point to another at distance at most ε, i.e. he is authorized a mapping T such that $d(x, T(x)) \leq \varepsilon$ for every x almost surely.

Properties of Wasserstein balls. The Wasserstein balls inherits from nice properties. Since $\mathbb{Q} \mapsto W_c(\mathbb{P}, \mathbb{Q})$ is convex, they are convex sets. Moreover the function $\mathbb{Q} \mapsto W_c(\mathbb{P}, \mathbb{Q})$ is lower semi-continuous for the narrow topology of measures, then the set $\mathcal{B}_c(\mathbb{P}, \eta)$ is closed for the narrow topology too. Concerning the compactness of this set, if $\mathcal{Z}$ is compact then the set $\mathcal{B}_c(\mathbb{P}, \eta)$ is also compact as a closed subset of the compact set $\mathcal{M}_1^+(\mathcal{Z})$. Yue et al. [2020] proved the compactness for l^p distances. In general, compactness is a case by case question.

Duality results The problem of computing DRO solutions is difficult become it concerns optimization over distribution. A strong duality leading to a relaxation of the problem was proved by Blanchet and Murthy [2019]. We state this theorem as follows.

Theorem 6 (Wasserstein DRO duality)**.** *Let $\mathbb{P}$ be a Borel probability distribution over $\mathcal{Z}$. Let $f : \mathcal{Z} \to \mathbb{R}$ be an upper semi continuous function. Let $c : \mathcal{Z} \to \mathbb{R}_+$ be a lower semi-continuous non-negative function.*

$$\sup_{\mathbb{Q} \in \mathcal{M}_+^1(\mathcal{Z}) |\ W_c(\mathbb{P},\mathbb{Q}) \leq \varepsilon} \mathbb{E}_{z\sim\mathbb{Q}}\left[f(z)\right] = \inf_{\lambda \geq 0} \mathbb{E}_{z\sim\mathbb{P}}\left[\sup_{z' \in \mathcal{Z}} f(z') - \lambda c(z, z')\right] + \lambda\varepsilon$$

This theorem was proved by [Blanchet and Murthy, 2019] using similar arguments to Kantorovich duality. The link with the adversarial attack problem is made clearer with this theorem. Indeed $\mathbb{E}_{z\sim\mathbb{P}}\left[\sup_{z'\in\mathcal{Z}} f(z') - \lambda c(z, z')\right]$ is closed to the adversarial attacks problem. We will make a direct link in the Chapter 4.

Adversarial classification as a Wasserstein-∞ DRO problem. The adversarial attack problem was studied under the light of DRO from a statistical point of view [Raghunathan et al., 2018], or to prove that adversarial classification is exactly a Wasserstein-∞ problem with a well-suited cost function [Pydi and Jog, 2021a]. The previous result from [Blanchet and Murthy, 2019] does not directly apply to Wasserstein-∞ distances but can be adapted. The Wasserstein-∞ DRO problem can be understood as follows: each point x of the distribution $\mathbb{P}$ can be moved to a distribution $\mathbb{Q}_x$ so that the worst-case "distance" $c(x, y)$ is smaller that ε. In general, one can prove the following result that proves that the adversarial classification problem is actually a Wasserstein-∞ DRO problem.

Theorem 7 (Duality for Wasserstein-∞ DRO)**.** *Let $\mathcal{Z}$ be a Polish space. Let $\mathbb{P}$ be a probability distribubtion over $\mathcal{Z}$. Let c be a non-negative lower-semicontinuous function over $\mathcal{Z}^2$ and $f : \mathcal{Z} \to \mathbb{R}$ be a Borel measurable function. Then the following strong duality holds*

$$\sup_{\mathbb{Q} |\ W_{\infty,c}(\mathbb{P},\mathbb{Q}) \leq \varepsilon} \mathbb{E}_{z\sim\mathbb{Q}}\left[f(z)\right] = \mathbb{E}_{z\sim\mathbb{P}}\left[\sup_{z'\in\mathcal{Z} |\ c(z,z')\leq\varepsilon} f(z')\right]$$

This result can be found in special case in [Pydi and Jog, 2021a]. For sake of completeness, we provide a proof of the result.

Proof. Let us define:

$$\tilde{f} : (z, z') \in \mathcal{Z}^2 \mapsto f(z') - \infty \times \mathbf{1}_{c(z,z')>\varepsilon} \quad .$$

$\tilde{f}$ is Borel-measurable, hence upper semi-analytic [Bertsekas and Shreve, 2004, Chapter 7]. We then deduce that

$$z \in \mathcal{Z} \mapsto \sup_{z'\in\mathcal{Z}} \tilde{f}(z, z') = \sup_{z'\in\mathcal{Z} |\ c(z,z')\leq\varepsilon} f(z')$$

is universally measurable, hence justifying the definition of the left-end term in the Theorem.

Now let $\mathbb{Q}$ such that $W_{\infty,c}(\mathbb{P}, \mathbb{Q}) \leq \varepsilon$. There exists $\gamma \in \Gamma_{\mathbb{P},\mathbb{Q}}$ such that $c(z, z') \leq \varepsilon$ γ-almost surely. Then we deduce

$$\begin{aligned}\mathbb{E}_{z'\sim\mathbb{Q}}\left[f(z')\right] = \mathbb{E}_{(z,z')\sim\gamma}\left[f(z')\right] &\leq \mathbb{E}_{(z,z')\sim\gamma}\left[\sup_{z'\in\mathcal{Z} |\ c(z,z')\leq\varepsilon} f(z')\right] \\ &\leq \mathbb{E}_{z\sim\mathbb{P}}\left[\sup_{z'\in\mathcal{Z} |\ c(z,z')\leq\varepsilon} f(z')\right]\end{aligned}$$

Hence we deduce that

$$\sup_{\mathbb{Q}|\ W_{\infty,c}(\mathbb{P},\mathbb{Q})\leq\varepsilon} \mathbb{E}_{z\sim\mathbb{Q}}\left[f(z)\right] \leq \mathbb{E}_{z\sim\mathbb{P}}\left[\sup_{z'\in\mathcal{Z}|\ c(z,z')\leq\varepsilon} f(z')\right]$$

Thanks to Bertsekas and Shreve [2004, Proposition 7.50], for any $\delta > 0$, there exists a universally measurable mapping $T : \mathcal{Z} \to \mathcal{Z}$ such that $\tilde{f}(z, T(z)) \geq \sup_{z'\in\mathcal{Z}} \tilde{f}(z, z') - \delta$ for every $z \in \mathcal{Z}$. Defining $\mathbb{Q} = T_\sharp\mathbb{P}$, we get that $W_{\infty,c}(\mathbb{P}, \mathbb{Q}) \leq \varepsilon$ and that:

$$\sup_{\mathbb{Q}|\ W_{\infty,c}(\mathbb{P},\mathbb{Q})\leq\varepsilon} \mathbb{E}_{z\sim\mathbb{Q}}\left[f(z)\right] \geq \mathbb{E}_{z\sim\mathbb{P}}\left[\sup_{z'\in\mathcal{Z}|\ c(z,z')\leq\varepsilon} f(z')\right] - \delta$$

Consequently, we deduce the expected result of the Theorem. □

When the problem is a classification problem (i.e., $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} = [K]$), one can replace f with $L(f(x), y)$ with L a measurable loss function and set the cost c equals to:

$$c((x, y), (x', y')) := \begin{cases} d(x, x') & \text{if } y = y' \\ +\infty & \text{otherwise.} \end{cases}$$

Hence, we recover the Adversarial classification problem using a Wasserstein-∞ DRO problem. We will see in Chapter 4, the geometric and topological properties of this set.

DRO, Game Theory and Adversarial Attacks. Recently, Pydi and Jog [2021b] got interest in the adversarial binary classification game where the attacker can play a randomized strategy in the ∞-Wasserstein ball of radius ε and the classifier is allowed to play any measurable function. In this case the authors proved the existence of Nash Equilibria, meaning that the classifier can be deterministic and optimal and the attacker requires to be "randomized". We will discuss and compare to this work in details after Chapter 4.

3.2 Surrogate losses in the Adversarial Setting

To account for the possibility of an adversary manipulating the inputs at test time, we need to revisit the standard risk minimization problem by penalizing any classification model that might change its decision when the point of interest is slightly changed. Essentially, this is done by replacing the standard (pointwise) 0/1 loss with an adversarial version that mimics its behavior locally but also penalizes any error in a given region around the point on which it is evaluated.

Yet, just like the 0/1 loss, its adversarial counterpart is not convex, which renders the risk minimization difficult. To circumvent this limitation, we take inspiration from the standard learning theory approach which consists in solving a simpler optimization problem where the non-convex loss function is replaced by a convex surrogate. In general, the surrogate loss is chosen to have a property called *consistency* [Bartlett et al., 2006, Steinwart, 2007, Zhang, 2004b], which essentially guarantees that any sequence of classifiers that minimizes the surrogate objective must also be a sequence that minimizes the Bayes risk. In the context of standard classification, a large family of convex losses, called *classifier-consistent*, exhibits this property. This class notoriously includes the hinge loss, the logistic loss and the square loss.

However, the adversarial version of these surrogate losses needs not to have the same consistency properties with respect to the adversarial 0/1 loss. In fact, most existing results in the standard framework rely on a reduction of the global consistency problem to a local point-wise problem,

called *calibration*. However, the same approach is not feasible in the adversarial setting, because the new losses are by nature non-point-wise. Then the optimum for a given input may depend on yet a whole other set of inputs [Awasthi et al., 2021a,c]. Studying the concepts of calibration and consistency in an adversarial context remains an open and understudied issue. Furthermore, this is a complex and technical area of research, that requires a rigorous analysis, since small tweaks in definitions can quickly make results meaningless or inaccurate. This difficulty is illustrated in the literature, where articles published in high profile conferences tend to contradict or refute each other Awasthi et al. [2021a,c], Bao et al. [2020].

Notations. In this section, let us consider a classification task with input space $\mathcal{X}$ and output space $\mathcal{Y} = \{-1, +1\}$. Let $(\mathcal{X}, d)$ be a proper Polish (i.e. completely separable) metric space representing the inputs space. For all $x \in \mathcal{X}$ and $\delta > 0$, we denote $B_\delta(x)$ the closed ball of radius δ and center x. We also assume that for all $x \in \mathcal{X}$ and $\delta > 0$, $B_\delta(x)$ contains at least two points[1]. Let us also endow $\mathcal{Y}$ with the trivial metric $d'(y, y') = \mathbf{1}_{y \neq y'}$. Then the space $(\mathcal{X} \times \mathcal{Y}, d \oplus d')$ is a proper Polish space. For any Polish space $\mathcal{Z}$, we denote $\mathcal{M}_+^1(\mathcal{Z})$ the Polish space of Borel probability measures on $\mathcal{Z}$. We will denote $\mathcal{F}(\mathcal{Z})$ the space of real valued Borel measurable functions on $\mathcal{Z}$. Finally, we denote $\bar{\mathbb{R}} := \mathbb{R} \cup \{\infty, +\infty\}$.

3.2.1 Notions of Calibration and Consistency

The 0/1-loss is both non-continuous and non-convex, and its direct minimization is a difficult problem. The concepts of calibration and consistency aim at identifying the properties that a loss must satisfy in order to be a good surrogate for the minimization of the 0/1-loss. In this section, we define these two concepts and explain the difference between them. First of all, we need to give a general definition of a loss function.

Definition 7 (Loss function)**.** *A loss function is a function $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{F}(\mathcal{X}) \to \mathbb{R}$ such that $L(\cdot, \cdot, f)$ is a Borel measurable for all $f \in \mathcal{F}(\mathcal{X})$.*

Note that this definition is not specific to the standard neither adversarial case. In general, a loss can either depend only on $f(x)$, or on other points related to x (e.g. the set of points within a distance ε of x). We now recall the definition of the risk associated with a loss L and a distribution $\mathbb{P}$.

Definition 8 (L-risk of a classifier)**.** *For a given loss function L, and a Borel probability distribution $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$ we define the risk of a classifier f associated with the loss L and a distribution $\mathbb{P}$ as*

$$\mathcal{R}_{L,\mathbb{P}}(f) := \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[L(x, y, f)\right].$$

We also define the optimal risk associated with the loss L as

$$\mathcal{R}^\star_{L,\mathbb{P}} := \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{L,\mathbb{P}}(f) \quad .$$

Essentially, the risk of a classifier is defined as the average loss over the distribution $\mathbb{P}$. When the loss L is difficult to optimize in practice (e.g when it is non-convex or non-differentiable), it is often preferred to optimize a surrogate loss function instead. In the literature [Bartlett et al., 2006, Steinwart, 2007, Zhang, 2004b], the notion of surrogate losses has been studied as a consistency problem. In a nutshell, a surrogate loss is said to be consistent if any minimizing sequence of classifiers for the risk associated with the surrogate loss is also one for the risk associated with L. Formally, the notion of consistency is as follows.

[1] For instance, for any norm $\|\cdot\|$, $(\mathbb{R}^d, \|\cdot\|)$ is a Polish metric space satisfying this property.

Definition 9 (Consistency). *Let L_1 and L_2 be two loss functions. For a given $\mathbb{P} \in \mathcal{M}_1^+(\mathcal{X} \times \mathcal{Y})$, L_2 is said to be consistent for $\mathbb{P}$ with respect to L_1 if for all sequences $(f_n)_n \in \mathcal{F}(\mathcal{X})^{\mathbb{N}}$:*

$$\mathcal{R}_{L_2,\mathbb{P}}(f_n) \to \mathcal{R}^\star_{L_2,\mathbb{P}} \implies \mathcal{R}_{L_1,\mathbb{P}}(f_n) \to \mathcal{R}^\star_{L_1,\mathbb{P}} \tag{3.2}$$

Furthermore, L_2 is said consistent with respect to a loss L_1 the above holds for any distribution $\mathbb{P}$.

Note that one can reformulate equivalently the previous definition as follows. For all $\epsilon > 0$, there exists $\delta > 0$ such that for every $f \in \mathcal{F}(\mathcal{X})$,

$$\mathcal{R}_{L_2,\mathbb{P}}(f) - \mathcal{R}^\star_{L_2,\mathbb{P}} \leq \delta \implies \mathcal{R}_{L_1,\mathbb{P}}(f) - \mathcal{R}^\star_{L_1,\mathbb{P}} \leq \epsilon$$

Consistency is in general a difficult problem to study because of its high dependency on the distribution $\mathbb{P}$ at hand. Accordingly, several previous works [Bartlett and Mendelson, 2002, Steinwart, 2007, Zhang, 2004b] introduced a weaker notion to study consistency from pointwise viewpoint. The simplified notion is called *calibration* and corresponds to consistency when $\mathbb{P}$ is a combination of Dirac distributions. The main building block in the analysis of the calibration problem is the calibration function, defined as follows.

Definition 10 (Calibration function). *Let L be a loss function. The calibration function $\mathcal{C}_L$ is*

$$\mathcal{C}_L(x, \eta, f) := \eta L(x, 1, f) + (1 - \eta) L(x, -1, f),$$

for any $\eta \in [0, 1]$, $x \in \mathcal{X}$ and $f \in \mathcal{F}(\mathcal{X})$. We also define the optimal calibration function as

$$\mathcal{C}^\star_L(x, \eta) := \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{C}_L(x, \eta, f).$$

Note that for any $x \in \mathcal{X}$ and $\eta \in [0, 1]$, $\mathcal{C}_L(x, \eta, f) = \mathcal{R}_{L,\mathbb{P}}(f)$ with $\mathbb{P} = \eta\delta_{(x,+1)} + (1 - \eta)\delta_{(x,-1)}$. The calibration function thus corresponds then to a pointwise notion of the risk, evaluated at point x. We now define what one means by calibration of a surrogate loss.

Definition 11 (Calibration). *Let L_1 and L_2 be two loss functions. We say that L_2 is* calibrated *with regards to L_1 if for every $\epsilon > 0$, $\eta \in [0, 1]$ and $x \in \mathcal{X}$, there exists $\delta > 0$ such that for all $f \in \mathcal{F}(\mathcal{X})$,*

$$\mathcal{C}_{L_2}(x, \eta, f) - \mathcal{C}^\star_{L_2}(x, \eta) \leq \delta \implies \mathcal{C}_{L_1}(x, \eta, f) - \mathcal{C}^\star_{L_1}(x, \eta) \leq \epsilon.$$

Furthermore, we say that L_2 is uniformly calibrated *with regards to L_1 if for every $\epsilon > 0$, there exists $\delta > 0$ such that for all $\eta \in [0, 1]$, $x \in \mathcal{X}$ and $f \in \mathcal{F}(\mathcal{X})$ we have*

$$\mathcal{C}_{L_2}(x, \eta, f) - \mathcal{C}^\star_{L_2}(x, \eta) \leq \delta \implies \mathcal{C}_{L_1}(x, \eta, f) - \mathcal{C}^\star_{L_1}(x, \eta) \leq \epsilon.$$

Similarly to consistency, one also give a sequential characterization for calibration and uniform calibration: L_2 is *calibrated* with regards to L_1 if for all $\eta \in [0, 1]$, $x \in \mathcal{X}$, for all $(f_n)_n \in \mathcal{F}(\mathcal{X})^{\mathbb{N}}$:

$$\mathcal{C}_{L_2}(x, \eta, f) - \mathcal{C}^\star_{L_2}(x, \eta) \xrightarrow[n\to\infty]{} 0 \implies \mathcal{C}_{L_1}(x, \eta, f) - \mathcal{C}^\star_{L_1}(x, \eta) \xrightarrow[n\to\infty]{} 0 \quad .$$

Also, L_2 is *uniformly calibrated* with regards to L_1 if for all $(f_n)_n \in \mathcal{F}(\mathcal{X})^{\mathbb{N}}$:

$$\sup_{\eta\in[0,1],x\in\mathcal{X}} \mathcal{C}_{L_2}(x, \eta, f) - \mathcal{C}^\star_{L_2}(x, \eta) \xrightarrow[n\to\infty]{} 0 \implies \sup_{\eta\in[0,1],x\in\mathcal{X}} \mathcal{C}_{L_1}(x, \eta, f) - \mathcal{C}^\star_{L_1}(x, \eta) \xrightarrow[n\to\infty]{} 0 \quad .$$

Connection between calibration and consistency. It is always true that calibration is a necessary condition for consistency. Yet there is no reason, in general, for the converse to be true. However, in the specific context usually studied in the literature (i.e., the standard classification with a well-defined 0/1-loss), the notions of consistency and calibration have been shown to be equivalent. [Bartlett et al., 2006, Steinwart, 2007, Zhang, 2004b]. In the next section, we come back on existing results regarding calibration and consistency in this specific (standard) classification setting.

3.2.2 Existing Results in the Standard Classification Setting

Classification is a standard task in machine learning that consists in finding a classification function $h : \mathcal{X} \to \mathcal{Y}$ that maps an input x to a label y. In binary classification, h is often defined as the sign of a real valued function $f \in \mathcal{F}(\mathcal{X})$. The loss usually used to characterize classification tasks corresponds to the accuracy of the classifier h. When h is defined as above, this loss is defined as follows.

Definition 12 (0/1 loss)**.** *Let $f \in \mathcal{F}(\mathcal{X})$. We define the* 0/1 loss *as follows*

$$l_{0/1}(x, y, f) = \mathbf{1}_{y \times sign(f(x)) \leq 0}$$

with a convention for the sign, e.g. $sign(0) = 1$. We will denote $\mathcal{R}_{\mathbb{P}}(f) := \mathcal{R}_{l_{0/1},\mathbb{P}}(f)$, $\mathcal{R}^{\star}_{\mathbb{P}} := \mathcal{R}^{\star}_{l_{0/1},\mathbb{P}}$, $\mathcal{C}(x, \eta, f) := \mathcal{C}_{l_{0/1}}(x, \eta, f)$ and $\mathcal{C}^{\star}(x, \eta) := \mathcal{C}^{\star}_{l_{0/1}}(x, \eta)$.

Note that this 0/1-loss is different from the one introduced by Awasthi et al. [2021a,c], Bao et al. [2020]: they used $\mathbf{1}_{y \times f(x) \leq 0}$ which is an usual 0/1 loss but unadapted to consistency and calibrated study. This loss penalizes indecision: i.e. predicting 0 would lead to a pointwise risk of 1 for $y = 1$ and $y = -1$ while the 0/1 loss $l_{0/1}$ returns 1 for $y = 1$ and 0 for $y = -1$. This definition was used by Awasthi et al. [2021a,c], Bao et al. [2020] to prove their calibration and consistency results. While Bartlett et al. [2006] was not explicit on the choice for the 0/1 loss, Steinwart [2007] explicitly mentions that the 0/1 loss is not a margin loss. The use of this loss is not suited for studying consistency and leads to inaccurate results as shown in the following counterexample. On $\mathcal{X} = \mathbb{R}$, let $\mathbb{P}$ defined as $\mathbb{P} = \frac{1}{2}\left(\delta_{x=0,y=1} + \delta_{x=0,y=-1}\right)$ and $\phi : \mathbb{R} \to \mathbb{R}$ be a margin based loss. The ϕ-risk minimization problem writes $\inf_{\alpha} \frac{1}{2}(\phi(\alpha) + \phi(-\alpha))$. For any convex functional ϕ the optimum is attained for $\alpha = 0$. $f_n : x \mapsto 0$ is a minimizing sequence for the ϕ-risk. However $R_{l_{\leq}}(f_n) = 1$ for all n and $R^{*}_{l_{\leq}} = \frac{1}{2}$. Then we deduce that no convex margin based loss is consistent wrt $l_{\leq}$. Consequently, the 0/1 loss to be used in adversarial consistency needs to be $l_{0/1,\varepsilon}(x, y, f) = \sup_{x' \in B_{\varepsilon}(x)} \mathbf{1}_{y \text{sign}(f(x)) \leq 0}$, otherwise the obtained results might be innacurate.

Some of the most prominent works [Bartlett et al., 2006, Steinwart, 2007, Zhang, 2004b] among them focus on the concept of margin losses, as defined below.

Definition 13 (Margin loss)**.** *A loss L is said to be a* margin loss *if there exists a measurable function $\phi : \mathbb{R} \to \mathbb{R}_{+}$ such that:*

$$L(x, y, f) = \phi(yf(x))$$

For simplicity, we will shortly say that ϕ is a margin loss function and we will denote $\mathcal{R}_{\phi}$ and $\mathcal{C}_{\phi}$ the risk associated with the margin loss ϕ. Notably, it has been demonstrated in several previous worksBartlett et al. [2006], Steinwart [2007], Zhang [2004b] that, for a margin loss ϕ, we have always have $\mathcal{C}^{\star}_{\phi}(x, \eta) = \inf_{\alpha \in \mathbb{R}} \eta\phi(\alpha) + (1 - \eta)\phi(-\alpha)$. This is in particular one of the main observation allowing to show the following strong result about the connection between consistency and calibration.

Theorem 8 (Bartlett et al. [2006], Steinwart [2007], Zhang [2004b]). *Let $\phi : \mathbb{R} \to \mathbb{R}_+$ be a continuous margin loss. Then the three following assertions are equivalent.*

1. *ϕ is calibrated with regards to $l_{0/1}$,*
2. *ϕ is uniformly calibrated $l_{0/1}$,*
3. *ϕ is consistent with regards to $l_{0/1}$.*

Moreover, if ϕ is convex and differentiable at 0, then ϕ is calibrated if and only $\phi'(0) < 0$.

The Hinge loss $\phi(t) = \max(1-t, 0)$ and the logistic loss $\phi(t) = \log(1+e^{-t})$ are classical examples of convex consistent losses. Convexity is a desirable property for faster optimization of the loss, but there exist other non-convex losses that are calibrated as the ramp loss ($\phi(t) = xxx$) or the sigmoid loss ($\phi(t) = (1 + e^t)^{-1}$). In the next section, we present the adversarial classification setting for which Theorem 8 may not hold anymore.

Remark 1. *The equivalence between calibration and consistency is a consequence from the fact that, over the large space of measurable functions, minimizing the loss pointwisely in the input by desintegrating with regards to x is equivalent to minimize the whole risk over measurable functions. This result is very powerful and simplify the study of calibration in the standard setting.*

3.2.3 Calibration and Consistency in the Adversarial Setting.

We now consider the adversarial classification setting where an adversary tries to manipulate the inputs at test time. Given $\varepsilon > 0$, they can move each point $x \sim \mathbb{P}$ to another point x' which is at distance at most ϵ from x[2]. The goal of this adversary is to maximize the 0/1 risk the shifted points from $\mathbb{P}$. Formally, the loss associated to adversarial classification is defined as follows.

Definition 14 (Adversarial 0/1 loss). *Let $\varepsilon \geq 0$. We define the adversarial 0/1 loss of level ε associated as:*

$$l_{0/1,\varepsilon}(x, y, f) = \sup_{x' \in B_\varepsilon(x)} \mathbf{1}_{ysign(f(x)) \leq 0}$$

We will denote $\mathcal{R}_{\varepsilon,\mathbb{P}}(f) := \mathcal{R}^\star_{l_{0/1,\varepsilon},\mathbb{P}}(f)$, $\mathcal{R}^\star_{\varepsilon,\mathbb{P}} := \mathcal{R}^\star_{l_{0/1,\varepsilon},\mathbb{P}}$, $\mathcal{C}_\varepsilon(x, \eta, f) := \mathcal{C}_{l_{0/1,\varepsilon}}(x, \eta, f)$ and $\mathcal{C}^\star_\varepsilon(x, \eta) := \mathcal{C}^\star_{l_{0/1,\varepsilon}}(x, \eta)$ for every $\mathbb{P}$, x, f and η.

Specificity of the adversarial case The adversarial risk minimization problem is much more challenging than its standard counterpart because an inner supremum is added to the optimization objective. With this inner supremum, it is no longer possible to reduce the distributional problem to a pointwise minimization as it is usually done in the standard classification framework. In fact, the notions of consistency and calibration are significantly different in the adversarial setting. This means that the results obtained in the standard classification may no longer be valid in the adversarial setting (e.g., the calibration need not be sufficient for consistency), which makes the study of consistency much more complicated. As a first step towards analyzing the adversarial classification problem, we now adapt the notion of margin loss to the adversarial setting.

Definition 15 (Adversarial margin loss). *Let $\phi : \mathbb{R} \to \mathbb{R}_+$ be a margin loss and $\varepsilon \geq 0$. We define the adversarial loss of level ε associated with ϕ as:*

$$\phi_\varepsilon(x, y, f) = \sup_{x' \in B_\varepsilon(x)} \phi(yf(x'))$$

[2]Note that after shifting x to x', the point need not be in the support of $\mathbb{P}$ anymore.

We say that ϕ is adversarially calibrated (resp. uniformly calibrated, resp. consistent) at level ε if ϕ_ε is calibrated (resp. uniformly calibrated, resp. consistent) wrt $l_{0/1,\varepsilon}$.

We can make a first observation: the calibration functions for ϕ and ϕ_ε are actually equal. This property might seem counter-intuitive at first sight as the adversarial risk is most of the time strictly larger than its standard counterpart. However, the calibration functions are only pointwise dependent, hence having the same prediction for any element of the ball $B_\varepsilon(x)$ suffices to reach the optimal calibration $\mathcal{C}^\star_\phi(x,\eta)$.

Proposition 2. *Let $\varepsilon > 0$. Let ϕ be a continuous classification margin loss. For all $x \in \mathcal{X}$ and $\eta \in [0,1]$, we have*

$$\mathcal{C}^\star_{\phi_\varepsilon}(x,\eta) = \inf_{\alpha\in\mathbb{R}} \eta\phi(\alpha) + (1-\eta)\phi(-\alpha) = \mathcal{C}^\star_\phi(x,\eta) \quad .$$

The last equality also holds for the adversarial $0/1$ loss.

$\mathcal{H}$-consistency and $\mathcal{H}$-calibration Awasthi et al. [2021a,c], Bao et al. [2020] proposed to study $\mathcal{H}$-calibration and $\mathcal{H}$-consistency in the adversarial setting, i.e. calibration and consistency when minimizing sequences are in $\mathcal{H}$. Similarly to the calibration function, the $\mathcal{H}$-calibration function is defined as follows.

Definition 16 ($\mathcal{H}$-calibration function)**.** *Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X})$. Let L be a loss function. We also define the optimal $\mathcal{H}$-calibration function:*

$$\mathcal{C}^\star_{L,\mathcal{H}}(x,\eta) := \inf_{f\in\mathcal{H}} \mathcal{C}_L(x,\eta,f)$$

We also define what are $\mathcal{H}$-calibrated losses.

Definition 17 ($\mathcal{H}$-calibration)**.** *Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X})$. Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X})$. Let L_1 and L_2 be two loss functions. We say that L_2 is* $\mathcal{H}$-calibrated *with regards to L_1 if for every $\epsilon > 0$, for all $\eta \in [0,1]$, $x \in \mathcal{X}$, there exists $\delta > 0$ for every $f \in \mathcal{H}$:*

$$\mathcal{C}_{L_2}(x,\eta,f) - \mathcal{C}^\star_{L_2,\mathcal{H}}(x,\eta) \leq \delta \implies \mathcal{C}_{L_1}(x,\eta,f) - \mathcal{C}^\star_{L_1,\mathcal{H}}(x,\eta) \leq \epsilon \quad .$$

Furthermore, we say that L_2 is uniformly $\mathcal{H}$-calibrated *with regards to L_1 if for every $\epsilon > 0$, there exists $\delta > 0$, for all $\eta \in [0,1]$, $x \in \mathcal{X}$, for every $f \in \mathcal{H}$:*

$$\mathcal{C}_{L_2}(x,\eta,f) - \mathcal{C}^\star_{L_2,\mathcal{H}}(x,\eta) \leq \delta \implies \mathcal{C}_{L_1}(x,\eta,f) - \mathcal{C}^\star_{L_1,\mathcal{H}}(x,\eta) \leq \epsilon \quad .$$

However, even in the standard classification setting, the link between both notions in this extended setting is not clear at all since a pointwise minimization of the risk cannot be done. To our knowledge, there is only one research paper [Long and Servedio, 2013] that focuses on this notion in standard setting. They do it in the restricted case of realisability, i.e. when the standard optimal risk associated with the $0/1$ loss equals 0. We believe that studying $\mathcal{H}$-consistency and $\mathcal{H}$-calibration in the adversarial setting is a bit anticipated. For these reasons, in Chapter 5) we mainly focus on calibration and consistency on the space of measurable functions $\mathcal{F}(\mathcal{X})$ although some results can be adapted to $\mathcal{H}$-calibration.

3.3 Robustness and Lipchitzness

In this section, we have interest in the deep link that exist between adversarial examples and Lipschitzness. Indeed, a Lipschitz function is a function that do not vary a lot when varying its

input and a classifier is robust if a small perturbation do not change the prediction. Formally, we recall a classifier h is *certifiably robust at level* ε at input x with label y if there exist a property depending on h, x, y and ε that implies that for all x' such that $d(x, x') \leq \varepsilon$, $h(x') = y$. We first recall a property linking Lipschitzness to Robustness. Then, we present the existing methods for building Lipschitz Neural Networks.

3.3.1 Lipschitz Property of Neural Networks

The Lipschitz constant has seen a growing interest in the last few years in the field of deep learning [Béthune et al., 2021, Combettes and Pesquet, 2020, Fazlyab et al., 2019, Virmaux and Scaman, 2018]. Indeed, numerous results have shown that neural networks with a small Lipschitz constant exhibit better generalization [Bartlett et al., 2017], higher robustness to adversarial attacks [Farnia et al., 2019, Szegedy et al., 2014, Tsuzuku et al., 2018], better training stability [Trockman et al., 2021, Xiao et al., 2018], improved Generative Adversarial Networks [Arjovsky et al., 2017], etc. Formally, we define the Lipschitz constant with respect to the ℓ_2 norm of a Lipschitz continuous function f as follows:

$$Lip_2(f) = \sup_{\substack{x,x' \in \mathcal{X} \\ x \neq x'}} \frac{\|f(x) - f(x')\|_2}{\|x - x'\|_2} \ .$$

Intuitively, if a classifier is Lipschitz, one can bound the impact of a given input variation on the output, hence obtaining guarantees on the adversarial robustness. We can formally characterize the robustness of a neural network with respect to its Lipschitz constant with the following proposition:

Proposition 3 (Tsuzuku et al. [2018])**.** *Let $f : \mathcal{X} \to \mathbb{R}^K$ be an L-Lipschitz continuous classifier for the ℓ_2 norm. Let $\varepsilon > 0$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ the label of x. If at point x, the margin $\mathcal{M}_{\mathbf{f}}(x)$ satisfies:*

$$\mathcal{M}_{\mathbf{f}}(x) := \max(0, f_y(x) - \max_{y' \neq y} f_{y'}(x)) > \sqrt{2} L \varepsilon$$

then we have for every τ such that $\|\tau\|_2 \leq \varepsilon$:

$$\operatorname*{argmax}_k f_k(x + \tau) = y$$

From Proposition 3, it is straightforward to compute a robustness certificate for a given point. Consequently, in order to build robust neural networks the margin needs to be large and the Lipschitz constant small to get optimal guarantees on the robustness for neural networks. Beyond adversarial robustness, Lipschitzness is very used in Wasserstein Generative Adversarial Networks. Indeed the discriminator objective writes as a Wasserstein-1 distance in its dual form:

$$W_1(\mathbb{P}, G_\sharp \mathbb{P}_z) = \sup_{f \ \text{1-Lip}} \mathbb{E}_{x \sim \mathbb{P}}[f(x)] - \mathbb{E}_{z \sim \mathbb{P}_z}[f(G(z))]$$

where $\mathbb{P}_z$ denotes the latent space, and G the generator function. It worth noting that Wasserstein GANs highly improved the stability of training for GANs.

Lipschitz Constant of Neural Networks. A neural network is a function f defined succession of linear and non-linear activation functions σ:

$$f(x) = (A_L \sigma (A_{L-1} \dots \sigma (A_1 x + b_1) \dots) + b_L)$$

Algorithm 2: Spectral normalization algorithm

Require: **Matrix W, Nb. Iter.** n
Initialize u and v
$v \leftarrow \mathbf{W}u/\|\mathbf{W}u\|_2$
$u \leftarrow \mathbf{W}^\top v/\|\mathbf{W}^\top v\|_2$
$h \leftarrow 2/\left(\sum_i (\mathbf{W}u \cdot v)_i\right)^2$
} n iterations
return h

Assuming that σ is 1-Lipschitz, we have:

$$\|f(x) - f(y)\|_2 \leq \|A_1\|_2 \dots \|A_L\|_2 \|x - y\|_2$$

with $\|A\|_2$ is the spectral norm of A defined as

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \lambda_{max}(A^T A) \quad .$$

where $\lambda_{max}(A^T A)$ denotes the greatest eigen value of $A^T A$. Note that $\|A\|_2$ is also the greatest singular value of A. Then the Lipschitz constant of f is upperbounded by $\|A_1\|_2 \dots \|A_L\|_2$. Hence to control the Lipschitz constant of a neural network, it is usual to control the spectral norm of each layer. It could be done either in penalizing this upperbound or imposing a spectral norm equals smaller than 1 for each layer.

Lipschitz Regularization of Neural Networks. Based on the insight that Lipschitz Neural Networks are more robust to adversarial attacks, researchers have developed several techniques to regularize and constrain the Lipschitz constant of neural networks by adding a regularization $\Omega(f)$ to the classification objective to encourage a smaller Lipschitz constant. However the computation of the Lipschitz constant of neural networks has been shown to be NP-hard [Virmaux and Scaman, 2018]. Most methods therefore tackle the problem by reducing or constraining the Lipschitz constant at the layer level. For instance, the work of Cisse et al. [2017], Huang et al. [2020a] and Wang et al. [2020] exploit the orthogonality of the weights matrices to build Lipschitz layers. Other approaches [Araujo et al., 2021, Gouk et al., 2018, Jia et al., 2017, Sedghi et al., 2018, Singla et al., 2021b] proposed to estimate or upper-bound the spectral norm of convolutional and dense layers using for instance the power iteration method [Golub et al., 2000]. While these methods have shown interesting results in terms of accuracy, empirical robustness and efficiency, they can not provide provable guarantees since the Lipschitz constant of the trained networks remains unknown or vacuous.

3.3.2 Learning 1-Lipschitz layers

Many research proposed methods to build 1-Lipschitz layers in order to boost adversarial robustness. These approaches provide deterministic guarantees for adversarial robustness. One can either normalize the weight matrices by their largest singular values making the layer 1-Lipschitz, *e.g.* [Anil et al., 2019, Farnia et al., 2019, Miyato et al., 2018, Yoshida and Miyato, 2017] or project the weight matrices on the Stiefel manifold [Li et al., 2019a, Singla and Feizi, 2021, Trockman et al., 2021].

The first natural idea to learn 1-Lipschitz layers is to normalize the matrices in the forward pass of a Neural Networks : $A_i \leftarrow \frac{A_i}{\|A_i\|_2}$. This natural idea was exploited by Miyato et al. [2018].

A key difficulty is the computation of the spectral norm $\|A_i\|_2$. The authors proposed to use the power iteration method to compute the spectral norm (see Algorithm 2). The number of iterations might be prohibitive, hence the authors proposed to use only one step in the training phase to make it faster. This method effectively approximated well the sprectral norm of the last layer. However, this method present some disadvantages. The spectral normalization has for effect crushing all smaller singular values. A consequence is the gradient vanishing that is very present in this structure.

Also, several works [Anil et al., 2019, Huang et al., 2021b, Singla et al., 2021a] proposed methods leveraging the properties of activation functions to constraints the Lipschitz of Neural Networks. These works are usually useful to help improving the performance of linear orthogonal layers.

Learning Orthogonal layers A workaround for the limitations of previously presented methods is to build norm preserving linear layers, i.e. orthogonal layers. We recall a matrix $\Omega \in \mathbb{R}^{d\times d}$ is said to be orthogonal if for every $x \in \mathbb{R}^d$, $\|\Omega x\|_2 = \|x\|_2$. Indeed such layers exactly preserve the norm, hence avoid crushing all singular values and gradient vanishing issues. Recently, there have been a trend in aiming at learning Orthogonal Layers in neural networks. The following approaches consist of projecting the weights matrices onto an orthogonal space in order to preserve gradient norms and enhance adversarial robustness by guaranteeing low Lipschitz constants. While both works have similar objectives, their execution is different .It is a difficult question to conciliate the convolution structure with orthogonality of linear layers. The presented works of Li et al. [2019a], Trockman et al. [2021] and Singla and Feizi [2021] (denoted BCOP, Cayley and SOC respectively) present the advantage of being "compatible" with convolutional structure in layers.

The BCOP layer (Block Convolution Orthogonal Parameterization) uses an iterative algorithm proposed by Björck et al. [1971] to orthogonalize a linear transformation. The BCOP layer relies on the following algorithm to orthonormalize a linear operator M:

$$M \times \left(I + \frac{1}{2}Q + \frac{3}{8}Q^2 + \cdots + (-1)^p \binom{\frac{1}{2}}{p} Q^p + \dots \right).$$

with $Q = I - A^T A$ To build a "convolutional layer" from the BCOP procedure, the authors proposed to work directly on the kernels of the convolutions, proposing block operations to orthogonalize convolutions.

Two other alternatives, the SOC layer (Skew Orthogonal Convolution) and the Cayley layer, used two different parametrization of the Special Orthogonal Group $SO_n(\mathbb{R})$ using skew-symmetric matrices. Indeed, in Riemmanian geometry, the space skew-symmetric matrices is isomorphic to the tangent space of $SO_n(\mathbb{R})$ at any point.

SOC layers uses the expontial mapping of a skew symmetric matrix defined using the following Taylor expansion:

$$\exp A := \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

which defines an orthogonal matrix, indeed $(\exp A)^T \exp A = \exp(A^T)\exp(A) = \exp(-A)\exp(A) = \exp(A - A) = I$. More precisely, the application $A \mapsto \exp A$ defines a surjective mapping of $SO_n(\mathbb{R})$ from the space of skew-symmetric matrices. To approximate the exponential of a matrix, the authors proposed to use a finite number of terms in its Taylor series expansion. To be adapted to convolutions, a skew-symmetric linear transformation $A = M - M^T$ can

be computed in a Deep Learning Framework using the convolution and convolution-transpose operators.

The Cayley method proposed by Trockman et al. [2021] use the Cayley transform to orthogonalize the weights matrices. Given a skew symmetric matrix A, the Cayley transform consists in computing the orthogonal matrix:

$$\text{Cayley}(A) = (I - A)^{-1}(I + A) \quad .$$

Like exponential mapping, the Cayley Tranform defines a surjective mapping of $SO_n(\mathbb{R})$ from the space of skew-symmetric matrices. To craft such operators, the authors proposed to work in the Fourier domain and directly on the kernels to compute the Cayley Transform.

Reshaped Kernel Methods. It has been shown by Cisse et al. [2017] and Tsuzuku et al. [2018] that the spectral norm of a convolution can be upper-bounded by the norm of a reshaped kernel matrix. Consequently, orthogonalizing directly this matrix upper-bound the spectral norm of the convolution by 1. While this method is more computationally efficient than orthogonalizing the whole convolution, it lacks expressivity as the other singular values of the convolution are certainly too constrained.

3.3.3 Residual Networks

During the training phase in neural networks, it may occur some issues as gradient vanishing or gradient expolosion [Hochreiter et al., 2001]. These issues limited the emergence of scalable and very deep neural networks until He et al. [2016] proposed the Residual Network (ResNet) architecture defined as follows.

$$\begin{cases} x_0 & = x \in \mathcal{X} \\ x_{t+1} & = x_t + F_t(x_t) \text{ for } \ t \in \{0, \dots, T\} \end{cases}$$

where $F_t(x_t)$ is typically a two layer neural networks. The ResNet uses residual connection that have the effect of limiting gradient vanishing issues. Combined with batch normalization, the issue of gradient explosion can also be mitigated, hence opening the possibility to very deep and stable architecture.

To theoretically analyse the ResNet architecture, several works [Chen et al., 2018b, E, 2017, Haber et al., 2017, Lu et al., 2018] proposed a "continuous time" interpretation inspired by dynamical systems that can be defined as follows.

Definition 18. *Let $(F_t)_{t\in[0,T]}$ be a family of functions on $\mathbb{R}^d$, we define the continuous time Residual Networks flow associated with F_t as:*

$$\begin{cases} x_0 & = x \in \mathcal{X} \\ \frac{dx_t}{dt} & = F_t(x_t) \textit{ for } \ t \in [0, T] \end{cases}$$

This continuous time interpretation helps as it allows us to consider the stability of the forward propagation through the stability of the associated dynamical system. A dynamical system is said to be *stable* if two trajectories starting from an input and another one remain sufficiently close to each other all along the propagation. This stability property takes all its sense in the context of adversarial classification.

It was argued by Haber et al. [2017] that when F_t does not depend on t or vary slowly with time[3], the stability can be characterized by the eigenvalues of the Jacobian matrix $\nabla_x F_t(x_t)$:

[3]This blurry definition of "vary slowly" makes the property difficult to apply.

the dynamical system is stable if the real part of the eigenvalues of the Jacobian stay negative throughout the propagation. This property however only relies on intuition and this condition might be difficult to verify in practice. In the following, in order to derive stability properties, we study gradient flows and convex potentials, which are sub-classes of Residual networks.

Other works [Huang et al., 2020b, Li et al., 2020b] also proposed to enhance adversarial robustness using dynamical systems interpretations of Residual Networks. Both works argues that using particular discretization scheme would make gradient attacks more difficult to compute due to numerical stability. These works did not provide any provable guarantees for such approaches.

Chapter 4

Game Theory of Adversarial Examples

Contents

In this chapter, we study the existence of Mixed Nash equilibria in the adversarial example game when both the adversary and the classifier can use randomized strategies. First, we motivate in Section 4.1 the necessity for using randomized strategies both with the attacker and the classifier. Then, we extend the work of Pydi and Jog [2021a], by rigorously reformulating the adversarial risk as a linear optimization problem over distributions. In fact, we cast the adversarial risk minimization problem as a Distributionally Robust Optimization (DRO) [Blanchet and Murthy,

2019] problem for a well suited cost function. This formulation naturally leads us, in Section 4.2, to analyze adversarial risk minimization as a zero-sum game. We demonstrate that, in this game, the duality gap always equals 0, meaning that it always admits approximate mixed Nash equilibria.

Afterwards, we aim at designing an efficient algorithm to learn an optimally robust randomized classifier. We focus on learning a finite mixture of classifiers. Taking inspiration from robust optimization Sinha et al. [2017] and subgradient methods Boyd [2003], we derive in Section 4.3 a first oracle algorithm to optimize a finite mixture. Then, following the line of work of [Cuturi, 2013], we introduce an entropic regularization to effectively compute an approximation of the optimal mixture. We validate our findings with experiments on simulated and real datasets, namely CIFAR-10 an CIFAR-100 Krizhevsky and Hinton [2009].

4.1 The Adversarial Attack Problem

4.1.1 A Motivating Example

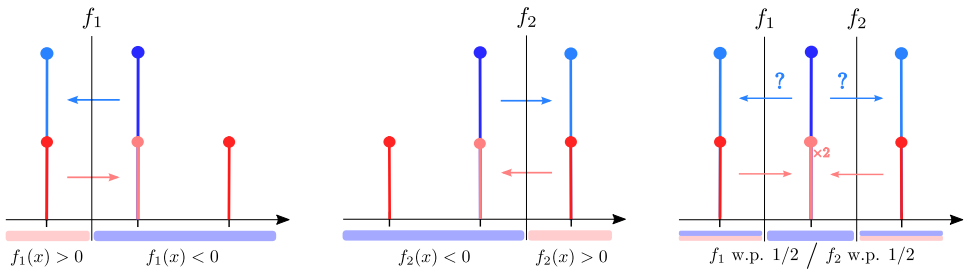


Figure 4.1: Motivating example: blue distribution represents label -1 and the red one, label $+1$. The height of columns represents their mass. The red and blue arrows represent the attack on the given classifier. On left: deterministic classifiers (f_1 on the left, f_2 in the middle) for whose, the blue point can always be attacked. On right: a randomized classifier, where the attacker has a probability $1/2$ of failing, regardless of the attack it selects.

Consider the binary classification task illustrated in Figure 4.1. We assume that all input-output pairs (X, Y) are sampled from a distribution $\mathbb{P}$ defined as follows

$$\mathbb{P}(Y = \pm 1) = 1/2 \quad \text{and} \quad \begin{cases} \mathbb{P}(X = 0 \mid Y = -1) = 1 \\ \mathbb{P}(X = \pm 1 \mid Y = 1) = 1/2 \end{cases}$$

Given access to $\mathbb{P}$, the adversary aims to maximize the expected risk, but can only move each point by at most 1 on the real line. In this context, we study two classifiers: $f_1(x) = -x - 1/2$ and $f_2(x) = x - 1/2$[1]. Both f_1 and f_2 have a standard risk of $1/4$. In the presence of an adversary, the risk (*a.k.a.* the adversarial risk) increases to 1. Here, using a randomized classifier can make the system more robust. Consider f where $f = f_1$ w.p. $1/2$ and f_2 otherwise. The standard risk of f remains $1/4$ but its adversarial risk is $3/4 < 1$. Indeed, when attacking f, any adversary will have to choose between moving points from 0 to 1 or to -1. Either way, the attack only works half of the time; hence an overall adversarial risk of $3/4$. Furthermore, if f knows the strategy the adversary uses, it can always update the probability it gives to f_1 and f_2 to get a better (possibly deterministic) defense. For example, if the adversary chooses to

[1] $(X, Y) \sim \mathbb{P}$ is misclassified by f_i if and only if $f_i(X)Y \leq 0$

always move 0 to 1, the classifier can set $f = f_1$ w.p. 1 to retrieve an adversarial risk of $1/2$ instead of $3/4$.

Now, what happens if the adversary can use randomized strategies, meaning that for each point it can flip a coin before deciding where to move? In this case, the adversary could decide to move points from 0 to 1 w.p. $1/2$ and to -1 otherwise. This strategy is still optimal with an adversarial risk of $3/4$ but now the classifier cannot use its knowledge of the adversary's strategy to lower the risk. We are in a state where neither the adversary nor the classifier can benefit from unilaterally changing its strategy. In the game theory terminology, this state is called a Mixed Nash equilibrium.

4.1.2 General setting

Let us consider a loss function: $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ satisfying the following set of assumptions.

Assumption 1 (Loss function)**.** *1) The loss function L is a non negative Borel measurable function. 2) For all $\theta \in \Theta$, $L(\theta, \cdot)$ is upper-semi continuous. 3) There exists $M > 0$ such that for all $\theta \in \Theta$, $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $0 \leq L(\theta, (x, y)) \leq M$.*

It is usual to assume upper-semi continuity when studying optimization over distributions [Blanchet and Murthy, 2019, Villani, 2003]. Furthermore, considering bounded (and positive) loss functions is also very common in learning theory [Bartlett and Mendelson, 2002] and is not restrictive.

In the adversarial examples framework, the loss of interest is the 0/1 loss, for whose surrogates are misunderstood and is the object of Chapter 5; hence it is essential that a 0/1 loss satisfies Assumption 1. In the binary classification setting (*i.e.* $\mathcal{Y} = \{-1, +1\}$) a possible 0/1 loss writes $L_{0/1}(\theta, (x, y)) = \mathbf{1}_{y f_\theta(x) \leq 0}$. Then, assuming that for all θ, $f_\theta(\cdot)$ is continuous and for all x, $f_\cdot(x)$ is continuous, the 0/1 loss satisfies Assumption 1. In particular, it is the case for neural networks with continuous activation functions.

4.1.3 Measure Theoretic Lemmas

We first recall and prove some important lemmas about theoretic measure.

Lemma 1 (Fubini's theorem)**.** *Let $l : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ satisfying Assumption 1. Then for all $\mu \in \mathcal{M}_+^1(\Theta)$, $\int L(\theta, \cdot) d\mu(\theta)$ is Borel measurable; for $\mathbb{Q} \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$, $\int L(\cdot, (x, y)) d\mathbb{Q}(x, y)$ is Borel measurable. Moreover: $\int L(\theta, (x, y)) d\mu(\theta) d\mathbb{Q}(x, y) = \int L(\theta, (x, y)) d\mathbb{Q}(x, y) d\mu(\theta)$*

Lemma 2. *Let $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ satisfying Assumption 1. Then for all $\mu \in \mathcal{M}_+^1(\Theta)$, $(x, y) \mapsto \int L(\theta, (x, y)) d\mu(\theta)$ is upper semi-continuous and hence Borel measurable.*

Proof. Let $(x_n, y_n)_n$ be a sequence of $\mathcal{X} \times \mathcal{Y}$ converging to $(x, y) \in \mathcal{X} \times \mathcal{Y}$. For all $\theta \in \Theta$, $M - L(\theta, \cdot)$ is non negative and lower semi-continuous. Then by Fatou's Lemma applied:

$$\begin{aligned} \int M - L(\theta, (x, y)) d\mu(\theta) &\leq \int \liminf_{n \to \infty} M - L(\theta, (x_n, y_n)) d\mu(\theta) \\ &\leq \liminf_{n \to \infty} \int M - L(\theta, (x_n, y_n)) d\mu(\theta) \end{aligned}$$

Then we deduce that: $\int M - L(\theta, \cdot) d\mu(\theta)$ is lower semi-continuous and then $\int L(\theta, \cdot) d\mu(\theta)$ is upper-semi continuous. □

Lemma 3. *Let $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ satisfying Assumption 1 Then for all $\mu \in \mathcal{M}_+^1(\Theta)$, $\mathbb{Q} \mapsto \int L(\theta, (x, y)) d\mu(\theta) d\mathbb{Q}(x, y)$ is upper semi-continuous for weak topology of measures.*

Proof. $-\int L(\theta,\cdot)d\mu(\theta)$ is lower semi-continuous from Lemma 2. Then $M-\int L(\theta,\cdot)d\mu(\theta)$ is lower semi-continuous and non negative. Let denote v this function. Let $(v_n)_n$ be a non-decreasing sequence of continuous bounded functions such that $v_n \to v$. Let $(\mathbb{Q}_k)_k$ converging weakly towards $\mathbb{Q}$. Then by monotone convergence:

$$\int v d\mathbb{Q} = \lim_n \int v_n d\mathbb{Q} = \lim_n \lim_k \int v_n d\mathbb{Q}_k \leq \lim_k \inf \int v d\mathbb{Q}_k$$

Then $\mathbb{Q} \mapsto \int v d\mathbb{Q}$ is lower semi-continuous and then $\mathbb{Q} \mapsto \int L(\theta,(x,y))d\mu(\theta)d\mathbb{Q}(x,y)$ is upper semi-continuous for weak topology of measures. □

Lemma 4. *Let $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0,\infty)$ satisfying Assumption 1. Then for all $\mu \in \mathcal{M}_+^1(\Theta)$, $(x,y) \mapsto \sup_{(x',y'),d(x,x')\leq\varepsilon,y=y'} \int L(\theta,(x',y'))d\mu(\theta)$ is universally measurable (i.e. measurable for all Borel probability measures). And hence the adversarial risk is well defined.*

Proof. Let $\phi : (x,y) \mapsto \sup_{(x',y'),d(x,x')\leq\varepsilon,y=y'} \int L(\theta,(x',y'))d\mu(\theta)$. Then for $u \in \bar{\mathbb{R}}$:

$$\{\phi(x,y) > u\} = \mathrm{Proj}_1 \left\{ ((x,y),(x',y')) \mid \int L(\theta,(x',y'))d\mu(\theta) - c_\varepsilon((x,y),(x',y')) > u \right\}$$

By Lemma 3: $((x,y),(x',y')) \mapsto \int L(\theta,(x',y'))d\mu(\theta) - c_\varepsilon((x,y),(x',y'))$ is upper-semicontinuous hence Borel measurable. So its level sets are Borel sets, and by [Bertsekas and Shreve, 2004, Proposition 7.39], the projection of a Borel set is analytic. And then $\{\phi(x,y) > u\}$ universally measurable thanks to [Bertsekas and Shreve, 2004, Corollary 7.42.1]. We deduce that ϕ is universally measurable. □

4.1.4 Adversarial Risk Minimization

The standard risk for a single classifier θ associated with the loss L satisfying Assumption 1 writes: $\mathcal{R}(\theta) := \mathbb{E}_{(x,y)\sim\mathbb{P}}[L(\theta,(x,y))]$. Similarly, the adversarial risk of θ at level ε associated with the loss L is defined as[2]

$$\mathcal{R}_\varepsilon(\theta) := \mathbb{E}_{(x,y)\sim\mathbb{P}} \left[\sup_{x'\in\mathcal{X},\ d(x,x')\leq\varepsilon} L(\theta,(x',y)) \right].$$

It is clear that $\mathcal{R}_0(\theta) = \mathcal{R}(\theta)$ for all θ. We can generalize these notions with distributions of classifiers. In other terms the classifier is then randomized according to some distribution $\mu \in \mathcal{M}_+^1(\Theta)$. A classifier is randomized if for a given input, the output of the classifier is a probability distribution. The standard risk of a randomized classifier μ writes $\mathcal{R}(\mu) = \mathbb{E}_{\theta\sim\mu}[\mathcal{R}(\theta)]$. Similarly, the adversarial risk of the randomized classifier μ at level ε is[3]

$$\mathcal{R}_\varepsilon(\mu) := \mathbb{E}_{(x,y)\sim\mathbb{P}} \left[\sup_{x'\in\mathcal{X},\ d(x,x')\leq\varepsilon} \mathbb{E}_{\theta\sim\mu}[L(\theta,(x',y))] \right].$$

For instance, for the 0/1 loss, the inner maximization problem, consists in maximizing the probability of misclassification for a given couple (x,y). Note that $\mathcal{R}(\delta_\theta) = \mathcal{R}(\theta)$ and $\mathcal{R}_\varepsilon(\delta_\theta) =$

[2]For the well-posedness, see Lemma 4.

[3]This risk is also well posed (see Lemma 4).

$\mathcal{R}_\varepsilon(\theta)$. In the remainder of this section, we study the adversarial risk minimization problems with randomized and deterministic classifiers and denote

$$\mathcal{V}_\varepsilon^{rand} := \inf_{\mu \in \mathcal{M}_+^1(\Theta)} \mathcal{R}_\varepsilon(\mu),\ \mathcal{V}_\varepsilon^{det} := \inf_{\theta \in \Theta} \mathcal{R}_\varepsilon(\theta) \tag{4.1}$$

Note that we can show that the standard risk infima are equal : $\mathcal{V}_0^{rand} = \mathcal{V}_0^{det}$.

Proposition 4. *Let $\mathbb{P}$ be a Borel probability distribution on $\mathcal{X} \times \mathcal{Y}$, and l a loss satisfying Assumption 1, then:*

$$\inf_{\mu \in \mathcal{M}_+^1(\Theta)} \mathcal{R}(\mu) = \inf_{\theta \in \Theta} \mathcal{R}(\theta)$$

Proof. It is clear that: $\inf_{\mu \in \mathcal{M}_+^1(\Theta)} \mathcal{R}(\mu) \leq \inf_{\theta \in \Theta} \mathcal{R}(\theta)$. Now, let $\mu \in \mathcal{M}_+^1(\Theta)$, then:

$$\begin{aligned}\mathcal{R}(\mu) = \mathbb{E}_{\theta \sim \mu}(\mathcal{R}(\theta)) &\geq \operatorname*{essinf}_{\mu} \mathbb{E}_{\theta \sim \mu}\left(\mathcal{R}(\theta)\right) \\ &\geq \inf_{\theta \in \Theta} \mathcal{R}(\theta).\end{aligned}$$

where essinf denotes the essential infimum. □

Remark 2. *No randomization is needed for minimizing the standard risk. Denoting $\mathcal{V}$ this common value, we also have the following inequalities for any $\varepsilon > 0$, $\mathcal{V} \leq \mathcal{V}_\varepsilon^{rand} \leq \mathcal{V}_\varepsilon^{det}$.*

4.1.5 Distributional Formulation of the Adversarial Risk

To account for the possible randomness of the adversary, we rewrite the adversarial attack problem as a convex optimization problem over distributions. Let us first introduce the set of adversarial distributions.

Definition 19 (Set of adversarial distributions)**.** *Let $\mathbb{P}$ be a Borel probability distribution on $\mathcal{X} \times \mathcal{Y}$ and $\varepsilon > 0$. We define the set of adversarial distributions as*

$$\begin{aligned}\mathcal{A}_\varepsilon(\mathbb{P}) := \big\{\mathbb{Q} \in \mathcal{M}_1^+(\mathcal{X} \times \mathcal{Y}) \mid \exists \gamma \in \mathcal{M}_1^+\left((\mathcal{X} \times \mathcal{Y})^2\right), \\ d(x, x') \leq \varepsilon,\ y = y'\ \ \gamma\text{-a.s.},\ \Pi_{1\sharp}\gamma = \mathbb{P},\ \Pi_{2\sharp}\gamma = \mathbb{Q}\big\}\end{aligned}$$

where Π_i denotes the projection on the i-th component, and $g_\sharp$ the push-forward measure by a measurable function g.

An attacker that can move the initial distribution $\mathbb{P}$ anywhere in $\mathcal{A}_\varepsilon(\mathbb{P})$ is not applying a point-wise deterministic perturbation as considered in the standard adversarial risk. In other words, for a point $(x, y) \sim \mathbb{P}$, the attacker could choose a distribution $q(\cdot \mid (x, y))$ whose support is included in $\{(x', y') \mid d(x, x') \leq \epsilon,\ y = y'\}$ from which he will sample the adversarial attack. In this sense, we say the attacker is allowed to be randomized.

Link with DRO. We immediately remark that $\mathcal{A}_\varepsilon(\mathbb{P})$ correspond in the Wasserstein-∞ set associated with the cost

$$d'((x, y), (x', y')) \mapsto \begin{cases} d(x, x') & \text{if } y = y' \\ +\infty & \text{otherwise.}\end{cases}$$

We also remark, such a set can be defined from usual (not ∞) Wasserstein uncertainty sets: for an arbitrary $\varepsilon > 0$, we define the cost c_ε as follows

$$c_\varepsilon((x, y), (x', y')) := \begin{cases} 0 & \text{if } d(x, x') \leq \varepsilon \text{ and } y = y' \\ +\infty & \text{otherwise.}\end{cases}$$

This cost is lower semi-continuous and penalizes to infinity perturbations that change the label or move the input by a distance greater than ε. As Proposition 5 shows, the Wasserstein ball associated with c_ε is equal to $\mathcal{A}_\varepsilon(\mathbb{P})$.

Proposition 5. *Let $\mathbb{P}$ be a Borel probability distribution on $\mathcal{X} \times \mathcal{Y}$ and $\varepsilon > 0$ and $\eta \geq 0$, then $\mathcal{B}_{c_\varepsilon}(\mathbb{P}, \eta) = \mathcal{A}_\varepsilon(\mathbb{P})$. Moreover, $\mathcal{A}_\varepsilon(\mathbb{P})$ is convex and compact for the weak topology of $\mathcal{M}_1^+(\mathcal{X} \times \mathcal{Y})$.*

Proof. Let $\eta > 0$. Let $\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})$. There exists $\gamma \in \mathcal{M}_1^+\left((\mathcal{X} \times \mathcal{Y})^2\right)$ such that, $d(x, x') \leq \varepsilon$, $y = y'$ γ-almost surely, and $\Pi_{1\sharp}\gamma = \mathbb{P}$, and $\Pi_{2\sharp}\gamma = \mathbb{Q}$. Then $\int c_\varepsilon d\gamma = 0 \leq \eta$. Then, we deduce that $W_{c_\varepsilon}(\mathbb{P}, \mathbb{Q}) \leq \eta$, and $\mathbb{Q} \in \mathcal{B}_{c_\varepsilon}(\mathbb{P}, \eta)$. Reciprocally, let $\mathbb{Q} \in \mathcal{B}_{c_\varepsilon}(\mathbb{P}, \eta)$. Then, since the infimum is attained in the Wasserstein definition, there exists $\gamma \in \mathcal{M}_1^+\left((\mathcal{X} \times \mathcal{Y})^2\right)$ such that $\int c_\varepsilon d\gamma \leq \eta$. Since $c_\varepsilon((x, x'), (y, y')) = +\infty$ when $d(x, x') > \varepsilon$ and $y \neq y'$, we deduce that, $d(x, x') \leq \varepsilon$ and $y = y'$, γ-almost surely. Then $\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})$. We have then shown that: $\mathcal{A}_\varepsilon(\mathbb{P}) = \mathcal{B}_{c_\varepsilon}(\mathbb{P}, \eta)$.

The convexity of $\mathcal{A}_\varepsilon(\mathbb{P})$ is then immediate from the relation with the Wasserstein uncertainty set.

Let us show first that $\mathcal{A}_\varepsilon(\mathbb{P})$ is relatively compact for weak topology. To do so we will show that $\mathcal{A}_\varepsilon(\mathbb{P})$ is tight and apply Prokhorov's theorem. Let $\delta > 0$, $(\mathcal{X} \times \mathcal{Y}, d \oplus d')$ being a Polish space, $\{\mathbb{P}\}$ is tight then there exists K_δ compact such that $\mathbb{P}(K_\delta) \geq 1 - \delta$. Let $\tilde{K}_\delta := \{(x', y') \mid \exists (x, y) \in K_\delta,\ d(x', x) \leq \varepsilon,\ y = y'\}$. Recalling that $(\mathcal{X}, d)$ is proper (i.e. the closed balls are compact), so $\tilde{K}_\delta$ is compact. Moreover for $\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})$, $\mathbb{Q}(\tilde{K}_\delta) \geq \mathbb{P}(K_\delta) \geq 1 - \delta$. And then, Prokhorov's theorem holds, and $\mathcal{A}_\varepsilon(\mathbb{P})$ is relatively compact for weak topology.

Let us now prove that $\mathcal{A}_\varepsilon(\mathbb{P})$ is closed to conclude. Let $(\mathbb{Q}_n)_n$ be a sequence of $\mathcal{A}_\varepsilon(\mathbb{P})$ converging towards some $\mathbb{Q}$ for weak topology. For each n, there exists $\gamma_n \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$ such that $d(x, x') \leq \varepsilon$ and $y = y'$ γ_n-almost surely and $\Pi_{1\sharp}\gamma_n = \mathbb{P}$, $\Pi_{2\sharp}\gamma_n = \mathbb{Q}_n$. $\{\mathbb{Q}_n, n \geq 0\}$ is relatively compact, then tight, then $\bigcup_n \Gamma_{\mathbb{P}, \mathbb{Q}_n}$ is tight, then relatively compact by Prokhorov's theorem. $(\gamma_n)_n \in \bigcup_n \Gamma_{\mathbb{P}, \mathbb{Q}_n}$, then up to an extraction, $\gamma_n \to \gamma$. Then $d(x, x') \leq \varepsilon$ and $y = y'$ γ-almost surely, and by continuity, $\Pi_{1\sharp}\gamma = \mathbb{P}$ and by continuity, $\Pi_{2\sharp}\gamma = \mathbb{Q}$. And hence $\mathcal{A}_\varepsilon(\mathbb{P})$ is closed.

Finally $\mathcal{A}_\varepsilon(\mathbb{P})$ is a convex compact set for the weak topology. □

Thanks to this result, we can reformulate the adversarial risk as the value of a convex problem over $\mathcal{A}_\varepsilon(\mathbb{P})$.

Proposition 6. *Let $\mathbb{P}$ be a Borel probability distribution on $\mathcal{X} \times \mathcal{Y}$ and μ a Borel probability distribution on Θ. Let $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ satisfying Assumption 1. Let $\varepsilon > 0$. Then:*

$$\mathcal{R}_\varepsilon(\mu) = \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{(x', y') \sim \mathbb{Q}, \theta \sim \mu} \left[L(\theta, (x', y')) \right]. \tag{4.2}$$

The supremum is attained. Moreover $\mathbb{Q}^* \in \mathcal{A}_\varepsilon(\mathbb{P})$ is an optimum of Problem (4.2) if and only if there exists $\gamma^* \in \mathcal{M}_1^+\left((\mathcal{X} \times \mathcal{Y})^2\right)$ such that: $\Pi_{1\sharp}\gamma^* = \mathbb{P}$, $\Pi_{2\sharp}\gamma^* = \mathbb{Q}^*$, $d(x, x') \leq \varepsilon$, $y = y'$ and $L(x', y') = \sup_{u \in \mathcal{X}, d(x, u) \leq \varepsilon} L(u, y)$ γ^*-almost surely.

Proof. Let $\mu \in \mathcal{M}_+^1(\Theta)$. Let $\tilde{f} : ((x, y), (x', y')) \mapsto \mathbb{E}_{\theta \sim \mu} \left[L(\theta, (x, y)) \right] - c_\varepsilon((x, y), (x', y'))$. $\tilde{f}$ is upper-semi continuous, hence upper semi-analytic. Then, by upper semi continuity of $\mathbb{E}_{\theta \sim \mu} \left[L(\theta, \cdot) \right]$ on the compact $\{(x', y') \mid\ d(x, x') \leq \varepsilon, y = y'\}$ and [Bertsekas and Shreve, 2004, Proposition 7.50], there exists a universally measurable mapping T such that $\mathbb{E}_{\theta \sim \mu} \left[L(\theta, T(x, y)) \right] = \sup_{(x', y'),\ d(x, x') \leq \varepsilon, y = y'} \mathbb{E}_{\theta \sim \mu} \left[L(\theta, (x, y)) \right]$. Let $\mathbb{Q} = T_\sharp \mathbb{P}$, then $\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})$. And then

$$\mathbb{E}_{(x, y) \sim \mathbb{P}} \left[\sup_{(x', y'),\ d(x, x') \leq \varepsilon, y = y'} \mathbb{E}_{\theta \sim \mu} \left[L(\theta, (x', y')) \right] \right] \leq \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{(x, y) \sim \mathbb{Q}} \left[\mathbb{E}_{\theta \sim \mu} \left[L(\theta, (x, y)) \right] \right]$$

.

Reciprocally, let $\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})$. There exists $\gamma \in \mathcal{M}_+^1((\mathcal{X} \times \mathcal{Y})^2)$, such that $d(x, x') \leq \varepsilon$ and $y = y'$ γ-almost surely, and, $\Pi_{1\sharp}\gamma = \mathbb{P}$ and $\Pi_{2\sharp}\gamma = \mathbb{Q}$. Then: $\mathbb{E}_{\theta\sim\mu}[L(\theta, (x', y'))] \leq \sup_{(u,v),\ d(x,u)\leq\varepsilon, y=v} \mathbb{E}_{\theta\sim\mu}[L(\theta, (u, v))]$ γ-almost surely. Then, we deduce that:

$$\begin{aligned}
\mathbb{E}_{(x',y')\sim\mathbb{Q}}\left[\mathbb{E}_{\theta\sim\mu}[L(\theta, (x', y'))]\right] &= \mathbb{E}_{(x,y,x',y')\sim\gamma}\left[\mathbb{E}_{\theta\sim\mu}[L(\theta, (x', y'))]\right] \\
&\leq \mathbb{E}_{(x,y,x',y')\sim\gamma}\left[\sup_{(u,v),\ d(x,u)\leq\varepsilon, y=v} \mathbb{E}_{\theta\sim\mu}[L(\theta, (u, v))]\right] \\
&\leq \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{(u,v),\ d(x,u)\leq\varepsilon, y=v} \mathbb{E}_{\theta\sim\mu}[L(\theta, (u, v))]\right]
\end{aligned}$$

Then we deduce the expected result:

$$\mathcal{R}_\varepsilon(\mu) = \sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{(x,y)\sim\mathbb{Q}}\left[\mathbb{E}_{\theta\sim\mu}[L(\theta, (x, y))]\right]$$

Let us show that the optimum is attained. $\mathbb{Q} \mapsto \mathbb{E}_{(x,y)\sim\mathbb{Q}}\left[\mathbb{E}_{\theta\sim\mu}[L(\theta, (x, y))]\right]$ is upper semi continuous by Lemma 3 for the weak topology of measures, and $\mathcal{A}_\varepsilon(\mathbb{P})$ is compact by Proposition 5, then by [Bertsekas and Shreve, 2004, Proposition 7.32], the supremum is attained for a certain $\mathbb{Q}^* \in \mathcal{A}_\varepsilon(\mathbb{P})$.

□

The adversarial attack problem is a DRO problem for the cost c_ε. Proposition 6 means that, against a fixed classifier μ, the randomized attacker that can move the distribution in $\mathcal{A}_\varepsilon(\mathbb{P})$ has exactly the same power as an attacker that moves every single point x in the ball of radius ε. By Proposition 6, we also deduce that the adversarial risk can be casted as a linear optimization problem over distributions.

Remark 3. *In a recent work, [Pydi and Jog, 2021a] proposed a similar adversary using Markov kernels but left as an open question the link with the classical adversarial risk, due to measurability issues. Proposition 6 solves these issues. The result is similar to [Blanchet and Murthy, 2019]. Although we believe its proof might be extended for infinite valued costs, [Blanchet and Murthy, 2019] did not treat that case. We provide an alternative proof in this special case.*

4.2 Nash Equilibria in the Adversarial Game

4.2.1 Adversarial Attacks as a Zero-Sum Game

Thanks to Proposition 4.1, the adversarial risk minimization problem can be seen as a two-player zero-sum game that writes as follows,

$$\inf_{\mu\in\mathcal{M}_+^1(\Theta)} \sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{(x,y)\sim\mathbb{Q},\theta\sim\mu}[L(\theta, (x, y))]. \tag{4.3}$$

In this game the classifier objective is to find the best distribution $\mu \in \mathcal{M}_1^+(\Theta)$ while the adversary is manipulating the data distribution. For the classifier, solving the infimum problem in Equation (4.3) simply amounts to solving the adversarial risk minimization problem – Problem (4.1), whether the classifier is randomized or not. Then, given a randomized classifier $\mu \in \mathcal{M}_1^+(\Theta)$, the goal of the attacker is to find a new data-set distribution $\mathbb{Q}$ in the set of adversarial distributions $\mathcal{A}_\varepsilon(\mathbb{P})$ that maximizes the risk of μ. More formally, the adversary looks for

$$\mathbb{Q} \in \operatorname*{argmax}_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{(x,y)\sim\mathbb{Q},\theta\sim\mu}[L(\theta, (x, y))].$$

In the game theoretic terminology, $\mathbb{Q}$ is also called the best response of the attacker to the classifier μ.

Remark 4. *Note that for a given classifier μ there always exists a "deterministic" best response, i.e. every single point (x, y) is mapped to another single point $T(x, y)$. Let $T : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ be defined such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $\mathbb{E}_{\theta\sim\mu}[L(T(x, y))] = \sup_{x',\ d(x,x')\leq\varepsilon} \mathbb{E}_{\theta\sim\mu}[L(x', y)]$. Thanks to [Bertsekas and Shreve, 2004, Proposition 7.50], T is $\mathbb{P}$-measurable. Moreover, we get that $\mathbb{Q} = (T, id)_\sharp\mathbb{P}$ belongs to the best response to μ. Therefore, T is the optimal "deterministic" attack against the classifier μ.*

4.2.2 Dual Formulation of the Game

Every zero sum game has a dual formulation that allows a deeper understanding of the framework. Here, from Proposition 6, we can define the dual problem of adversarial risk minimization for randomized classifiers. This dual problem also characterizes a two-player zero-sum game that writes as follows,

$$\sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \inf_{\mu\in\mathcal{M}_+^1(\Theta)} \mathbb{E}_{(x,y)\sim\mathbb{Q},\theta\sim\mu}[L(\theta, (x, y))]. \tag{4.4}$$

In this dual game problem, the adversary plays first and seeks an adversarial distribution that has the highest possible risk when faced with an arbitrary classifier. This means that it has to select an adversarial perturbation for every input x, without seeing the classifier first. In this case, as pointed out by the motivating example in Section 4.1.1, the attack can (and should) be randomized to ensure maximal harm against several classifiers. Then, given an adversarial distribution, the classifier objective is to find the best possible classifier on this distribution. Let us denote $\mathcal{D}^\varepsilon$ the value of the dual problem. Since the weak duality is always satisfied, we get

$$\mathcal{D}_\varepsilon \leq \mathcal{V}_\varepsilon^{rand} \leq \mathcal{V}_\varepsilon^{det}. \tag{4.5}$$

Inequalities in Equation (4.5) mean that the lowest risk the classifier can get (regardless of the game we look at) is $\mathcal{D}^\varepsilon$. In particular, this means that the primal version of the game, *i.e.* the adversarial risk minimization problem, will always have a value greater or equal to $\mathcal{D}^\varepsilon$. As we discussed in Section 4.1.1, this lower bound may not be attained by a deterministic classifier. As we will demonstrate in the next section, optimizing over randomized classifiers allows to approach $\mathcal{D}^\varepsilon$ arbitrary closely.

Note that, we can always define the dual problem when the classifier is deterministic,

$$\sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \inf_{\theta\in\Theta} \mathbb{E}_{(x,y)\sim\mathbb{Q}}[L(\theta, (x, y))].$$

We can deduce an immediate corollary from Proposition 4 that the dual problems for deterministic and randomized classifiers have the same value.

Corollary 1. *Under Assumption 1, the dual for randomized and deterministic classifiers are equal.*

4.2.3 Nash Equilibria for Randomized Strategies

In the adversarial examples game, a Nash equilibrium is a couple $(\mu^*, \mathbb{Q}^*) \in \mathcal{M}_+^1(\Theta) \times \mathcal{A}_\varepsilon(\mathbb{P})$ where both the classifier and the attacker have no incentive to deviate unilaterally from their strategies μ^* and $\mathbb{Q}^*$. More formally, $(\mu^*, \mathbb{Q}^*)$ is a Nash equilibrium of the adversarial examples game if $(\mu^*, \mathbb{Q}^*)$ is a saddle point of the objective function

$$(\mu, \mathbb{Q}) \mapsto \mathbb{E}_{(x,y)\sim\mathbb{Q},\theta\sim\mu}[L(\theta, (x, y))].$$

Alternatively, we can say that $(\mu^*, \mathbb{Q}^*)$ is a Nash equilibrium if and only if μ^* solves the adversarial risk minimization problem – Problem (4.1), $\mathbb{Q}^*$ the dual problem – Problem (4.6), and $\mathcal{D}^\varepsilon = \mathcal{V}^\varepsilon_{rand}$. In our problem, $\mathbb{Q}^*$ always exists but it might not be the case for μ^*. Then for any $\delta > 0$, we say that $(\mu_\delta, \mathbb{Q}^*)$ is a δ-approximate Nash equilibrium if $\mathbb{Q}^*$ solves the dual problem and μ_δ satisfies $\mathcal{D}^\varepsilon \geq \mathcal{R}_\varepsilon(\mu_\delta) - \delta$.

We now state our main result: the existence of approximate Nash equilibria in the adversarial examples game when both the classifier and the adversary can use randomized strategies. More precisely, we demonstrate that the duality gap between the adversary and the classifier problems is zero, which gives as a corollary the existence of Nash equilibria.

Theorem 9. *Let $\mathbb{P} \in \mathcal{M}^1_+(\mathcal{X} \times \mathcal{Y})$. Let $\varepsilon > 0$. Let $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ satisfying Assumption 1. Then strong duality always holds in the randomized setting:*

$$\begin{aligned} &\inf_{\mu \in \mathcal{M}^+_1(\Theta)} \max_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{\theta \sim \mu, (x,y) \sim \mathbb{Q}} \left[L(\theta, (x, y)) \right] \\ = &\max_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \inf_{\mu \in \mathcal{M}^+_1(\Theta)} \mathbb{E}_{\theta \sim \mu, (x,y) \sim \mathbb{Q}} \left[L(\theta, (x, y)) \right] \end{aligned} \tag{4.6}$$

The supremum is always attained. If Θ is a compact set, and for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $L(\cdot, (x, y))$ is lower semi-continuous, the infimum is also attained.

Proof. $\mathcal{A}_\varepsilon(\mathbb{P})$, endowed with the weak topology of measures, is a Hausdorff compact convex space, thanks to Proposition 5. Moreover, $\mathcal{M}^1_+(\Theta)$ is clearly convex and $(\mathbb{Q}, \mu) \mapsto \int l d\mu d\mathbb{Q}$ is bilinear, hence concave-convex. Moreover thanks to Lemma 3, for all μ, $\mathbb{Q} \mapsto \int l d\mu d\mathbb{Q}$ is upper semi-continuous. Then Fan's theorem applies and strong duality holds. □

Corollary 2. *Under Assumption 1, for any $\delta > 0$, there exists a δ-approximate Nash-Equibilrium $(\mu_\delta, \mathbb{Q}^*)$. Moreover, if the infimum is attained, there exists a Nash equilibrium $(\mu^*, \mathbb{Q}^*)$ to the adversarial examples game.*

Bose et al. [2021] mentioned a particular form of Theorem 9 for convex cases. It is still a direct corollary of Fan's theorem. This theorem can be stated as follows:

Theorem 10. *Let $\mathbb{P} \in \mathcal{M}^1_+(\mathcal{X} \times \mathcal{Y})$, $\varepsilon > 0$ and Θ a convex set. Let L be a loss satisfying Assumption 1, and also, $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $L(\cdot, (x, y))$ is a convex function, then we have the following:*

$$\inf_{\theta \in \Theta} \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{\mathbb{Q}} \left[L(\theta, (x, y)) \right] = \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \inf_{\theta \in \Theta} \mathbb{E}_{\mathbb{Q}} \left[L(\theta, (x, y)) \right]$$

The supremum is always attained. If Θ is a compact set then, the infimum is also attained.

Theorem 9 shows that $\mathcal{D}^\varepsilon = \mathcal{V}^\varepsilon_{rand}$. From a game theoretic perspective, this means that the minimal adversarial risk for a randomized classifier against any attack (primal problem) is the same as the maximal risk an adversary can get by using an attack strategy that is oblivious to the classifier it faces (dual problem). This suggests that playing randomized strategies for the classifier could substantially improve robustness to adversarial examples. In the next section, we will design an algorithm that efficiently learn a randomized classifier and show improved adversarial robustness over classical deterministic defenses.

Remark 5. *Theorem 9 remains true if one replaces $\mathcal{A}_\varepsilon(\mathbb{P})$ with any other Wasserstein compact uncertainty sets (see [Yue et al., 2020] for conditions of compactness).*

4.3 Finding the Optimal Classifiers

4.3.1 An Entropic Regularization

Let $\{(x_i, y_i)\}_{i=1}^N$ samples independently drawn from $\mathbb{P}$ and denote $\widehat{\mathbb{P}} := \frac{1}{N}\sum_{i=1}^N \delta_{(x_i,y_i)}$ the associated empirical distribution. One can show the adversarial empirical risk minimization can be casted as:

$$\widehat{\mathcal{R}}_{adv}^{\varepsilon,*} := \inf_{\mu\in\mathcal{M}_1^+(\Theta)} \sum_{i=1}^N \sup_{\mathbb{Q}_i\in\Gamma_{i,\varepsilon}} \mathbb{E}_{(x,y)\sim\mathbb{Q}_i,\theta\sim\mu}\left[L(\theta,(x,y))\right]$$

where $\Gamma_{i,\varepsilon}$ is defined as :

$$\Gamma_{i,\varepsilon} := \Big\{\mathbb{Q}_i \mid \int d\mathbb{Q}_i = \frac{1}{N},\ \int c_\varepsilon((x_i,y_i),\cdot)d\mathbb{Q}_i = 0\Big\}.$$

Proposition 7. *Let $\hat{\mathbb{P}} := \frac{1}{N}\sum_{i=1}^N \delta_{(x_i,y_i)}$. Let l be a loss satisfying Assumption 1. Then we have:*

$$\frac{1}{N}\sum_{i=1}^N \sup_{x,\ d(x,x_i)\le\varepsilon} \mathbb{E}_{\theta\sim\mu}\left[l(\theta,(x,y))\right] = \sum_{i=1}^N \sup_{\mathbb{Q}_i\in\Gamma_{i,\varepsilon}} \mathbb{E}_{(x,y)\sim\mathbb{Q}_i,\theta\sim\mu}\left[l(\theta,(x,y))\right]$$

where $\Gamma_{i,\varepsilon}$ is defined as :

$$\Gamma_{i,\varepsilon} := \Big\{\mathbb{Q}_i \mid \int d\mathbb{Q}_i = \frac{1}{N},\ \int c_\varepsilon((x_i,y_i),\cdot)d\mathbb{Q}_i = 0\Big\}.$$

Proof. This proposition is a direct application of Proposition 6 for diracs $\delta_{(x_i,y_i)}$. □

In the following, we regularize the above objective by adding an entropic term to each inner supremum problem. Let $\boldsymbol{\alpha} := (\alpha_i)_{i=1}^N \in \mathbb{R}_+^N$ such that for all $i \in \{1,\dots,N\}$, and let us consider the following optimization problem:

$$\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,*} := \inf_{\mu\in\mathcal{M}_1^+(\Theta)} \sum_{i=1}^N \sup_{\mathbb{Q}_i\in\Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}\left[L(\theta,(x,y))\right] - \alpha_i \mathrm{KL}\left(\mathbb{Q}_i \Big|\Big| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)$$

where $\mathbb{U}_{(x,y)}$ is an arbitrary distribution of support equal to:

$$S_{(x,y)}^{(\varepsilon)} := \Big\{(x',y') :\ \text{s.t. } c_\varepsilon((x,y),(x',y')) = 0\Big\},$$

and for all $\mathbb{Q},\mathbb{U} \in \mathcal{M}_+(\mathcal{X}\times\mathcal{Y})$,

$$\mathrm{KL}(\mathbb{Q}||\mathbb{U}) := \begin{cases} \int \log(\frac{d\mathbb{Q}}{d\mathbb{U}})d\mathbb{Q} + |\mathbb{U}| - |\mathbb{Q}| & \text{if } \mathbb{Q}\ll\mathbb{U} \\ +\infty & \text{otherwise.}\end{cases}$$

Note that when $\boldsymbol{\alpha} = 0$, we recover the problem of interest $\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,*} = \widehat{\mathcal{R}}_{adv}^{\varepsilon,*}$. Moreover, we show the regularized supremum tends to the standard supremum when $\boldsymbol{\alpha} \to 0$.

Proposition 8. *For $\mu \in \mathcal{M}_1^+(\Theta)$, one has*

$$\lim_{\alpha_i\to 0} \sup_{\mathbb{Q}_i\in\Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}\left[L(\theta,(x,y))\right] - \alpha_i KL\left(\mathbb{Q}\Big|\Big|\frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)$$
$$= \sup_{\mathbb{Q}_i\in\Gamma_{i,\varepsilon}} \mathbb{E}_{(x,y)\sim\mathbb{Q}_i,\theta\sim\mu}\left[L(\theta,(x,y))\right].$$

Proof. Let us first show that for $\alpha \geq 0$, $\sup_{\mathbb{Q}_i \in \Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}\left[L(\theta,(x,y))\right] - \alpha \mathrm{KL}\left(\mathbb{Q}_i \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)$ admits a solution. Let $\alpha \geq 0$, $(\mathbb{Q}^n_{\alpha,i})_{n\geq 0}$ a sequence such that

$$\mathbb{E}_{\mathbb{Q}^n_{\alpha,i},\mu}\left[L(\theta,(x,y))\right] - \alpha \mathrm{KL}\left(\mathbb{Q}^n_{\alpha,i} \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right) \underset{n}{\to} \sup_{\mathbb{Q}_i \in \Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}\left[L(\theta,(x,y))\right] - \alpha \mathrm{KL}\left(\mathbb{Q}_i \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right).$$

As $\Gamma_{i,\varepsilon}$ is tight ($(\mathcal{X},d)$ is a proper metric space therefore all the closed ball are compact) and by Prokhorov's theorem, we can extract a subsequence which converges toward $\mathbb{Q}^*_{\alpha,i}$. Moreover, L is upper semi-continuous (u.s.c), thus $\mathbb{Q} \to \mathbb{E}_{\mathbb{Q},\mu}\left[L(\theta,(x,y))\right]$ is also u.s.c.[4] Moreover $\mathbb{Q} \to -\alpha \mathrm{KL}\left(\mathbb{Q} \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)$ is also u.s.c. [5], therefore, by considering the limit superior as n goes to infinity we obtain that

$$\begin{aligned}
&\limsup_{n\to+\infty} \mathbb{E}_{\mathbb{Q}^n_{\alpha,i},\mu}\left[L(\theta,(x,y))\right] - \alpha \mathrm{KL}\left(\mathbb{Q}^n_{\alpha,i} \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right) \\
&= \sup_{\mathbb{Q}_i \in \Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}\left[L(\theta,(x,y))\right] - \alpha \mathrm{KL}\left(\mathbb{Q}_i \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right) \\
&\leq \mathbb{E}_{\mathbb{Q}^*_{\alpha,i},\mu}\left[L(\theta,(x,y))\right] - \alpha \mathrm{KL}\left(\mathbb{Q}^*_{\alpha,i} \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)
\end{aligned}$$

from which we deduce that $\mathbb{Q}^*_{\alpha,i}$ is optimal.

Let us now show the result. We consider a positive sequence of $(\alpha_i^{(\ell)})_{\ell\geq 0}$ such that $\alpha_i^{(\ell)} \to 0$. Let us denote $\mathbb{Q}^*_{\alpha_i^{(\ell)},i}$ and $\mathbb{Q}_i^*$ the solutions of $\max_{\mathbb{Q}_i \in \Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}\left[L(\theta,(x,y))\right] - \alpha_i^{(\ell)} \mathrm{KL}\left(\mathbb{Q}_i \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)$ and $\max_{\mathbb{Q}_i \in \Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}\left[L(\theta,(x,y))\right]$ respectively. Since $\Gamma_{i,\varepsilon}$ is tight, $(\mathbb{Q}^*_{\alpha_i^{(\ell)},i})_{\ell\geq 0}$ is also tight and we can extract by Prokhorov's theorem a subsequence which converges towards $\mathbb{Q}^*$. Moreover we have

$$\mathbb{E}_{\mathbb{Q}_i^*,\mu}\left[L(\theta,(x,y))\right] - \alpha_i^{(\ell)} \mathrm{KL}\left(\mathbb{Q}_i^* \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right) \leq \mathbb{E}_{\mathbb{Q}^*_{\alpha_i^{(\ell)},i},\mu}\left[L(\theta,(x,y))\right] - \alpha_i^{(\ell)} \mathrm{KL}\left(\mathbb{Q}^*_{\alpha_i^{(\ell)},i} \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)$$

from which follows that

$$0 \leq \mathbb{E}_{\mathbb{Q}_i^*,\mu}\left[L(\theta,(x,y))\right] - \mathbb{E}_{\mathbb{Q}^*_{\alpha_i^{(\ell)},i},\mu}\left[L(\theta,(x,y))\right] \leq \alpha_i^{(\ell)}\left(\mathrm{KL}\left(\mathbb{Q}_i^* \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right) - \mathrm{KL}\left(\mathbb{Q}^*_{\alpha_i^{(\ell)},i} \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)\right)$$

Then by considering the limit superior we obtain that

$$\limsup_{\ell\to+\infty} \mathbb{E}_{\mathbb{Q}^*_{\alpha_i^{(\ell)},i},\mu}\left[L(\theta,(x,y))\right] = \mathbb{E}_{\mathbb{Q}_i^*,\mu}\left[L(\theta,(x,y))\right].$$

from which follows that

$$\mathbb{E}_{\mathbb{Q}_i^*,\mu}\left[L(\theta,(x,y))\right] \leq \mathbb{E}_{\mathbb{Q}^*,\mu}\left[L(\theta,(x,y))\right]$$

and by optimality of $\mathbb{Q}_i^*$ we obtain the desired result. □

[4] Indeed by considering a decreasing sequence of continuous and bounded functions which converge towards $\mathbb{E}_\mu\left[L(\theta,(x,y))\right]$ and by definition of the weak convergence the result follows.

[5] for $\alpha = 0$ the result is clear, and if $\alpha > 0$, note that $\mathrm{KL}\left(\cdot \middle|\middle| \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)$ is lower semi-continuous

By adding an entropic term to the objective, we obtain an explicit formulation of the supremum involved in the sum: as soon as $\boldsymbol{\alpha} > 0$ (which means that each $\alpha_i > 0$), each sub-problem becomes just the Fenchel-Legendre transform of $\mathrm{KL}(\cdot|\mathbb{U}_{(x_i,y_i)}/N)$ which has the following closed form:

$$\begin{aligned}&\sup_{\mathbb{Q}_i \in \Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}\left[L(\theta,(x,y))\right] - \alpha_i \mathrm{KL}\left(\mathbb{Q}_i || \frac{1}{N}\mathbb{U}_{(x_i,y_i)}\right)\\ &= \frac{\alpha_i}{N}\log\left(\int_{\mathcal{X}\times\mathcal{Y}} \exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,(x,y))\right]}{\alpha_i}\right) d\mathbb{U}_{(x_i,y_i)}\right).\end{aligned}$$

Finally, we end up with the following problem:

$$\inf_{\mu\in\mathcal{M}_1^+(\Theta)} \sum_{i=1}^N \frac{\alpha_i}{N}\log\left(\int \exp\frac{\mathbb{E}_{\mu}\left[L(\theta,(x,y))\right]}{\alpha_i} d\mathbb{U}_{(x_i,y_i)}\right).$$

In order to solve the above problem, one needs to compute the integral involved in the objective. To do so, we estimate it by randomly sampling $m_i \geq 1$ samples $(u_1^{(i)},\dots,u_{m_i}^{(i)}) \in (\mathcal{X}\times\mathcal{Y})^{m_i}$ from $\mathbb{U}_{(x_i,y_i)}$ for all $i \in \{1,\dots,N\}$ which leads to the following optimization problem

$$\inf_{\mu\in\mathcal{M}_1^+(\Theta)} \sum_{i=1}^N \frac{\alpha_i}{N}\log\left(\frac{1}{m_i}\sum_{j=1}^{m_i}\exp\frac{\mathbb{E}_{\mu}\left[L(\theta,u_j^{(i)})\right]}{\alpha_i}\right) \tag{4.7}$$

denoted $\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,\boldsymbol{m}}$ where $\boldsymbol{m} := (m_i)_{i=1}^N$ in the following. Now we aim at controlling the error made with our approximations. We decompose the error into two terms

$$|\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,\boldsymbol{m}} - \widehat{\mathcal{R}}_{adv}^{\varepsilon,*}| \leq |\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,*} - \widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,\boldsymbol{m}}| + |\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,*} - \widehat{\mathcal{R}}_{adv}^{\varepsilon,*}|$$

where the first one corresponds to the statistical error made by our estimation of the integral, and the second to the approximation error made by the entropic regularization of the objective. First, we show a control of the statistical error using Rademacher complexities [Bartlett and Mendelson, 2002].

Proposition 9. *Let $m \geq 1$ and $\alpha > 0$ and denote $\boldsymbol{\alpha} := (\alpha,\dots,\alpha) \in \mathbb{R}^N$ and $\boldsymbol{m} := (m,\dots,m) \in \mathbb{R}^N$. Then by denoting $\tilde{M} = \max(M,1)$, we have with a probability of at least $1-\delta$*

$$|\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,*} - \widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\varepsilon,\boldsymbol{m}}| \leq \frac{2e^{M/\alpha}}{N}\sum_{i=1}^N R_i + 6\tilde{M}e^{M/\alpha}\sqrt{\frac{\log(\frac{4}{\delta})}{2mN}}$$

where $R_i := \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\theta\in\Theta}\sum_{j=1}^m \sigma_j l(\theta,u_j^{(i)})\right]$ and $\boldsymbol{\sigma} := (\sigma_1,\dots,\sigma_m)$ with σ_i i.i.d. sampled as $\mathbb{P}[\sigma_i = \pm 1] = 1/2$.

Proof. Let us denote for all $\mu \in \mathcal{M}_1^+(\Theta)$,

$$\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\boldsymbol{\varepsilon},\mathbf{m}}(\mu) := \sum_{i=1}^N \frac{\alpha_i}{N}\log\left(\frac{1}{m_i}\sum_{j=1}^{m_i}\exp\frac{\mathbb{E}_{\mu}\left[L(\theta,u_j^{(i)})\right]}{\alpha_i}\right).$$

Let also consider $(\mu_n^{(\mathbf{m})})_{n\geq 0}$ and $(\mu_n)_{n\geq 0}$ two sequences such that

$$\widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\boldsymbol{\varepsilon},\mathbf{m}}(\mu_n^{(\mathbf{m})}) \xrightarrow[n\to+\infty]{} \widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\boldsymbol{\varepsilon},\mathbf{m}}, \quad \widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\boldsymbol{\varepsilon}}(\mu_n) \xrightarrow[n\to+\infty]{} \widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\boldsymbol{\varepsilon},*}.$$

We first remarks that

$$\begin{aligned}\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}} &\leq \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}(\mu_n) + \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}(\mu_n) - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon}}_{adv,\boldsymbol{\alpha}}(\mu_n) + \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon}}_{adv,\boldsymbol{\alpha}}(\mu_n) - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}} \\ &\leq \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \left|\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}(\mu) - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon}}_{adv,\boldsymbol{\alpha}}(\mu)\right| + \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon}}_{adv,\boldsymbol{\alpha}}(\mu_n) - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}},\end{aligned}$$

and by considering the limit, we obtain that

$$\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}} \leq \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \left|\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}(\mu) - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon}}_{adv,\boldsymbol{\alpha}}(\mu)\right|$$

Simarly we have that

$$\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}} \leq \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon}}_{adv,\boldsymbol{\alpha}}(\mu_n^{(\boldsymbol{m})}) + \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon}}_{adv,\boldsymbol{\alpha}}(\mu_n^{(\boldsymbol{m})}) - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}(\mu_n^{(\boldsymbol{m})}) + \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}(\mu_n^{(\boldsymbol{m})}) - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}$$

from which follows that

$$\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}} \leq \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \left|\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}(\mu) - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon}}_{adv,\boldsymbol{\alpha}}(\mu)\right|$$

Therefore we obtain that

$$\begin{aligned}\left|\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}\right| \leq \sum_{i=1}^N \frac{\alpha}{N} \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \Bigg| &\log\left(\frac{1}{m_i}\sum_{j=1}^{m_i} \exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,u_j^{(i)}))\right]}{\alpha}\right)\right) \\ &- \log\left(\int_{\mathcal{X}\times\mathcal{Y}} \exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,(x,y))\right]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}\right)\Bigg|.\end{aligned}$$

Observe that $L \geq 0$, therefore because the log function is 1-Lipschitz on $[1,+\infty)$, we obtain that

$$\begin{aligned}\left|\widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\boldsymbol{\varepsilon},\mathbf{m}}_{adv,\boldsymbol{\alpha}}\right| \leq \sum_{i=1}^N \frac{\alpha}{N} \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \Bigg| &\frac{1}{m_i}\sum_{j=1}^{m_i} \exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,u_j^{(i)}))\right]}{\alpha}\right) \\ &- \int_{\mathcal{X}\times\mathcal{Y}} \exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,(x,y))\right]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}\Bigg|.\end{aligned}$$

Let us now denote for all $i = 1,\ldots,N$,

$$\begin{aligned}\widehat{R}_i(\mu,\boldsymbol{u}^{(i)}) &:= \sum_{j=1}^{m_i} \exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,u_j^{(i)}))\right]}{\alpha}\right) \\ R_i(\mu) &:= \int_{\mathcal{X}\times\mathcal{Y}} \exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,(x,y))\right]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}.\end{aligned}$$

and let us define

$$f(\boldsymbol{u}^{(1)},\ldots,\boldsymbol{u}^{(N)}) := \sum_{i=1}^N \frac{\alpha}{N} \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \left|\widehat{R}_i(\mu) - R_i(\mu)\right|$$

where $\boldsymbol{u}^{(i)} := (u_1^{(i)}, \dots, u_1^{(m)})$. By denoting $z^{(i)} = (u_1^{(i)}, \dots, u_{k-1}^{(i)}, z, u_{k+1}^{(i)}, \dots, u_m^{(i)})$, we have that

$$\begin{aligned}
&|f(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(N)}) - f(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(i-1)}, \boldsymbol{z}^{(i)}, \boldsymbol{u}^{(i+1)}, \dots, \boldsymbol{u}^{(N)})| \\
&\qquad \leq \frac{\alpha}{N} \Big| \sup_{\mu \in \mathcal{M}_1^+(\Theta)} \left| \widehat{R}_i(\mu, \boldsymbol{u}^{(i)}) - R_i(\mu) \right| - \sup_{\mu \in \mathcal{M}_1^+(\Theta)} \left| \widehat{R}_i(\mu, \boldsymbol{z}^{(i)}) - R_i(\mu) \right| \Big| \\
&\qquad \leq \frac{\alpha}{N} \Big| \frac{1}{m} \left[\exp\left(\frac{\mathbb{E}_{\theta \sim \mu} \left[L(\theta, u_k^{(i)}) \right]}{\alpha} \right) - \exp\left(\frac{\mathbb{E}_{\theta \sim \mu} \left[L(\theta, z^{(i)}) \right]}{\alpha} \right) \right] \Big| \\
&\qquad \leq \frac{2 \exp(M/\alpha)}{Nm}
\end{aligned}$$

where the last inequality comes from the fact that the loss is upper bounded by $L \leq M$. Then by appling the McDiarmid's Inequality, we obtain that with a probability of at least $1 - \delta$,

$$\left| \widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\boldsymbol{\varepsilon},*} - \widehat{\mathcal{R}}_{adv,\boldsymbol{\alpha}}^{\boldsymbol{\varepsilon},\mathbf{m}} \right| \leq \mathbb{E}(f(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(N)})) + \frac{2 \exp(M/\alpha)}{\sqrt{mN}} \sqrt{\frac{\log(2/\delta)}{2}}.$$

Thanks to [Shalev-Shwartz and Ben-David, 2014, Lemma 26.2], we have for all $i \in \{1, \dots, N\}$

$$\mathbb{E}(f(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(N)})) \leq 2\mathbb{E}(\mathrm{Rad}(\mathcal{F}_i \circ \mathbf{u}^{(\mathbf{i})}))$$

where for any class of function $\mathcal{F}$ defined on $\mathcal{Z}$ and point $\boldsymbol{z} : (z_1, \dots, z_q) \in \mathcal{Z}^q$

$$\begin{aligned}
&\mathcal{F} \circ \boldsymbol{z} := \Big\{ (f(z_1), \dots, f(z_q)),\ f \in \mathcal{F} \Big\} \quad , \quad \mathrm{Rad}(\mathcal{F} \circ \boldsymbol{z}) := \frac{1}{q} \mathbb{E}_{\boldsymbol{\sigma} \sim \{\pm 1\}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{q} \sigma_i f(z_i) \right] \\
&\mathcal{F}_i := \Big\{ u \to \exp\left(\frac{\mathbb{E}_{\theta \sim \mu} \left[L(\theta, u) \right]}{\alpha} \right),\ \mu \in \mathcal{M}_1^+(\Theta) \Big\}.
\end{aligned}$$

Moreover as $x \to \exp(x/\alpha)$ is $\frac{\exp(M/\alpha)}{\alpha}$-Lipstchitz on $(-\infty, M]$, by [Shalev-Shwartz and Ben-David, 2014, Lemma 26.9], we have

$$\mathrm{Rad}(\mathcal{F}_i \circ \mathbf{u}^{(\mathbf{i})}) \leq \frac{\exp(M/\alpha)}{\alpha} \mathrm{Rad}(\mathcal{H}_i \circ \mathbf{u}^{(\mathbf{i})})$$

where

$$\mathcal{H}_i := \Big\{ u \to \mathbb{E}_{\theta \sim \mu} \left[L(\theta, u) \right],\ \mu \in \mathcal{M}_1^+(\Theta) \Big\}.$$

Let us now define

$$g(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(N)}) := \sum_{j=1}^{N} \frac{2 \exp(M/\alpha)}{N} \mathrm{Rad}(\mathcal{H}_j \circ \mathbf{u}^{(\mathbf{j})}).$$

We observe that

$$\begin{aligned}
&|g(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(N)}) - g(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(i-1)}, \boldsymbol{z}^{(i)}, \boldsymbol{u}^{(i+1)}, \dots, \boldsymbol{u}^{(N)})| \\
&\qquad \leq \frac{2 \exp(M/\alpha)}{N} |\mathrm{Rad}(\mathcal{H}_i \circ \mathbf{u}^{(\mathbf{i})}) - \mathrm{Rad}(\mathcal{H}_i \circ \mathbf{z}^{(\mathbf{i})})| \\
&\qquad \leq \frac{2 \exp(M/\alpha)}{N} \frac{2M}{m}.
\end{aligned}$$

By Applying the McDiarmid's Inequality, we have that with a probability of at least $1-\delta$

$$\mathbb{E}(g(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(N)})) \leq g(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(N)}) + \frac{4\exp(M/\alpha)M}{\sqrt{mN}}\sqrt{\frac{\log(2/\delta)}{2}}.$$

Remarks also that

$$\begin{aligned}\mathrm{Rad}(\mathcal{H}_i \circ \mathbf{u}^{(\mathbf{i})}) &= \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}\sim\{\pm 1\}}\left[\sup_{\mu\in\mathcal{M}_1^+(\Theta)}\sum_{j=1}^{m}\sigma_i\mathbb{E}_\mu(l(\theta,u_j^{(i)}))\right] \\ &= \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}\sim\{\pm 1\}}\left[\sup_{\theta\in\Theta}\sum_{j=1}^{m}\sigma_i l(\theta,u_j^{(i)})\right]\end{aligned}$$

Finally, applying a union bound leads to the desired result.

□

We deduce from the above Proposition that in the particular case where Θ is finite such that $|\Theta| = l$, with probability of at least $1-\delta$

$$|\widehat{\mathcal{R}}^{\varepsilon,*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\varepsilon,\boldsymbol{m}}_{adv,\boldsymbol{\alpha}}| \in \mathcal{O}\left(Me^{M/\alpha}\sqrt{\frac{\log(l)}{m}}\right).$$

This case is of particular interest when one wants to learn the optimal mixture of some given classifiers in order to minimize the adversarial risk. In the following proposition, we control the approximation error made by adding an entropic term to the objective.

Proposition 10. *Denote for $\beta > 0$, $(x,y) \in \mathcal{X}\times\mathcal{Y}$ and $\mu \in \mathcal{M}_1^+(\Theta)$,*

$$A^{(x,y)}_{\beta,\mu} := \{u | \sup_{v\in S^{(\varepsilon)}_{(x,y)}} \mathbb{E}_\mu[L(\theta,v)] \leq \mathbb{E}_\mu[L(\theta,u)] + \beta\}$$

. If there exists C_β such that for all $(x,y) \in \mathcal{X}\times\mathcal{Y}$ and $\mu \in \mathcal{M}_1^+(\Theta)$, $\mathbb{U}_{(x,y)}\left(A^{(x,y)}_{\beta,\mu}\right) \geq C_\beta$ then we have

$$|\widehat{\mathcal{R}}^{\varepsilon,*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\varepsilon,*}_{adv}| \leq 2\alpha|\log(C_\beta)| + \beta.$$

The assumption made in the above Proposition states that for any given random classifier μ, and any given point (x,y), the set of β-optimal attacks at this point has at least a certain amount of mass depending on the β chosen. This assumption is always met when β is sufficiently large. However in order to obtain a tight control of the error, a trade-off exists between β and the smallest amount of mass C_β of β-optimal attacks.

Proof. Following the same steps than the proof of Proposition 9, let $(\mu_n^\varepsilon)_{n\geq 0}$ and $(\mu_n)_{n\geq 0}$ two sequences such that

$$\widehat{\mathcal{R}}^{\varepsilon}_{adv,\boldsymbol{\alpha}}(\mu_n^\varepsilon) \xrightarrow[n\to+\infty]{} \widehat{\mathcal{R}}^{\varepsilon,*}_{adv,\boldsymbol{\alpha}}, \quad \widehat{\mathcal{R}}^{\varepsilon}_{adv}(\mu_n) \xrightarrow[n\to+\infty]{} \widehat{\mathcal{R}}^{\varepsilon,*}_{adv}.$$

Remarks that

$$\begin{aligned}\widehat{\mathcal{R}}^{\varepsilon,*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\varepsilon,*}_{adv} &\leq \widehat{\mathcal{R}}^{\varepsilon,*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\varepsilon}_{adv,\boldsymbol{\alpha}}(\mu_n) + \widehat{\mathcal{R}}^{\varepsilon}_{adv,\boldsymbol{\alpha}}(\mu_n) - \widehat{\mathcal{R}}^{\varepsilon}_{adv}(\mu_n) + \widehat{\mathcal{R}}^{\varepsilon}_{adv}(\mu_n) - \widehat{\mathcal{R}}^{\varepsilon,*}_{adv} \\ &\leq \sup_{\mu\in\mathcal{M}_1^+(\Theta)}\left|\widehat{\mathcal{R}}^{\varepsilon}_{adv,\boldsymbol{\alpha}}(\mu) - \widehat{\mathcal{R}}^{\varepsilon}_{adv}(\mu)\right| + \widehat{\mathcal{R}}^{\varepsilon}_{adv}(\mu_n) - \widehat{\mathcal{R}}^{\varepsilon,*}_{adv}\end{aligned}$$

Then by considering the limit we obtain that

$$\widehat{\mathcal{R}}^{\varepsilon,*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\varepsilon,*}_{adv} \leq \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \left|\widehat{\mathcal{R}}^{\varepsilon}_{adv,\boldsymbol{\alpha}}(\mu) - \widehat{\mathcal{R}}^{\varepsilon}_{adv}(\mu)\right|.$$

Similarly, we obtain that

$$\widehat{\mathcal{R}}^{\varepsilon,*}_{adv} - \widehat{\mathcal{R}}^{\varepsilon,*}_{adv,\boldsymbol{\alpha}} \leq \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \left|\widehat{\mathcal{R}}^{\varepsilon}_{adv,\boldsymbol{\alpha}}(\mu) - \widehat{\mathcal{R}}^{\varepsilon}_{adv}(\mu)\right|,$$

from which follows that

$$\begin{aligned}\left|\widehat{\mathcal{R}}^{\varepsilon,*}_{adv,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\varepsilon,*}_{adv}\right| \leq \frac{1}{N}\sum_{i=1}^{N} \sup_{\mu\in\mathcal{M}_1^+(\Theta)} \Bigg| \quad & \alpha\log\left(\int_{\mathcal{X}\times\mathcal{Y}} \exp\left(\frac{\mathbb{E}_\mu[L(\theta,(x,y))]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}\right) \\ & - \sup_{u\in S^{\varepsilon}_{(x_i,y_i)}} \mathbb{E}_\mu[L(\theta,u)]\Bigg|.\end{aligned}$$

Let $\mu\in\mathcal{M}_1^+(\Theta)$ and $i\in\{1,\dots,N\}$, then we have

$$\begin{aligned}&\left|\alpha\log\left(\int_{\mathcal{X}\times\mathcal{Y}} \exp\left(\frac{\mathbb{E}_\mu[L(\theta,(x,y))]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}\right) - \sup_{u\in S^{\varepsilon}_{(x_i,y_i)}} \mathbb{E}_\mu[L(\theta,u)]\right| \\ &= \left|\alpha\log\left(\int_{\mathcal{X}\times\mathcal{Y}} \exp\left(\frac{\mathbb{E}_\mu[L(\theta,(x,y))] - \sup_{u\in S^{\varepsilon}_{(x_i,y_i)}} \mathbb{E}_\mu[L(\theta,u)]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}\right)\right| \\ &= \alpha\Bigg|\log\left(\int_{A^{(x_i,y_i)}_{\beta,\mu}} \exp\left(\frac{\mathbb{E}_\mu[L(\theta,(x,y))] - \sup_{u\in S^{\varepsilon}_{(x_i,y_i)}} \mathbb{E}_\mu[L(\theta,u)]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}\right. \\ &+ \left.\int_{(A^{(x_i,y_i)}_{\beta,\mu})^c} \exp\left(\frac{\mathbb{E}_\mu[L(\theta,(x,y))] - \sup_{u\in S^{\varepsilon}_{(x_i,y_i)}} \mathbb{E}_\mu[L(\theta,u)]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}\right)\Bigg| \\ &\leq \alpha\left|\log\left(\exp(-\frac{\beta}{\alpha})\mathbb{U}_{(x_i,y_i)}\left(A^{(x_i,y_i)}_{\beta,\mu}\right)\right)\right| \\ &+ \alpha\left|\log\left(1 + \frac{\exp(\beta/\alpha)}{\mathbb{U}_{(x_i,y_i)}\left(A^{(x_i,y_i)}_{\beta,\mu}\right)} \int_{(A^{(x_i,y_i)}_{\beta,\mu})^c} \exp\left(\frac{\mathbb{E}_\mu[L(\theta,(x,y))] - \sup_{u\in S^{\varepsilon}_{(x_i,y_i)}} \mathbb{E}_\mu[L(\theta,u)]}{\alpha}\right) d\mathbb{U}_{(x_i,y_i)}\right)\right| \\ &\leq \alpha\log(1/C_\beta) + \beta + \frac{\alpha}{C_\beta} \\ &\leq 2\alpha\log(1/C_\beta) + \beta\end{aligned}$$

□

Now that we have shown that solving (4.7) allows to obtain an approximation of the true solution $\widehat{\mathcal{R}}^{\varepsilon,*}_{adv}$, we next aim at deriving an algorithm to compute it.

4.3.2 Proposed Algorithms

From now on, we focus on finite class of classifiers. Let $\Theta = \{\theta_1,\dots,\theta_l\}$, we aim to learn the optimal mixture of classifiers in this case. The adversarial empirical risk is therefore defined as:

$$\widehat{\mathcal{R}}^{\varepsilon}_{adv}(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \sup_{\mathbb{Q}_i\in\Gamma_{i,\varepsilon}} \mathbb{E}_{(x,y)\sim\mathbb{Q}_i}\left[\sum_{k=1}^{l} \lambda_k L(\theta_k,(x,y))\right]$$

for $\boldsymbol{\lambda} \in \Delta_l := \{\boldsymbol{\lambda} \in \mathbb{R}^l_+ \text{ s.t. } \sum_{i=1}^l \lambda_i = 1\}$, the probability simplex of $\mathbb{R}^l$. One can notice that $\widehat{\mathcal{R}}^{\varepsilon}_{adv}(\cdot)$ is a continuous convex function, hence $\min_{\boldsymbol{\lambda}\in\Delta_l} \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda})$ is attained for a certain $\boldsymbol{\lambda}^*$. Then there exists a non-approximate Nash equilibrium $(\boldsymbol{\lambda}^*, \mathbb{Q}^*)$ in the adversarial game when Θ is finite. Here, we present two algorithms to learn the optimal mixture of the adversarial risk minimization problem.

Algorithm 3: Oracle-based Algorithm

$\boldsymbol{\lambda}_0 = \frac{\mathbf{1}_L}{L}; T;\ \eta = \frac{2}{M\sqrt{LT}}$

for $t = 1, \dots, T$ **do**

  $\tilde{\mathbb{Q}}$ s.t. $\exists \mathbb{Q}^* \in \mathcal{A}_{\varepsilon}(\mathbb{P})$ best response to $\boldsymbol{\lambda}_{t-1}$ and for all $k \in [L]$,
  $|\mathbb{E}_{\tilde{\mathbb{Q}}}(l(\theta_k, (x, y))) - \mathbb{E}_{\mathbb{Q}^*}(l(\theta_k, (x, y)))| \leq \delta$

  $\boldsymbol{g}_t = \left(\mathbb{E}_{\tilde{\mathbb{Q}}}(l(\theta_1, (x, y)), \dots, \mathbb{E}_{\tilde{\mathbb{Q}}}(l(\theta_L, (x, y))\right)^T$

  $\boldsymbol{\lambda}_t = \Pi_{\Delta_L}\left(\boldsymbol{\lambda}_{t-1} - \eta \boldsymbol{g}_t\right)$

end

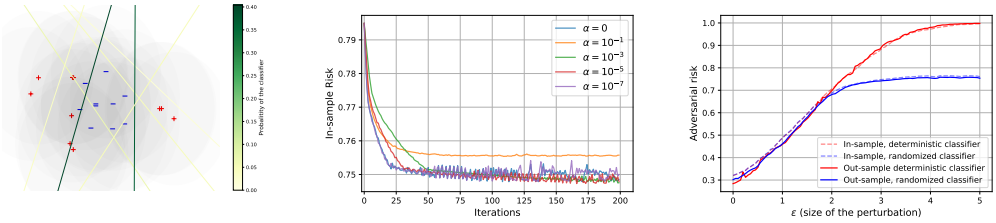


Figure 4.2: On left, 40 data samples with their set of possible attacks represented in shadow and the optimal randomized classifier, with a color gradient representing the probability of the classifier. In the middle, convergence of the oracle ($\alpha = 0$) and regularized algorithm for different values of regularization parameters. On right, in-sample and out-sample risk for randomized and deterministic minimum risk in function of the perturbation size ε. In the latter case, the randomized classifier is optimized with oracle Algorithm 3.

An Entropic Relaxation. Using the results from Section 4.3.1, adding an entropic term to the objective allows to have a simple reformulation of the problem, as follows:

$$\inf_{\boldsymbol{\lambda}\in\Delta_l} \sum_{i=1}^{N} \frac{\varepsilon_i}{N} \log\left(\frac{1}{m_i}\sum_{j=1}^{m_i} \exp\left(\frac{\sum_{k=1}^{l} \lambda_k L(\theta_k, u_j^{(i)})}{\varepsilon_i}\right)\right)$$

Note that in $\boldsymbol{\lambda}$, the objective is convex and smooth. One can apply the accelerated PGD [Beck and Teboulle, 2009, Tseng, 2008] which enjoys an optimal convergence rate for first order methods of $\mathcal{O}(T^{-2})$ for T iterations.

A First Oracle Algorithm. Indepedently from the entropic regularization,we present an oracle-based algorithm inspired from [Sinha et al., 2017] and the convergence of projected sub-gradient methods [Boyd, 2003]. The computation of the inner supremum problem is usually NP-hard. Let us justify it on a mixture of linear classifiers in binary classification: $f_{\theta_k, b_k}(x) = \langle \theta, x \rangle + b_k$ for $k \in [L]$ and $\boldsymbol{\lambda} = \mathbf{1}_L/L$. Let us consider the ℓ_2 norm and $x = 0$ and $y = 1$. Then the problem

of attacking x is the following:

$$\sup_{\tau,\ \|\tau\|\leq\varepsilon} \frac{1}{L}\sum_{k=1}^{L} \mathbf{1}_{\langle\theta_k, x+\tau\rangle+b_k\leq 0}$$

This problem is equivalent to a linear binary classification problem on τ, which is known to be NP-hard. Assuming the existence of a δ-approximate oracle to this supremum, we algorithm is presented in Algorithm 3. We get the following guarantee for this algorithm.

Proposition 11. *Let $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ satisfying Assumption 1. Then, Algorithm 3 satisfies:*

$$\min_{t\in[T]} \widehat{\mathcal{R}}^{\varepsilon}_{adv}(\boldsymbol{\lambda}_t) - \widehat{\mathcal{R}}^{\varepsilon,*}_{adv} \leq 2\delta + \frac{2M\sqrt{l}}{\sqrt{T}}$$

Proof. Thanks to Danskin theorem, if $\mathbb{Q}^*$ is a best response to $\boldsymbol{\lambda}$, then

$$\boldsymbol{g}^* := \left(\mathbb{E}_{\mathbb{Q}^*}\left[L(\theta_1,(x,y))\right], \ldots, \mathbb{E}_{\mathbb{Q}^*}\left[L(\theta_l,(x,y))\right]\right)^T$$

is a subgradient of $\boldsymbol{\lambda} \to \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda})$. Let $\eta \geq 0$ be the learning rate. Then we have for all $t \geq 1$:

$$\begin{aligned}
\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*\|^2 &\leq \|\boldsymbol{\lambda}_{t-1} - \eta\boldsymbol{g}_t - \boldsymbol{\lambda}^*\|^2 \\
&= \|\boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^*\|^2 - 2\eta\langle \boldsymbol{g}_t, \boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^*\rangle + \eta^2\|\boldsymbol{g}_t\|^2 \\
&\leq \|\boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^*\|^2 - 2\eta\langle \boldsymbol{g}^*_t, \boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^*\rangle + 2\eta\langle \boldsymbol{g}^*_t - \boldsymbol{g}_t, \boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^*\rangle + \eta^2 M^2 l \\
&\leq \|\boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^*\|^2 - 2\eta\left(\mathcal{R}^{\varepsilon}(\boldsymbol{\lambda}_t) - \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda}^*)\right) + 4\eta\delta + \eta^2 M^2 l
\end{aligned}$$

We then deduce by summing:

$$2\eta\sum_{t=1}^{T} \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda}_t) - \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda}^*) \leq 4\delta\eta T + \|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^*\|^2 + \eta^2 M^2 l T$$

Then we have:

$$\min_{t\in[T]} \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda}_t) - \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda}^*) \leq 2\delta + \frac{4}{\eta T} + M^2 l\eta$$

The left-hand term is minimal for $\eta = \frac{2}{M\sqrt{lT}}$, and for this value:

$$\min_{t\in[T]} \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda}_t) - \mathcal{R}^{\varepsilon}(\boldsymbol{\lambda}^*) \leq 2\delta + \frac{2M\sqrt{l}}{\sqrt{T}}$$

□

.

The main drawback of the above algorithm is that one needs to have access to an oracle to guarantee the convergence of the proposed algorithm whereas its regularized version in order to approximate the solution and propose a simple algorithm to solve it.

4.3.3 A General Heuristic Algorithm

So far, our algorithms are not easily practicable in the case of deep learning. Adversarial examples are known to be easily transferrable from one model to another [Papernot et al., 2016a, Tramèr et al., 2017]. So we aim at learning diverse models. To this end, and support our theoretical claims, we propose an heuristic algorithm (see Algorithm 4) to train a robust mixture of l classifiers. We alternatively train these classifiers with adversarial examples against the current mixture and update the probabilities of the mixture according to the algorithms we proposed in Section 4.3.2.

Algorithm 4: Adversarial Training for Mixtures

l: number of models, T: number of iterations,
T_θ: number of updates for the models $\boldsymbol{\theta}$,
T_λ: number of updates for the mixture $\boldsymbol{\lambda}$,
$\boldsymbol{\lambda}_0 = (\lambda_0^1, \dots \lambda_0^l)$, $\boldsymbol{\theta}_0 = (\theta_0^1, \dots \theta_0^l)$
for $t = 1, \dots, T$ **do**
  Let B_t be a batch of data.
  if $t \mod (T_\theta l + 1) \neq 0$ **then**
    k sampled uniformly in $\{1, \dots, l\}$
    $\tilde{B}_t \leftarrow$ Attack of images in B_t for the model $(\boldsymbol{\lambda}_t, \boldsymbol{\theta}_t)$
    $\theta_k^t \leftarrow$ Update θ_k^{t-1} with $\tilde{B}_t$ for fixed $\boldsymbol{\lambda}_t$ with a SGD step
  else
    $\boldsymbol{\lambda}_t \leftarrow$Update $\boldsymbol{\lambda}_{t-1}$ on B_t for fixed $\boldsymbol{\theta}_t$ with oracle-based or regularized algorithm with T_λ iterations.
  end
end

4.4 Experiments

4.4.1 Synthetic Dataset

To illustrate our theoretical findings, we start by testing our learning algorithm on the following synthetic two-dimensional problem. Let us consider the distribution $\mathbb{P}$ defined as $\mathbb{P}(Y = \pm 1) = 1/2$, $\mathbb{P}(X \mid Y = -1) = \mathcal{N}(0, I_2)$ and $\mathbb{P}(X \mid Y = 1) = \frac{1}{2}[\mathcal{N}((-3, 0), I_2) + \mathcal{N}((3, 0), I_2)]$. We sample 1000 training points from this distribution and randomly generate 10 linear classifiers that achieves a standard training risk lower than 0.4. To simulate an adversary with budget ε in ℓ_2 norm, we proceed as follows. For every sample $(x, y) \sim \mathbb{P}$ we generate 1000 points uniformly at random in the ball of radius ε and select the one maximizing the risk for the 0/1 loss. Figure 4.2 (left) illustrates the type of mixture we get after convergence of our algorithms. Note that in this toy problem, we are likely to find the optimal adversary with this sampling strategy if we sample enough attack points.

To evaluate the convergence of our algorithms, we compute the adversarial risk of our mixture for each iteration of both the oracle and regularized algorithms. Figure 4.2 illustrates the convergence of the algorithms w.r.t the regularization parameter. We observe that the risk for both algorithms converge. Moreover, they converge towards the oracle minimizer when the regularization parameter α goes to 0.

Finally, to demonstrate the improvement randomized techniques offer against deterministic defenses, we plot in Figure 4.2 (right) the minimum adversarial risk for both randomized and

deterministic classifiers w.r.t. ε. The adversarial risk is strictly better for randomized classifier whenever the adversarial budget ε is bigger than 2. This illustration validates our analysis of Theorem 9, and motivates a in depth study of a more challenging framework, namely image classification with neural networks.

4.4.2 CIFAR Datasets

Experimental Setup. We now implement our heuristic algorithm (Alg. 4) on CIFAR-10 and CIFAR-100 datasets for both Adversarial Traning [Madry et al., 2018] and TRADES [Zhang et al., 2019a] loss. To evaluate the performance of Algorithm 4, we trained from 1 to 4 ResNet18 [He et al., 2016] models on 200 epochs per model[6]. We study the robustness with regards to ℓ_∞ norm and fixed adversarial budget $\varepsilon = 8/255$. The attack we used in the inner maximization of the training is an adapted (adaptative) version of PGD for mixtures of classifiers with 10 steps. Note that for one single model, Algorithm 4 exactly corresponds to adversarial training [Madry et al., 2018] or TRADES. For each of our setups, we made two independent runs and select the best one. The training time of our algorithm is around four times longer than a standard Adversarial Training (with PGD 10 iter.) with two models, eight times with three models and twelve times with four models. We trained our models with a batch of size 1024 on 8 Nvidia V100 GPUs.

Optimizer. For each of our models, The optimizer we used in all our implementations is SGD with learning rate set to 0.4 at epoch 0 and is divided by 10 at half training then by 10 at the three quarters of training. The momentum is set to 0.9 and the weight decay to 5×10^{-4}. The batch size is set to 1024.

Adaptation of Attacks. Since our classifier is randomized, we need to adapt the attack accordingly. To do so we used the expected loss:

$$\tilde{L}\left((\boldsymbol{\lambda}, \boldsymbol{\theta}), (x, y)\right) = \sum_{k=1}^{L} \lambda_k L(\theta_k, (x, y))$$

to compute the gradient in the attacks, regardless the loss (DLR or cross-entropy). For the inner maximization at training time, we used a PGD attack on the cross-entropy loss with $\varepsilon = 0.03$. For the final evaluation, we used the untargeted DLR attack with default parameters.

Regularization in Practice. The entropic regularization in higher dimensional setting need to be adapted to be more likely to find adversaries. To do so, we computed PGD attacks with only 3 iterations with 5 different restarts instead of sampling uniformly 5 points in the ℓ_∞-ball. In our experiments in the main paper, we use a regularization parameter $\alpha = 0.001$. The learning rate for the minimization on $\boldsymbol{\lambda}$ is always fixed to 0.001.

Alternate Minimization Parameters. Algorithm 4 implies an alternate minimization algorithm. We set the number of updates of $\boldsymbol{\theta}$ to $T_\theta = 50$ and, the update of $\boldsymbol{\lambda}$ to $T_\lambda = 25$.

4.4.3 Effect of the Regularization

In this subsection, we experimentally investigate the effect of the regularization. In Figure 4.4, we notice, that the regularization has the effect of stabilizing, reducing the variance and improving

[6] $L \times 200$ epochs in total, where L is the number of models.

the level of the robust accuracy for adversarial training for mixtures (Algorithm 4). The standard accuracy curves are very similar in both cases.

Evaluation Protocol. At each epoch, we evaluate the current mixture on test data against PGD attack with 20 iterations. To select our model and avoid overfitting [Rice et al., 2020], we kept the most robust against this PGD attack. To make a final evaluation of our mixture of models, we used an adapted version of AutoPGD untargeted attacks [Croce et al., 2020b] for randomized classifiers with both Cross-Entropy (CE) and Difference of Logits Ratio (DLR) loss. For both attacks, we made 100 iterations and 5 restarts.

Results. The results are presented in Figure 4.3. We remark our algorithm outperforms a standard adversarial training in all the cases by more 1% on CIFAR-10 and CIFAR-100, without additional loss of standard accuracy as it is attested by the left figures. On TRADES, the gain is even more important by more than 2% in robust accuracy. Moreover, it seems our algorithm, by adding more and more models, reduces the overfitting of adversarial training. It also appears that robustness increases as the number of models increases. So far, experiments are computationally very costful and it is difficult to raise precise conclusions. Further, hyperparameter tuning [Gowal et al., 2020] such as architecture, unlabeled data [Carmon et al., 2019] or activation function may still increase the results.

4.4.4 Additional Experiments on WideResNet28x10

We now evaluate our algorithm on WideResNet28x10 Zagoruyko and Komodakis [2016] architecture. Due to computation costs, we limit ourselves to 1 and 2 models, with regularization parameter set to 0.001 as in the paper experiments section. Results are reported in Figure 4.5. We remark this architecture can lead to more robust models, corroborating the results from Gowal et al. [2020].

4.4.5 Overfitting in Adversarial Robustness

We further investigate the overfitting of our heuristic algorithm. We plotted in Figure 4.6 the robust accuracy on ResNet18 with 1 to 5 models. The most robust mixture of 5 models against PGD with 20 iterations arrives at epoch 198, *i.e.* at the end of the training, contrary to 1 to 4 models, where the most robust mixture occurs around epoch 101. However, the accuracy against AGPD with 100 iterations in lower than the one at epoch 101 with global robust accuracy of 47.6% at epoch 101 and 45.3% at epoch 198. This strange phenomenon would suggest that the more powerful the attacks are, the more the models are subject to overfitting. We leave this question to further works.

4.5 Discussions and Open Questions

On the need of Randomization. While we give a concrete example where randomized is needed to be optimal in Section 4.1.1, [Pydi and Jog, 2021b] show there is no duality gap when the classifier is allowed to play a deterministic measurable classifier. In other words, randomization would not be useful for this game. We conjecture, as the hypothesis class Θ grows, the duality gap decreases to 0. However, in finite samples cases, it is not realistic to optimize over the space of measurable functions. One may ask if we could find conditions on the space of classifiers and the distribution $\mathbb{P}$ such that randomization is required. Pinot et al.

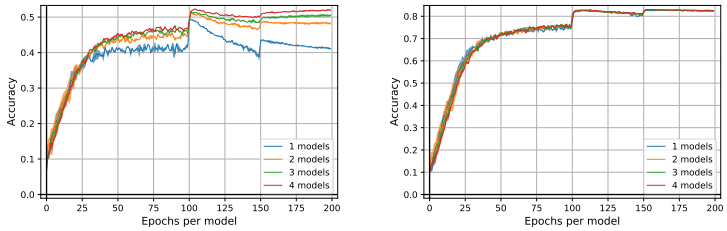
[2020] partially answered this question when the attacker is regularized, but the general case is still an open question.

Statistical guarantees for randomized classifiers. Although it is possible to derive uniform convergence bounds for the adversarial classification problem [Awasthi et al., 2020, Yin et al., 2019] for deterministic classifiers, deriving bounds for randomized classifiers is still an open question. One may think of adpating PAC-Bayes bounds [Guedj, 2019] but the proof scheme cannot apply for adversarial classification. A first attempt to derive such bounds was proposed by Viallard et al. [2021], but there is still much to do on this subject.

Learning Optimal Randomized Classifiers. For a given loss, learning the optimal randomized classifier for a continuous parameter space is also an open question. It is a difficult question since it requires learning over the space of ditributions. Attempts have been made to optimize over the space of distributions [Chizat, 2021a,b, Kent et al., 2021] often using Wasserstein Gradient Flows [Ambrosio et al., 2005] and particular flows [Wibisono, 2018]. Recently, Domingo-Enrich et al. [2020] proposed a particular flow to optimize a minmax problem in the space of distributions. While this paper gives good insights, the results are to preliminary to be adapted and applied to adversarial learning problems.

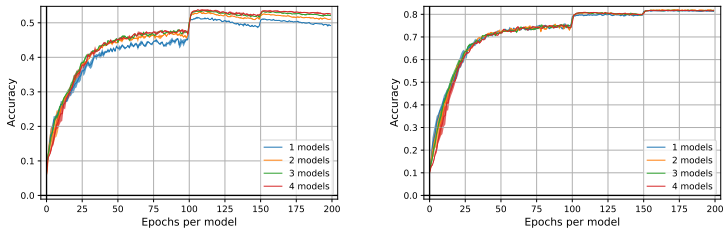
Adversarial Training, CIFAR-10 dataset results

Models	Acc.	APGD_{CE}	APGD_{DLR}	Rob. Acc.
1	81.9%	47.6%	47.7%	45.6%
2	81.9%	49.0%	49.6%	47.0%
3	81.7%	49.0%	49.3%	46.9%
4	**82.6%**	**49.7%**	**49.8%**	**47.2%**



TRADES, CIFAR-10 dataset results

Models	Acc.	APGD_{CE}	APGD_{DLR}	Rob. Acc.
1	79.6%	50.9%	48.9%	48.3%
2	80.3%	52.3%	51.2%	50.2%
3	80.7%	52.8%	51.7%	50.7%
4	**80.9%**	**53.0%**	**51.8%**	**50.8%**



Adversarial Training, CIFAR-100 dataset results

Models	Acc.	APGD_{CE}	APGD_{DLR}	Rob. Acc.
1	55.2%	24.1%	23.8%	22.5%
2	55.2%	25.3%	26.1%	23.6%
3	**55.4%**	25.7%	26.8%	24.2%
4	55.3%	**26.0%**	**27.5%**	**24.5%**

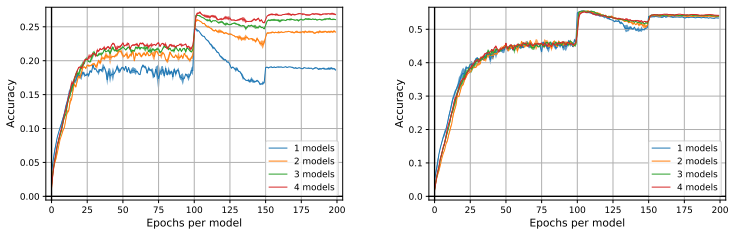


Figure 4.3: Upper plots: Adversarial Training, CIFAR-10 dataset results. Middle plots: TRADES, CIFAR-10 dataset results. Bottom plots: CIFAR-100 dataset results. On left: Comparison of our algorithm with a standard adversarial training (one model). We reported the results for the model with the best robust accuracy obtained over two independent runs because adversarial training might be unstable. Standard and Robust accuracy (respectively in the middle and on right) on CIFAR-10 test images in function of the number of epochs per classifier with 1 to 3 ResNet18 models. The performed attack is PGD with 20 iterations and $\varepsilon = 8/255$.

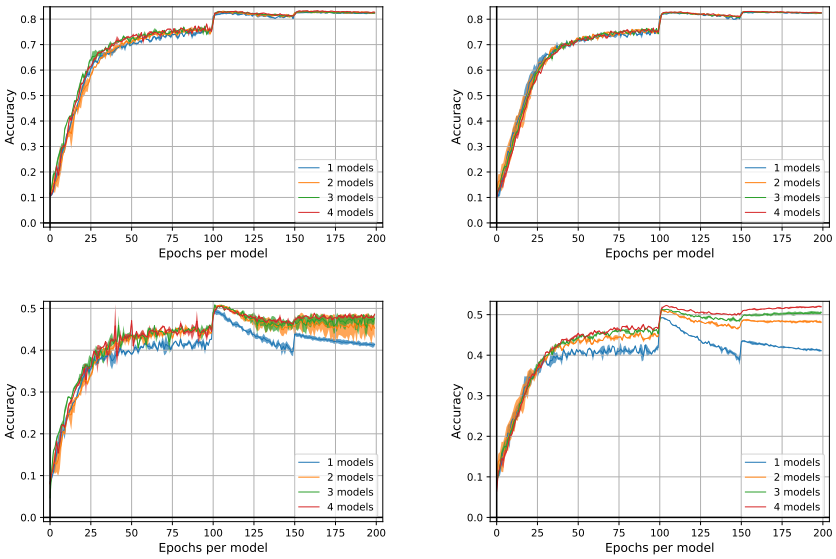


Figure 4.4: On top: Standard accuracies over epochs with respectively no regularization and regularization set to $\alpha = 0.001$. On bottom: Robust accuracies for the same parameters against PGD attack with 20 iterations and $\varepsilon = 0.03$.

Models	**Acc.**	**APGD$_{\text{CE}}$**	**APGD$_{\text{DLR}}$**	**Rob. Acc.**
1	85.2%	49.9%	50.2%	48.5%
2	**86.0%**	**51.5%**	**52.1%**	**49.6%**

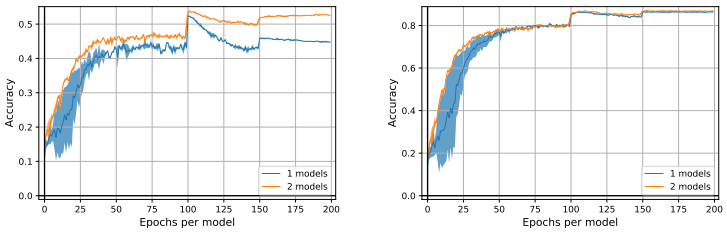


Figure 4.5: Comparison of our algorithm with a standard adversarial training (one model) on WideResNet28x10. We reported the results for the model with the best robust accuracy obtained over two independent runs because adversarial training might be unstable. Standard and Robust accuracy (respectively in the middle and on right) on CIFAR-10 test images in function of the number of epochs per classifier with 1 and 2 WideResNet28x10 models. The performed attack is PGD with 20 iterations and $\varepsilon = 8/255$.

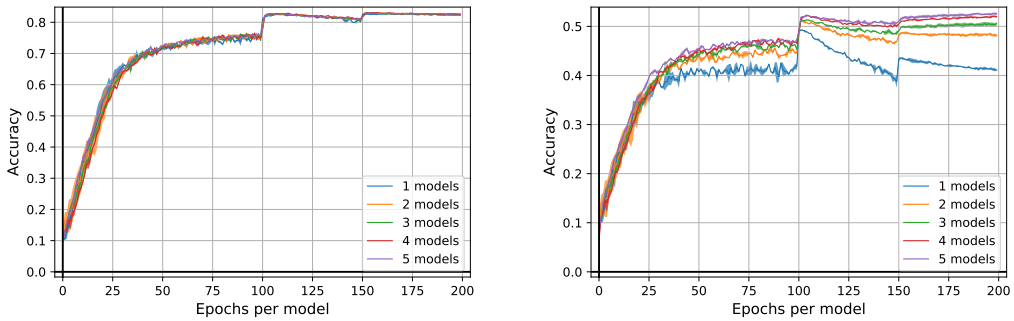


Figure 4.6: Standard and Robust accuracy (respectively on left and on right) on CIFAR-10 test images in function of the number of epochs per classifier with 1 to 5 ResNet18 models. The performed attack is PGD with 20 iterations and $\varepsilon = 8/255$. The best mixture for 5 models occurs at the end of training (epoch 198).

Chapter 5

Calibration and Consistency in Presence of Adversarial Attacks

Contents

The objective of this chapter is to study the problem of calibration and consistency in presence of adversaries. We study, in Section 5.1, the problem of calibration in the adversarial setting and provide both necessary and sufficient conditions for a loss to be calibrated in this setting. It also worth noting that our results are easily extendable to $\mathcal{H}$-calibration (see Section 5.1.4). One on the main takeaway of our analysis is that no convex surrogate loss can be calibrated in the adversarial setting. We however characterize a set of non-convex loss functions, namely *shifted odd functions* that solve the calibration problem in the adversarial setting. Finally, we focus on the problem of consistency in the adversarial setting in Section 5.2. Based on min-max arguments, we provide insights that might help paving a way to prove consistency of shifted odd functions in the adversarial setting. Specifically, we proved strong duality results for these losses and show tight links with the 0/1-loss. From these insights, we are able to provide a close but weaker property to consistency.

Notations. Let us consider a classification task with input space $\mathcal{X}$ and output space $\mathcal{Y} = \{-1, +1\}$. Let $(\mathcal{X}, d)$ be a proper Polish (i.e. completely separable) metric space representing the inputs space. For all $x \in \mathcal{X}$ and $\delta > 0$, we denote $B_\delta(x)$ the closed ball of radius δ and center x. We also assume that for all $x \in \mathcal{X}$ and $\delta > 0$, $B_\delta(x)$ contains at least two points[1]. Let us also endow $\mathcal{Y}$ with the trivial metric $d'(y, y') = \mathbf{1}_{y \neq y'}$. Then the space $(\mathcal{X} \times \mathcal{Y}, d \oplus d')$ is

[1]For instance, for any norm $\|\cdot\|$, $(\mathbb{R}^d, \|\cdot\|)$ is a Polish metric space satisfying this property.

a proper Polish space. For any Polish space $\mathcal{Z}$, we denote $\mathcal{M}_+^1(\mathcal{Z})$ the Polish space of Borel probability measures on $\mathcal{Z}$. We will denote $\mathcal{F}(\mathcal{Z})$ the space of real valued Borel measurable functions on $\mathcal{Z}$. Finally, we denote $\bar{\mathbb{R}} := \mathbb{R} \cup \{\infty, +\infty\}$. Moreover, we take back the definitions introduced in Section 3.2.

5.1 Solving Adversarial Calibration

In this section, we study the calibration of adversarial margin losses with regards to the adversarial 0/1 loss. We first provide necessary and sufficient conditions under which margin losses are adversarially calibrated. We then show that a wide range of surrogate losses that are calibrated in the standard setting are not calibrated in the adversarial setting. Finally we propose a class of losses that are calibrated in the adversarial setting, namely the *shifted losses*.

5.1.1 Necessary and Sufficient Conditions for Calibration

One of our main contributions is to find necessary and sufficient conditions for calibration in the adversarial setting. In a nutshell, we identify that for studying calibration it is central to understand the case where there might be indecesions for classifiers (i.e. $\eta = 1/2$). Indeed in this case, either labelling positevely or negatively the input x would lead the same loss for x. Next result provides a necessary conditions for calibration.

Theorem 11 (Necessary conditions for Calibration)**.** *Let ϕ be a continuous margin loss and $\varepsilon > 0$. If ϕ is adversarially calibrated at level ε, then ϕ is calibrated in the standard classification setting and $0 \notin \operatorname{argmin}_{\alpha \in \bar{\mathbb{R}}} \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha)$.*

While the condition of calibration in the standard classification setting seems natural, we need to understand why $0 \notin \operatorname{argmin}_{\alpha \in \bar{\mathbb{R}}} \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha)$. The intuition behind the results in that a sequence of functions simply converging towards 0 in the ball of radius ε around some x can take positive and negative values thus leading to suboptimal 0/1 adversarial risk.

Proof. Let show that if $0 \in \operatorname{argmin}_{\alpha \in \mathbb{R}} \phi(\alpha) + \phi(-\alpha)$ then ϕ is not calibrated for the adversarial problem. For that, let $x \in \mathcal{X}$ and we fix $\eta = \frac{1}{2}$. For $n \geq 1$, we define $f_n(u) = \frac{1}{n}$ for $u \neq x$ and $-\frac{1}{n}$ for $u = x$. Since $|\mathcal{B}_\varepsilon(x)| \geq 2$, we have

$$\mathcal{C}_{\phi_\varepsilon}(x, \frac{1}{2}, f_n) = \max\left(\phi(\frac{1}{n}), \phi(-\frac{1}{n})\right) \xrightarrow[n\to\infty]{} \phi(0)$$

As, $\phi(0) = \inf_{\alpha \in \mathbb{R}} \frac{1}{2}\left(\phi(\alpha) + \phi(-\alpha)\right)$, the above means that $(f_n)_n$ is a minimizing sequence for $\alpha \mapsto \frac{1}{2}\left(\phi(\alpha) + \phi(-\alpha)\right)$. Then thanks to Proposition 2, $(f_n)_n$ is also a minimizing sequence for $f \mapsto \mathcal{C}_{\phi_\varepsilon}(x, \frac{1}{2}, f)$. However, for every integer n, we have $\mathcal{C}_{0/1,\varepsilon}(x, \frac{1}{2}, f_n) = 1 \neq \frac{1}{2}$. As $\inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{C}_\varepsilon(x, \frac{1}{2}, f) = \frac{1}{2}$, ϕ is not calibrated with regards to the 0/1 loss in the adversarial setting at level ε. We also immediately notice that if ϕ is calibrated with to 0/1 loss in the adversarial setting at level ε then ϕ is calibrated in the standard setting.

□

It turns out that, given an additional assumption, this condition is actually sufficient to ensure calibration.

Theorem 12 (Sufficient conditions for Calibration)**.** *Let ϕ be a continuous margin loss and $\varepsilon > 0$. If ϕ is strictly decreasing in a neighbourhood of 0 and calibrated in the standard setting and $0 \notin \operatorname{argmin}_{\alpha \in \bar{\mathbb{R}}} \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha)$, then ϕ is adversarially uniformly calibrated at level ε.*

Proof. Let $\epsilon \in (0, \frac{1}{2})$. Thanks to Theorem 8, ϕ is uniformly calibrated in the standard setting, then there exists $\delta > 0$, such that for all $x \in \mathcal{X}$, $\eta \in [0,1]$, $f \in \mathcal{F}(\mathcal{X})$:

$$\mathcal{C}_\phi(x,\eta,f) - \mathcal{C}_\phi^\star(x,\eta) \leq \delta \implies \mathcal{C}_{0/1}(x,\eta,f) - \mathcal{C}_{0/1}^\star(x,\eta) \leq \epsilon.$$

Case $\eta \neq \frac{1}{2}$: Let $x \in \mathcal{X}$ and $f \in \mathcal{F}(\mathcal{X})$ such that:

$$\mathcal{C}_{\phi_\varepsilon}(x,\eta,f) - \mathcal{C}_{\phi_\varepsilon}^\star(x,\eta) = \sup_{u,v \in B_\varepsilon(x)} \eta\phi(f(u)) + (1-\eta)\phi(-f(v)) - \mathcal{C}_{\phi_\varepsilon}^\star(x,\eta) \leq \delta$$

We recall thanks to Proposition 2 that for every $u, v \in \mathcal{X}$ $\mathcal{C}_{\phi_\varepsilon}^\star(u,\eta) = \mathcal{C}_\phi^\star(v,\eta) = \inf_{\alpha\in\mathbb{R}} \eta\phi(\alpha) + (1-\eta)\phi(-\alpha)$. Then in particular, for all $x' \in B_\varepsilon(x)$, we have:

$$\mathcal{C}_\phi(x',\eta,f) - \mathcal{C}_\phi^\star(x',\eta) \leq \sup_{u,v \in B_\varepsilon(x)} \eta\phi(f(u)) + (1-\eta)\phi(-f(v)) - \mathcal{C}_{\phi_\varepsilon}^\star(x,\eta) \leq \delta \quad .$$

Then since ϕ is calibrated for standard classification, for all $x' \in B_\varepsilon(x)$, $\mathcal{C}(x',\eta,f) - \mathcal{C}^\star(x',\eta) \leq \epsilon$. Since, $\epsilon < \frac{1}{2}$, we have $\mathcal{C}(x',\eta,f) = \mathcal{C}^\star(x',\eta)$ and then for all $x' \in B_\varepsilon(x)$, $f(x') < 0$ if $\eta < 1/2$ or $f(x') \geq 0$ if $\eta > 1/2$. We then deduce that

$$\mathcal{C}_\varepsilon(x,\eta,f) = \eta \sup_{x' \in B_\varepsilon(x)} \mathbf{1}_{f(x')\leq 0} + (1-\eta) \sup_{x' \in B_\varepsilon(x)} \mathbf{1}_{f(x')>0} = \min(\eta, 1-\eta) = \mathcal{C}_\varepsilon^\star(x,\eta)$$

Then we deduce, $\mathcal{C}_\varepsilon(x,\eta,f) - \mathcal{C}_\varepsilon^\star(x,\eta) \leq \epsilon$.

Case $\eta = \frac{1}{2}$: This shows us that calibration problems will only arise when $\eta = \frac{1}{2}$, i.e. on points where the Bayes classifier is indecise. For this case, we will reason by contradiction: we can construct a sequence of points α_n and β_n, whose risk converge to the same optimal value, while one sequence remains close to some positive value, and the other to some negative value. Assume that for all n, there exist $f_n \in \mathcal{F}(\mathcal{X})$ and $x_n \in \mathcal{X}$ such that

$$\mathcal{C}_{\phi_\varepsilon}(x_n, \frac{1}{2}, f_n) - \mathcal{C}_{\phi_\varepsilon}^\star(x_n, \frac{1}{2}) \leq \frac{1}{n} \text{ and exists } u_n, v_n \in B_\varepsilon(x_n),\ f_n(u_n) \times f_n(v_n) \leq 0.$$

Let denote $\alpha_n = f_n(u_n)$ and $\beta_n = f_n(v_n)$. Moreover, we have, thanks to Proposition 2:

$$0 \leq \frac{1}{2}\phi(\alpha_n) + \frac{1}{2}\phi(-\alpha_n) - \inf_{\alpha\in\mathbb{R}} \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha) \leq \mathcal{C}_{\phi_\varepsilon}(x, \frac{1}{2}, f_n) - \mathcal{C}_{\phi_\varepsilon}^\star(x, \frac{1}{2}) \leq \frac{1}{n}$$

Then we deduce that $(\alpha_n)_n$ is a minimizing sequence for $u \mapsto \frac{1}{2}\phi(u) + \frac{1}{2}\phi(-u)$ and similarly $(\beta_n)_n$ is also a minimizing sequence for $u \mapsto \frac{1}{2}\phi(u) + \frac{1}{2}\phi(-u)$. Now note that there always exist $\alpha, \beta \in \mathbb{R}$ such that, up to an extraction of a subsequence, we have $\alpha_n \xrightarrow[n\to\infty]{} \alpha$ and $\beta_n \xrightarrow[n\to\infty]{} \beta$. Furthermore by continuity of ϕ and since $0 \notin \operatorname{argmin} \phi(u) + \phi(-u)$, $\alpha \neq 0$ and $\beta \neq 0$. Without loss of generality one can assume that $\alpha < 0 < \beta$, then for n sufficiently large, $\alpha_n < 0 < \beta_n$. Moreover have

$$\begin{aligned} 0 &\leq \frac{1}{2}\max\left(\phi(\alpha_n), \phi(\beta_n)\right) + \frac{1}{2}\max\left(\phi(-\alpha_n), \phi(-\beta_n)\right) - \mathcal{C}_{\phi_\varepsilon}^\star(x, \frac{1}{2}) \\ &\leq \mathcal{C}_{\phi_\varepsilon}(x, \frac{1}{2}, f_n) - \mathcal{C}_{\phi_\varepsilon}^\star(x, \frac{1}{2}) \leq \frac{1}{n} \end{aligned}$$

so that we deduce:

$$\frac{1}{2}\max\left(\phi(\alpha_n), \phi(\beta_n)\right) + \frac{1}{2}\max\left(\phi(-\alpha_n), \phi(-\beta_n)\right) \longrightarrow \inf_\alpha \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha) \tag{5.1}$$

Since, for n sufficiently large, $\alpha_n < 0 < \beta_n$ and ϕ is strictly decreasing, $\max(\phi(\alpha_n), \phi(\beta_n)) = \phi(\alpha_n)$ and $\max(\phi(-\alpha_n), \phi(-\beta_n)) = \phi(-\beta_n)$. Moreover, there exists $\lambda > 0$ such that for n sufficiently large $\phi(\alpha_n) - \phi(\beta_n) \geq \lambda$. Then for n sufficiently large:

$$\begin{aligned}\frac{1}{2}\max(\phi(\alpha_n), \phi(\beta_n)) + \frac{1}{2}\max(\phi(-\alpha_n), \phi(-\beta_n)) &= \frac{1}{2}\phi(\alpha_n) + \frac{1}{2}\phi(-\beta_n) \\ &= \frac{1}{2}(\phi(\alpha_n) - \phi(\beta_n)) + \frac{1}{2}\phi(-\beta_n) + \frac{1}{2} + \phi(\beta_n) \\ &\geq \frac{1}{2}\lambda + \inf_u \frac{1}{2}\phi(u) + \frac{1}{2}\phi(-u)\end{aligned}$$

which lead to a contradiction with Equation 5.1. Then there exists a non zero integer n_0 such that for all $f \in \mathcal{F}(\mathcal{X})$, $x \in \mathcal{X}$

$$\mathcal{C}_{\phi_\varepsilon}(x, \frac{1}{2}, f) - \mathcal{C}^\star_{\phi_\varepsilon}(x, \frac{1}{2}) \leq \frac{1}{n_0} \implies \forall u, v \in B_\varepsilon(x),\ f(u) \times f(v) > 0.$$

The rightend term is equivalent to: for all $u \in B_\varepsilon(x)$, $f(u) > 0$ or for all $u \in B_\varepsilon(x)$, $f(u) < 0$. Then $\mathcal{C}_\varepsilon(x, \eta, f) = \frac{1}{2}$ and then $\mathcal{C}_\varepsilon(x, \eta, f) = \mathcal{C}^\star_\varepsilon(x, \eta)$

Putting all that together, for all $x \in \mathcal{X}$, $\eta \in [0, 1]$, $f \in \mathcal{F}(\mathcal{X})$:

$$\mathcal{C}_{\phi_\varepsilon}(x, \eta, f) - \mathcal{C}^\star_{\phi_\varepsilon}(x, \eta) \leq \min(\delta, \frac{1}{n_0}) \implies \mathcal{C}_\varepsilon(x, \eta, f) - \mathcal{C}^\star_\varepsilon(x, \eta) \leq \epsilon.$$

Then ϕ is adversarially uniformly calibrated at level ε □

Remark 6 (Strictly decreasing hypothesis)**.** *For the reciprocal, the additional assumption of strict decreasing of ϕ in a neighbourhood of 0 is not restrictive for losses. In Theorem 8, this assumption is stated as a necessary and sufficient condition for convex losses to be calibrated.*

5.1.2 Negative results

Thanks to Theorem 11, we can present two notable corollaries dismissing two important classes of surrogate losses in the standard setting. The first class of losses are convex margin losses. This losses are maybe the most widely used in modern day machine learning as they comprise the logistic loss or the margin loss that are the building block of most classification algorithms.

Corollary 1. *Let $\varepsilon > 0$. Then no convex margin loss can be adversarially calibrated at level ϵ.*

Given Theorem 11, this result is trivial : a convex loss satisfies $\frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha) \geq \phi(0)$, hence $0 \in \operatorname{argmin}_{\alpha \in \mathbb{R}} \phi(\alpha) + \phi(-\alpha)$. Then, ϕ is not adversarially calibrated at level ε. This result seems counter-intuitive and highlights the difficulty of optimizing and understanding the adversarial risk. Since convex losses are not calibrated, one may hope to rely on famous non convex losses such as that sigmoid and ramp losses. But, unfortunately, such losses are not neither calibrated.

Corollary 2. *Let $\varepsilon > 0$. Let $\lambda \in \mathbb{R}$ and ψ be a lower-bounded odd function such that for all $\alpha \in \mathbb{R}$, $\psi > -\lambda$. We define ψ as $\phi(\alpha) = \lambda + \psi(\alpha)$. Then ϕ is not adversarially calibrated at level ε.*

Indeed, $\frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha) = \lambda$, so that $\operatorname{argmin}_{\alpha \in \mathbb{R}} \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha) = \mathbb{R}$. Thanks to Theorem 11, ϕ is not adversarially calibrated at level ε.

5.1.3 Positive results

Theorem 12 also gives sufficient conditions for ϕ to be adversarially calibrated. Inspired by this result, we devise a class of margin losses that are indeed calibrated in the adversarial settings. We call this class shifted odd losses and we define it as follows.

Definition 20 (Shifted odd losses)**.** *We say that ϕ is a* shifted odd margin loss *if there exists $\lambda \geq 0$, $\tau > 0$, and a continuous lower bounded strictly decreasing odd function ψ in a neighbourhood of 0 such that for all $\alpha \in \mathbb{R}$, $\psi(\alpha) \geq -\lambda$ and $\phi(\alpha) = \lambda + \psi(\alpha - \tau)$.*

The key difference between a standard odd margin losses and a shifted odd margin losses is the variations of the function $\alpha \mapsto \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha)$. The primary difference is that, in the standard case the optima of this function is located in 0 while they are located in $-\infty$ and $+\infty$ in the adversarial setting. Let us give some examples of margin Shifted odd losses below.

Example (Shifted odd losses)**.** *For every $\varepsilon > 0$ and every $\tau > 0$, the shifted logistic loss, defined as follows, is adversarially calibrated at level ε: $\phi : \alpha \mapsto (1 + \exp(\alpha - \tau))^{-1}$ This loss is plotted on left in Figure 5.1. We also plotted on right in Figure 5.1 $\alpha \mapsto \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha)$ to justify that $0 \notin \operatorname{argmin}_{\alpha \in \mathbb{R}} \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha)$. Also note that the shifted ramp loss also satisfy the same properties.*

A consequence of Theorem 12 is that shifted odd losses are adversarially calibrated, as demonstrated in Proposition 12 stated below.

Proposition 12. *Let ϕ be a shifted odd margin loss. For every $\varepsilon > 0$, ϕ is adversarially calibrated at level ε.*

Proof. Let $\lambda > 0$, $\tau > 0$ and ϕ be a strictly decreasing odd function such that $\tilde{\phi}$ defined as $\tilde{\phi}(\alpha) = \lambda + \phi(\alpha - \tau)$ is non-negative.

Proving that $0 \notin \operatorname{argmin}_{\alpha \in \mathbb{R}} \frac{1}{2}\tilde{\phi}(t) + \frac{1}{2}\tilde{\phi}(-t)$. ϕ is clearly strictly decreasing and non negative then it admits a limit $l := -\lim_{t \to +\infty} \tilde{\phi}(t) \geq 0$. Then we have:

$$\lim_{t \to +\infty} \tilde{\phi}(t) = \lambda + l \quad \text{and} \quad \lim_{t \to -\infty} \tilde{\phi}(t) = \lambda - l$$

Consequently we have:

$$\lim_{t \to \infty} \frac{1}{2}\tilde{\phi}(t) + \frac{1}{2}\tilde{\phi}(-t) = \lambda$$

On the other side $\tilde{\phi}(0) = \lambda + \phi(-\tau) > \lambda + \phi(0) = \lambda$ since $\tau > 0$ and ϕ is strictly decreasing. Then $0 \notin \operatorname{argmin} \frac{1}{2}\tilde{\phi}(t) + \frac{1}{2}\tilde{\phi}(-t)$.

Proving that $\tilde{\phi}$ is calibrated for standard classification Let $\epsilon > 0$, $\eta \in [0, 1]$, $x \in \mathcal{X}$. If $\eta = \frac{1}{2}$, it is clear that for all $f \in \mathcal{F}(\mathcal{X})$, $\mathcal{C}(x, \frac{1}{2}, f) = \mathcal{C}^\star(x, \frac{1}{2}) = \frac{1}{2}$. Let us now assume that $\eta \neq \frac{1}{2}$, we have for all $f \in \mathcal{F}(\mathcal{X})$:

$$\begin{aligned}
\mathcal{C}_{\tilde{\phi}}(x, \eta, f) &= \lambda + \eta\phi(f(x) - \tau) + (1 - \eta)\phi(-f(x) - \tau) \\
&= \lambda + (\eta - \frac{1}{2})\left(\phi(f(x) - \tau) - \phi(-f(x) - \tau)\right) + \frac{1}{2}\left(\phi(f(x) - \tau) + \phi(-f(x) - \tau)\right)
\end{aligned}$$

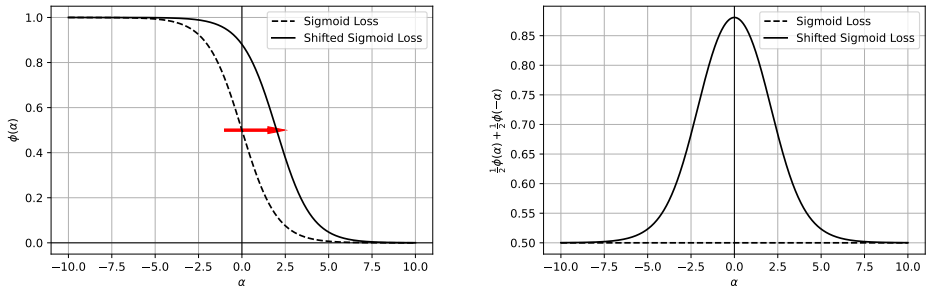


Figure 5.1: Illustration of the a calibrated loss in the adversarial setting. The sigmoid loss satisfy the hypothesis for ψ. Its shifted version is then calibrated for adversarial classification.

Let us show that $\operatorname{argmin}_{t\in\bar{\mathbb{R}}} \frac{1}{2}\tilde{\phi}(t) + \frac{1}{2}\tilde{\phi}(-t) = \{-\infty, +\infty\}$. We have for all t:

$$\begin{aligned}\frac{1}{2}\tilde{\phi}(t) + \frac{1}{2}\tilde{\phi}(-t) &= \lambda + \frac{1}{2}\left(\phi(t-\tau) + \phi(-t-\tau)\right) \\ &= \lambda + \frac{1}{2}\left(\phi(t-\tau) - \phi(t+\tau)\right) > \lambda\end{aligned}$$

since $t+\tau < t-\tau$ and ϕ is strictly decreasing. Hence by continuity of ϕ the optimum are attained when $t \to \infty$ or $t \to -\infty$. Then $\operatorname{argmin}_{t\in\bar{\mathbb{R}}} \frac{1}{2}\tilde{\phi}(t) + \frac{1}{2}\tilde{\phi}(-t) = \{-\infty, +\infty\}$.

Without loss of generality, let $\eta > 1/2$. $t \mapsto (\eta - \frac{1}{2})\left(\phi(t-\tau) - \phi(-t-\tau)\right)$ is strictly decreasing and $\operatorname{argmin}_{t\in\bar{\mathbb{R}}} \frac{1}{2}\left(\phi(t-\tau) + \phi(-t-\tau)\right) = \{-\infty, +\infty\}$, then we have $\operatorname{argmin}_{t\in\bar{\mathbb{R}}} \lambda + (\eta - \frac{1}{2})\left(t-\tau\right) - \phi(-t-\tau)) + \frac{1}{2}\left(\phi(t-\tau) + \phi(-t-\tau)\right) = \{+\infty\}$. By continuity of ϕ, we deduce that for $\delta > 0$ sufficiently small:

$$\mathcal{C}_{\tilde{\phi}}(x,\eta,f) - \mathcal{C}^{\star}_{\tilde{\phi}}(x,\eta) \leq \delta \implies f(x) > 0$$

The same reasoning holds for $\eta < \frac{1}{2}$. Then we deduce that $\tilde{\phi}$ is calibrated for standard classification.

Finally we get that that $\tilde{\phi}$ is calibrated for adversarial classification for every $\varepsilon > 0$. □

5.1.4 About $\mathcal{H}$-calibration

Our results naturally extends to $\mathcal{H}$-calibration. With mild assumptions on $\mathcal{H}$, it is possible to recover all the results made on calibration on $\mathcal{F}(\mathcal{X})$. First, it worth noting that, if $\mathcal{H}$ contains all constant functions, then the notion of $\mathcal{H}$-calibration and uniform $\mathcal{H}$-calibration are equivalent in the standard setting.

Proposition. *Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X})$. Let us assume that $\mathcal{H}$ contains all constant functions. Let ϕ be a continuous classification margin loss. ϕ is uniformly $\mathcal{H}$-calibrated for standard classification if and only if ϕ is uniformly calibrated for standard classification. It also holds for non-uniform calibration.*

Proof. Let us assume that ϕ is a continuous classification margin loss and that ϕ is uniformly calibrated. Let $\epsilon > 0$. There exists $\delta > 0$ such that, for all $\eta \in [0,1]$, $x \in \mathcal{X}$ and $f \in \mathcal{F}(\mathcal{X})$:

$$\mathcal{C}_{\phi}(x,\eta,f) - \mathcal{C}^{\star}_{\phi}(x,\eta) \leq \delta \implies \mathcal{C}(x,\eta,f) - \mathcal{C}^{\star}(x,\eta) \leq \epsilon \quad .$$

Let $\eta \in [0,1]$, $x \in \mathcal{X}$ and $f \in \mathcal{H}$ such that $\mathcal{C}_\phi(x,\eta,f) - \mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) \leq \delta$. Thanks to the previous proposition, $\mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) = \mathcal{C}^\star_\phi(x,\eta)$, and $f \in \mathcal{F}(\mathcal{X})$, then $\mathcal{C}_\phi(x,\eta,f) - \mathcal{C}^\star_\phi(x,\eta) \leq \delta$ and then:

$$\mathcal{C}(x,\eta,f) - \mathcal{C}^\star_{\mathcal{H}}(x,\eta) = \mathcal{C}(x,\eta,f) - \mathcal{C}^\star(x,\eta) \leq \epsilon$$

Then ϕ is uniformly $\mathcal{H}$-calibrated in standard classification.

Reciprocally, let us assume that ϕ is a continuous classification margin loss and that ϕ is uniformly $\mathcal{H}$-calibrated. Let $\epsilon > 0$. There exists $\delta > 0$ such that, for all $\eta \in [0,1]$, $x \in \mathcal{X}$ and $f \in \mathcal{H}$:

$$\mathcal{C}_\phi(x,\eta,f) - \mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) \leq \delta \implies \mathcal{C}(x,\eta,f) - \mathcal{C}^\star_{\mathcal{H}}(x,\eta) \leq \epsilon \quad .$$

Let $\eta \in [0,1]$, $x \in \mathcal{X}$ and $f \in \mathcal{H}$ such that $\mathcal{C}_\phi(x,\eta,f) - \mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) \leq \delta$. $\mathcal{C}_\phi(x,\eta,f) = \eta\phi(f(x)) + (1-\eta)\phi(-f(x))$. Let $\tilde{f} : u \mapsto f(x)$ for all $u \in \mathcal{X}$, then $\tilde{f} \in \mathcal{H}$ since $\tilde{f}$ is constant, $\mathcal{C}_\phi(x,\eta,f) = \mathcal{C}_\phi(x,\eta,\tilde{f})$ and $\mathcal{C}(x,\eta,f) = \mathcal{C}(x,\eta,\tilde{f})$. Thanks to the previous proposition, $\mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) = \mathcal{C}^\star_\phi(x,\eta)$. Then: $\mathcal{C}_\phi(x,\eta,\tilde{f}) - \mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) \leq \delta$ and then:

$$\mathcal{C}(x,\eta,f) - \mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) = \mathcal{C}(x,\eta,\tilde{f}) - \mathcal{C}^\star_\phi(x,\eta) \leq \epsilon$$

Then ϕ is uniformly calibrated in standard classification. □

Proposition 2 also naturally extends naturally to $\mathcal{H}$-calibration as long as $\mathcal{H}$ contains all constant functions.

Proposition. *Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X})$. Let us assume that $\mathcal{H}$ contains all constant functions. Let $\varepsilon > 0$ and ϕ be a continuous classification margin loss. For all $x \in \mathcal{X}$ and $\eta \in [0,1]$, we have*

$$\mathcal{C}^\star_{\phi_\varepsilon,\mathcal{H}}(x,\eta) = \mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) = \inf_{\alpha\in\mathbb{R}} \eta\phi(\alpha) + (1-\eta)\phi(-\alpha) = \mathcal{C}^\star_{\phi_\varepsilon}(x,\eta) = \mathcal{C}^\star_\phi(x,\eta) \quad .$$

The last equality also holds for the adversarial $0/1$ loss.

The proof is exactly the same that Proposition 2 since we used a constant function to prove the equality. We then get the necessary and sufficient conditions as follows.

Proposition (Necessary conditions for $\mathcal{H}$-Calibration of adversarial losses)**.** *Let $\varepsilon > 0$. Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X})$. Let us assume that $\mathcal{H}$ contains all constant functions and that there exists $x \in \mathcal{X}$ and $(f_n)_n \in \mathcal{H}^{\mathbb{N}}$ such that $f_n(u) \to 0$ for all $u \in B_\varepsilon(x)$ and for all $n \in \mathbb{N}$, $\sup_{u\in B_\varepsilon(x)} f_n(u) > 0$ and $\inf_{u\in B_\varepsilon(x)} f_n(u) < 0$ Let ϕ be a continuous margin loss . If ϕ is adversarially uniformly $\mathcal{H}$-calibrated at level ε, then ϕ is uniformly calibrated in the standard classification setting and $0 \notin \operatorname{argmin}_{\alpha\in\bar{\mathbb{R}}} \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha)$.*

TODO

Proposition (Sufficient conditions for $\mathcal{H}$-Calibration of adversarial losses)**.** *Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X})$. Let us assume that $\mathcal{H}$ contains all constant functions. Let ϕ be a continuous strictly decreasing margin loss and $\varepsilon > 0$. If ϕ is calibrated in the standard classification setting and $0 \notin \operatorname{argmin}_{\alpha\in\bar{\mathbb{R}}} \frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha)$, then ϕ is adversarially uniformly $\mathcal{H}$-calibrated at level ε.*

TODO

5.2 Towards Adversarial Consistency

In this section, we focus our study on the problem of adversarial consistency. In a first part, taking inspiration from Awasthi et al. [2021a], Long and Servedio [2013], we study the ε-realisable case, i.e. the case where the adversarial risk at level ε equals zero. In a second part, we analyze the behaviour of a candidate class of losses.

5.2.1 The Realisable Case

The feasible setting is an important case where there are no possible adversaries for the Bayes optimal classifier. Formally, this means that the risk of adversity is 0, as shown in the following definition.

Definition 21 (ε-realisability)**.** *Let $\mathbb{P}$ be a Borel probability distribution on $\mathcal{X} \times \mathcal{Y}$ and $\varepsilon \geq 0$. We say that $\mathbb{P}$ is* ε-realisable *if $\mathcal{R}^{\star}_{\varepsilon,\mathbb{P}} = 0$.*

In the case of realisable probability distribution, calibrated (and consequently consistent) margin losses in the standard classification setting are also calibrated and consistent in the adversarial case.

Proposition 13. *Let $\varepsilon > 0$. Let $\mathbb{P}$ be an ε-realisable distribution and ϕ be a calibrated margin loss in the standard setting. Then ϕ is adversarially consistent at level ε.*

The intuition behind this result is that if a probability distribution is ε-realisable, the marginal distributions are sufficiently separated so that there are no possible adversarial attacks, each point in the ϵ-neighbourhood of the support of the distribution can be classified independently of each other. To formally prove this result, we need a preliminary lemma.

Lemma 5. *Let $\mathbb{P}$ be an ε-realisable distribution and ϕ be a calibrated margin loss in the standard setting. Then $\mathcal{R}^{\star}_{\phi_\varepsilon,\mathbb{P}} = \inf_{\alpha\in\mathbb{R}} \phi(\alpha)$.*

Proof. Let $a \in \mathbb{R}$ be such that $\phi(a) - \inf_{\alpha\in\mathbb{R}} \phi(\alpha) \leq \epsilon$. $\mathbb{P}$ being ε-realisable, there exists a measurable function f such that:

$$\begin{aligned}\mathcal{R}_{\varepsilon,\mathbb{P}}(f) = \mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\varepsilon(x)} \mathbf{1}_{y\operatorname{sign}(f(x))\leq 0}\right] &= \mathbb{P}\left[\exists x' \in B_\varepsilon(x), \operatorname{sign}(f(x')) \neq y\right] \\ &\leq \epsilon' := \frac{\epsilon}{\max(1, \phi(-a))}.\end{aligned}$$

Denoting $p = \mathbb{P}(y = 1)$, $\mathbb{P}_1 = \mathbb{P}[\cdot|y = 1]$ and $\mathbb{P}_{-1} = \mathbb{P}[\cdot|y = -1]$, we have:

$$p \times \mathbb{P}_1\left[\exists x' \in B_\varepsilon(x), f(x') < 0\right] \leq \epsilon' \text{ and } (1-p) \times \mathbb{P}_{-1}\left[\exists x' \in B_\varepsilon(x), f(x') \geq 0\right] \leq \epsilon'.$$

Let us now define g as:

$$g(x) = \begin{cases} a \text{ if } f(x) \geq 0 \\ -a \text{ if } f(x) < 0 \end{cases}$$

We have:

$$\begin{aligned}\mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(g) &= \mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\varepsilon(x)} \phi(yg(x))\right] \\ &= p \times \mathbb{E}_{\mathbb{P}_1}\left[\sup_{x'\in B_\varepsilon(x)} \phi(g(x))\right] + (1-p) \times \mathbb{E}_{\mathbb{P}_{-1}}\left[\sup_{x'\in B_\varepsilon(x)} \phi(-g(x))\right]\end{aligned}$$

We have:

$$\begin{aligned}
p \times \mathbb{E}_{\mathbb{P}_1}\left[\sup_{x' \in B_\varepsilon(x)} \phi(g(x))\right] &\leq p \times \mathbb{E}_{\mathbb{P}_1}\left[\sup_{x' \in B_\varepsilon(x)} \phi(g(x))\mathbf{1}_{f(x')<0}\right] + p \times \mathbb{E}_{\mathbb{P}_1}\left[\sup_{x' \in B_\varepsilon(x)} \phi(g(x))\mathbf{1}_{f(x')\geq 0}\right] \\
&= \phi(-a) \times p \times \mathbb{P}_1\left[\exists x' \in B_\varepsilon(x), f(x') < 0\right] \\
&+ \phi(a) \times p \times (1 - \mathbb{P}_1\left[\exists x' \in B_\varepsilon(x), f(x') < 0\right]) \\
&\leq \phi(-a)\epsilon' + p \times \phi(a) \\
&\leq p \times \inf_{\alpha \in \mathbb{R}} \phi(\alpha) + 2\epsilon
\end{aligned}$$

Similarly we get that:

$$(1-p) \times \mathbb{E}_{\mathbb{P}_{-1}}\left[\sup_{x' \in B_\varepsilon(x)} \phi(-g(x))\right] \leq (1-p) \times \inf_{\alpha \in \mathbb{R}} \phi(\alpha) + 2\epsilon$$

We get: $\mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(g) \leq \inf_{\alpha\in\mathbb{R}} \phi(\alpha) + 4\epsilon$ and, hence $\mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} = \inf_{\alpha\in\mathbb{R}} \phi(\alpha)$. □

We are now set to prove the result of consistency in the realisable case.

Proof. Let $0 < \epsilon < 1$. Thanks to Theorem 8, ϕ is uniformly calibrated for standard classification, then, there exists $\delta > 0$ such that for all $f \in \mathcal{F}(\mathcal{X})$ and for all x:

$$\phi(yf(x)) - \inf_{\alpha\in\mathbb{R}} \phi(\alpha) \leq \delta \implies \mathbf{1}_{y\text{sign}f(x)\leq 0} = 0$$

Let now $f \in \mathcal{F}(\mathcal{X})$ be such that $\mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(f) \leq \mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} + \delta\epsilon$.Thanks to Lemma 5, we have:

$$\mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(f) - \mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} = \mathbb{E}_{\mathbb{P}}\left[\sup_{x' \in B_\varepsilon(x)} \phi(yf(x)) - \inf_{\alpha\in\mathbb{R}} \phi(\alpha)\right] \leq \delta\epsilon$$

Then by Markov inequality:

$$\mathbb{P}\left[\sup_{x' \in B_\varepsilon(x)} \phi(yf(x)) - \inf \phi \geq \delta\right] \leq \frac{\mathbb{E}_{\mathbb{P}}\left[\sup_{x' \in B_\varepsilon(x)} \phi(yf(x)) - \inf \phi\right]}{\delta} \leq \epsilon$$

So we have $\mathbb{P}\left[\forall x' \in B_\varepsilon(x), \phi(yf(x)) - \inf \phi \leq \delta\right] \geq 1-\epsilon$ and then $\mathbb{P}\left[\forall x' \in B_\varepsilon(x), \mathbf{1}_{y\text{sign}f(x)\leq 0} = 0\right] \geq 1 - \epsilon$. Since $\mathbb{P}$ is ε-realisable, we have $\mathcal{R}^\star_{\varepsilon,\mathbb{P}} = 0$ and:

$$\mathcal{R}_{\varepsilon,\mathbb{P}}(f) - \mathcal{R}^\star_{\varepsilon,\mathbb{P}} = \mathcal{R}_{\varepsilon,\mathbb{P}}(f) = \mathbb{P}\left[\exists x' \in B_\varepsilon(x), \text{sign}(f(x')) \neq y\right] \leq \epsilon$$

which concludes the proof. □

5.2.2 Towards the General Case

In this section, we seek to pave the way towards proving the consistency of shifted odd losses. We will observe that their behavior is actually very similar to that of the 0/1 loss, which makes them good candidates to be consistent losses. To this end, we first add an extra hypothesis to the odd shifted losses in order to simplify our technical analysis.

Definition 22 (0/1-like margin losses)**.** *ϕ is a* 0/1-like margin loss *if there exists $\lambda \geq 0$, $\tau \geq 0$, and a continuous lower bounded strictly decreasing odd function ψ in a neighbourhood of 0 such that for all $\alpha \in \mathbb{R}$, $\psi(\alpha) \geq -\lambda$ and $\phi(\alpha) = \lambda + \psi(\alpha - \tau)$ and*

$$\lim_{t \to -\infty} \phi(t) = 1 \text{ and } \lim_{t \to +\infty} \phi(t) = 0$$

Note here that the losses here are not necessarily shifted, making this condition weaker. Consequently, we cannot hope that such losses are consistent neither calibrated, but they might help in paving a way towards consistency. Note also that if ϕ is a odd or shifted odd loss, one can always find a rescaling of ϕ such that ϕ becomes a 0/1-like margin loss. Note also that such a rescaling does neither change the notion of consistency and calibration for ϕ nor for its rescaled version.

Based on min-max arguments, we provide below some results better characterizing0/1-like margin loss functions in the adversarial setting. Let us first recall the notions of *midpoint property* and *adversarial distributions set* that will be useful from now on as well as an important existing result from Pydi and Jog [2021b].

Definition 23. *Let $(\mathcal{X}, d)$ be a proper Polish metric space. We say that $\mathcal{X}$ satisfy the* midpoint property *if for all $x_1, x_2 \in \mathcal{X}$ there exist $x \in \mathcal{X}$ such that $d(x, x_1) = d(x, x_2) = \frac{d(x_1, x_2)}{2}$.*

We recall also the set $\mathcal{A}_\varepsilon(\mathbb{P})$ of adversarial distributions introduced in Chapter 4.

Definition 24. *Let $\mathbb{P}$ be a Borel probability distribution and $\varepsilon > 0$. We define the set of* adversarial distributions *$\mathcal{A}_\varepsilon(\mathbb{P})$ as:*

$$\begin{aligned}\mathcal{A}_\varepsilon(\mathbb{P}) := \big\{ &\mathbb{Q} \in \mathcal{M}_1^+(\mathcal{X} \times \mathcal{Y}) \mid \exists \gamma \in \mathcal{M}_1^+\left((\mathcal{X} \times \mathcal{Y})^2\right), \\ &d(x, x') \leq \varepsilon,\ y = y' \ \ \gamma\text{-a.s.},\ \Pi_{1\sharp}\gamma = \mathbb{P},\ \Pi_{2\sharp}\gamma = \mathbb{Q}\big\}\end{aligned}$$

Theorem 13 (Pydi and Jog [2021b])**.** *Let $\mathcal{X}$ be a Polish space satisfying the midpoint property. Then strong duality holds:*

$$\mathcal{R}^\star_\varepsilon(\mathbb{P}) = \inf_{f \in \mathcal{F}(\mathcal{X})} \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}_\mathbb{Q}(f) = \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_\mathbb{Q}(f)$$

Moreover the supremum of the right-end term is attained.

Note that in the original version of the theorem, Pydi and Jog [2021b] did not prove that the supremum is attained. We add the proof for this property in Appendix xxx.

Connections between 0/1-like margin loss and 0/1 loss: a min-max viewpoint. Thanks the the above concepts, we can now present some results identifying the similarity and the differences between the 0/1 loss and a 0/1-like margin losses. We first, show that for a given fixed probability distribution $\mathbb{P}$, the adversarial optimal risk associated with a 0/1-like margin loss and the 0/1 loss are equal.

Theorem 14. *Let $\mathcal{X}$ be a Polish space satisfying the midpoint property. Let $\varepsilon \geq 0$. Let $\mathbb{P}$ be a Borel probability distribution over $\mathcal{X} \times \mathcal{Y}$. Let ϕ be a 0/1-like margin loss. Then, we have:*

$$\mathcal{R}^\star_{\phi_\varepsilon, \mathbb{P}} = \mathcal{R}^\star_{\varepsilon, \mathbb{P}}$$

In particular, we note that this property holds true for the standard risk. To prove this result, we need the following lemma.

Lemma 6. *Let $\mathbb{Q}$ be a Borel probability distribution over $\mathcal{X} \times \mathcal{Y}$. Let ϕ be a 0/1-like shifted odd loss, then: $\mathcal{R}^\star_{\phi, \mathbb{Q}} = \mathcal{R}^\star_\mathbb{Q}$.*

Proof. Bartlett et al. [2006], Steinwart [2007] proved that: for every margin losses ϕ,

$$\begin{aligned}\mathcal{R}^\star_{\phi,\mathbb{Q}} = \inf_{f\in\mathcal{F}(X)} \mathbb{E}_{(x,y)\sim\mathbb{Q}}\left[\phi(yf(x))\right] &= \mathbb{E}_{x\sim\mathbb{Q}_x}\left[\inf_{\alpha\in\mathbb{R}} \mathbb{Q}(y=1|x)\phi(\alpha) + (1-\mathbb{Q}(y=-1|x))\phi(-\alpha)\right] \\ &= \mathbb{E}_{x\sim\mathbb{Q}_x}\left[\mathcal{C}^\star_\phi(\mathbb{Q}(y=1|x), x)\right]\end{aligned}$$

We also have $\mathcal{R}^\star_{\mathbb{Q}} = \mathbb{E}_{x\sim\mathbb{Q}_x}\left[\mathcal{C}^\star(\mathbb{Q}(y=1|x), x)\right]$. Moreover, if ϕ is a 0/1-like shifted odd loss, then: for every $x\in\mathcal{X}$ and $\eta\in[0,1]$, $\mathcal{C}^\star_\phi(\eta, x) = \min(\eta, 1-\eta) = \mathcal{C}^\star(\eta, x)$. We can then conclude that $\mathcal{R}^\star_{\phi,\mathbb{Q}} = \mathcal{R}^\star_{\mathbb{Q}}$. □

We are now set to prove Theorem 14.

Proof. Let $\epsilon > 0$ and $\mathbb{P}$ be a distribution. Let f such that $\mathcal{R}_{\varepsilon,\mathbb{P}}(f) \leq \mathcal{R}^\star_{\varepsilon,\mathbb{P}} + \epsilon$. Let $a > 0$ such that $\phi(a) \geq 1-\epsilon$ and $\phi(-a) \leq \epsilon$. We define g as:

$$g(x) = \begin{cases} a & \text{if } f(x) \geq 0 \\ -a & \text{if } f(x) < 0 \end{cases}$$

We have $\phi(yg(x)) = \phi(a)\mathbf{1}_{ysign(f(x))\leq 0} + \phi(-a)\mathbf{1}_{ysign(f(x))>0}$. Then

$$\begin{aligned}\mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(g) &= \mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\varepsilon(x)} \phi(yg(x))\right] \\ &= \mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\varepsilon(x)} \phi(a)\mathbf{1}_{ysign(f(x'))\leq 0} + \phi(-a)\mathbf{1}_{ysign(f(x'))>0}\right] \\ &\leq \mathbb{E}_{\mathbb{P}}\left[\sup_{x'\in B_\varepsilon(x)} \mathbf{1}_{ysign(f(x'))\leq 0}\right] + \phi(-a) \\ &\leq \mathcal{R}^\star_{\varepsilon,\mathbb{P}} + 2\epsilon \quad .\end{aligned}$$

Then we have $\mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} \leq \mathcal{R}^\star_{\varepsilon,\mathbb{P}}$. On the other side, we have:

$$\begin{aligned}\mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} &\geq \sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \inf_{f\in\mathcal{F}(\mathcal{X})} \mathcal{R}_{\phi,\mathbb{Q}}(f) = \sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}^\star_{\phi,\mathbb{Q}} \\ &= \sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}^\star_{\mathbb{Q}} = \sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \inf_{f\in\mathcal{F}(\mathcal{X})} \mathcal{R}_{\mathbb{Q}}(f) \\ &= \inf_{f\in\mathcal{F}(\mathcal{X})} \sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}_{\mathbb{Q}}(f) = \mathcal{R}^\star_{\varepsilon,\mathbb{P}}\end{aligned}$$

Then finally we get that $\mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} = \mathcal{R}^\star_{\varepsilon,\mathbb{P}}$.

□

From this result, we can derive two interesting corollaries about 0/1-like margin losses. First, strong duality holds for the risk associated with ϕ.

Corollary 3 (Strong duality for ϕ)**.** *Let assume that $\mathcal{X}$ be a Polish space satisfying the midpoint property. Let $\varepsilon \geq 0$. Let $\mathbb{P}$ be a Borel probability distribution over $\mathcal{X} \times \mathcal{Y}$. Let ϕ be a $0/1$-like margin loss. Then, we have:*

$$\inf_{f \in \mathcal{F}(\mathcal{X})} \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}_{\phi,\mathbb{Q}}(f) = \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{\phi,\mathbb{Q}}(f)$$

Moreover the supremum is attained.

Note that there is no reason that the infimum is attained. A second interesting corollary is the equality of the set of optimal attacks, i.e. distributions of $\mathcal{A}_\varepsilon(\mathbb{P})$ that realizes maximizes the dual problem, for the same for the $0/1$ loss and $0/1$-like margin loss.

Corollary 4 (Optimal attacks)**.** *Let assume that $\mathcal{X}$ be a Polish space satisfying the midpoint property. Let $\varepsilon \geq 0$. Let $\mathbb{P}$ be a Borel probability distribution over $\mathcal{X} \times \mathcal{Y}$. Then, an optimal attack $\mathbb{Q}^\star$ of level ε exists for both the $0/1$ loss and ϕ. Moreover, for $\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})$. $\mathbb{Q}$ is an optimal attack for the loss ϕ if and only if it is an optimal attack for the $0/1$ loss.*

A step towards consistency. From the previous results, we are able to prove a first result toward teh demonstration of consistency. This result is much weaker than consistency result, but it guarrantees [...]

Proposition 14. *Let assume that $\mathcal{X}$ be a Polish space satisfying the midpoint property. Let $\varepsilon \geq 0$. Let $\mathbb{P}$ be a Borel probability distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathbb{Q}^\star$ be an optimal attack of level ε. Let $(f_n)_n$ be a sequence of $\mathcal{F}(\mathcal{X})$ such that $\mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(f_n) \to \mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}}$. Then $\mathcal{R}_{\mathbb{Q}^\star}(f_n) \to \mathcal{R}^\star_{\varepsilon,\mathbb{P}}$.*

We hope this result and its proof may lead to a full proof of consistency. This result is significantly weaker than consistency as stated in the following remark. In the proof of the previous results, we did not use the assumptions that losses are shifted. In our opinion, it is the key element that we miss and need to use to conclude the consistency of this family of losses. The shift in the loss would force the classifier to goes to $\pm\infty$ on the ε neighbourhood support of the distribution of $\mathbb{P}$. This question is complicated and is left as further work.

5.3 Discussions and Open Questions

In this chapter, we set some solid theoretical foundations for the study of adversarial consistency. We highlighted the importance of the definition of the $0/1$ loss, as well as the nuance between calibration and consistency that is specific to the adversarial setting. Furthermore, we solved the calibration problem, by giving a necessary and sufficient condition for decreasing, continuous margin losses to be adversarially calibrated. Since this is a necessary condition for consistency, an important consequence of this result is that no convex margin loss can be consistent. This rules out most of the commonly used surrogates, and spurs the need for new families of consistent, yet easily optimisable families of losses.

Consistency of $0/1$-like shifted margin losses. In Section 5.2.2, we introduced candidates losses for consistency. While these losses might lead to promising results, there is still a gap to prove the consistency of these losses. This question is left as further work. TO ADD STH

Necessary and sufficients conditions for consistency. While we provided necessary and sufficient conditions for calibration in the adversarial setting, it is a difficult and open question to solve the problem of consistency. One may ask if the conditions we found for calibration might be necessary or sufficient for consistency. While there is an intuition that the notion of

calibration is much weaker than consistency, we did not prove this. It would be challenging to find a counter-example for a loss that is calibrated but not consistent in the adversarial setting.

Chapter 6

A Dynamical System Perspective for Lipschitz Neural Networks

Contents

In this chapter, we study the design of Lipschitz Layers under the light of the dynamical system interpretation of Neural Networks. We recall briefly the continuous time interpretation. Let $(F_t)_{t\in[0,T]}$ be a family of functions on $\mathbb{R}^d$, we define the continuous time Residual Networks flow associated with F_t as:

$$\begin{cases} x_0 & = x \in \mathcal{X} \\ \frac{dx_t}{dt} & = F_t(x_t) \text{ for } \ t \in [0,T] \end{cases}$$

From this continuous and dynamical interpretation, we analyze the Lipschitzness property of Neural Networks. We then study the discretization schemes that can preserve the Lipschitzness properties. With this point of view, we can readily recover several previous methods that build 1-Lipschitz neural networks [Singla and Feizi, 2021, Trockman et al., 2021]. Therefore, the dynamical system perspective offers a general and flexible framework to build Lipschitz

Neural Networks facilitating the discovery of new approaches. In this vein, we introduce convex potentials in the design of the Residual Network flow and show that this choice of parametrization yields to by-design 1-Lipschitz neural networks. At the very core of our approach lies a new 1-Lipschitz non-linear operator that we call *Convex Potential Layer* which allows us to adapt convex potential flows to the discretized case. These blocks enjoy the desirable property of stabilizing the training of the neural network by controlling the gradient norm, hence overcoming the exploding gradient issue. We experimentally demonstrate our approach by training large-scale neural networks on several datasets, reaching state-of-the art results in terms of under-attack and certified accuracy.

6.1 A Framework to design Lipschitz Layers

The continuous time interpretation allows us to better investigate the robustness properties and assess how a difference of the initial values (the inputs) impacts the inference flow of the model. Let us consider two continuous flows x_t and z_t associated with F_t but differing in their respective initial values x_0 and z_0. Our goal is to characterize the time evolution of $\|x_t - z_t\|$ by studying its time derivative. We recall that every matrix $M \in \mathbb{R}^{d\times d}$ can be uniquely decomposed as the sum of a symmetric and skew-symmetric matrix $M = S(M) + A(M)$. By applying this decomposition to the Jacobian matrix $\nabla_x F_t(x)$ of F_t, we can show that the time derivative of $\|x_t - z_t\|^2$ only involves the symmetric part $S(\nabla_x F_t(x))$.

For two symmetric matrices $S_1, S_2 \in \mathbb{R}^{d\times d}$, we denote $S_1 \preceq S_2$ if, for all $x \in \mathbb{R}^d$, $\langle x, (S_2 - S_1)x\rangle \geq 0$. By focusing on the symmetric part of the Jacobian matrix we can show the following proposition.

Proposition 15. *Let $(F_t)_{t\in[0,T]}$ be a family of differentiable functions almost everywhere on $\mathbb{R}^d$. Let us assume that there exists two measurable functions $t \mapsto \mu_t$ and $t \mapsto \lambda_t$ such that*

$$\mu_t I \preceq S(\nabla_x F_t(x)) \preceq \lambda_t I$$

for all $x \in \mathbb{R}^d$, and $t \in [0, T]$. Then the flow associated with F_t satisfies for all initial conditions x_0 and z_0:

$$\|x_0 - z_0\| e^{\int_0^t \mu_s ds} \leq \|x_t - z_t\| \leq \|x_0 - z_0\| e^{\int_0^t \lambda_s ds}$$

Proof. Consider the time derivative of the square difference between the two flows x_t and z_t associated with the function F_t and following the definition 18:

$$\begin{aligned}
\frac{d}{dt}\|x_t - z_t\|_2^2 &= 2\big\langle x_t - z_t, \frac{d}{dt}(x_t - z_t)\big\rangle \\
&= 2\big\langle x_t - z_t, F_{\theta_t}(x_t) - F_{\theta_t}(z_t)\big\rangle \\
&= 2\big\langle x_t - z_t, \int_0^1 \nabla_x F_{\theta_t}(x_t + s(z_t - z_t))(x_t - z_t) ds\big\rangle, \text{ by Taylor-Lagrange formula} \\
&= 2\int_0^1 \big\langle x_t - z_t, \nabla_x F_{\theta_t}(x_t + s(z_t - z_t))(x_t - z_t)\big\rangle ds \\
&= 2\int_0^1 \big\langle x_t - z_t, S(\nabla_x F_{\theta_t}(x_t + s(z_t - z_t)))(x_t - z_t)\big\rangle ds
\end{aligned}$$

In the last step, we used that for every skew-symmetric matrix A and vector x, $\|x, Ax\| = 0$. Since $\mu_t I \preceq S(\nabla_x F_{\theta_t}(x_t + s(z_t - y_t))) \preceq \lambda_t I$, we get

$$2\mu_t \|x_t - z_t\|_2^2 \leq \frac{d}{dt}\|x_t - z_t\|_2^2 \leq 2\lambda_t \|x_t - z_t\|_2^2$$

Then by Gronwall Lemma, we have

$$\|x_0 - y_0\| e^{\int_0^t \mu_s ds} \leq \|x_t - y_t\| \leq \|x_0 - y_0\| e^{\int_0^t \lambda_s ds}$$

which concludes the proof. □

The symmetric part plays even a more important role since one can show that a twice differentiable function whose Jacobian is always skew-symmetric is actually linear. Indeed, let $F := (F_1, \dots, F_d) : \mathbb{R}^d \to \mathbb{R}^d$ be a twice differentiable function such that $\nabla F(x)$ is skew-symmetric for all $x \in \mathbb{R}^d$. Then we have for all i, j, k:

$$\partial_i \partial_j F_k = -\partial_i \partial_k F_j = -\partial_k \partial_i F_j = \partial_k \partial_j F_i = \partial_j \partial_k F_i = -\partial_j \partial_i F_k = -\partial_i \partial_j F_k$$

So we have $\partial_i \partial_j F_k = 0$ and then F is linear: there exists a skew-symmetric matrix A such that $F(x) = Ax$. Moreover, constraining $S(\nabla_x F_t(x))$ in the general case is technically difficult and a solution resorts to a more intuitive parametrization of F_t as the sum of two functions $F_{1,t}$ and $F_{2,t}$ whose Jacobian matrix are respectively symmetric and skew-symmetric. Thus, such a parametrization enforces $F_{2,t}$ to be linear and skew-symmetric. For the choice of $F_{1,t}$, we propose to rely on potential functions: a function $F_{1,t} : \mathbb{R}^d \to \mathbb{R}^d$ derives from a simpler family of scalar valued function in $\mathbb{R}^d$, called the *potential*, via the gradient operation. Moreover, since the Hessian of the potential is symmetric, the Jacobian for $F_{1,t}$ is then also symmetric. If we had the convex property to this potential, its Hessian has positive eigenvalues. Therefore we introduce the following corollary.

Corollary 3. *Let $(f_t)_{t\in[0,T]}$ be a family of convex differentiable functions on $\mathbb{R}^d$ and $(A_t)_{t\in[0,T]}$ a family of skew symmetric matrices. Let us define*

$$F_t(x) = -\nabla_x f_t(x) + A_t x,$$

then the flow associated with F_t satisfies for all initial conditions x_0 and z_0:

$$\|x_t - z_t\| \leq \|x_0 - z_0\|$$

Proof. For all t, x, we have $F_t(x) = -\nabla_x f_t(x) + A_t x \quad \text{so} \quad \nabla_x F_t(x) = -\nabla_x^2 f_t(x) + A_t$. Then $S(\nabla_x F_t(x)) = -\nabla_x^2 f_t(x)$. Since f is convex, we have $\nabla_x^2 f_t(x) \succeq 0$. So by application of Proposition 15, we deduce $\|x_t - y_t\| \leq \|x_0 - y_0\|$ for all trajectories starting from x_0 and y_0. □

This simple property suggests that if we could parameterize F_t with convex potentials, it would be less sensitive to input perturbations and therefore more robust to adversarial examples. We also remark that the skew symmetric part is then norm-preserving. However, the discretization of such flow is challenging in order to maintain this property of stability.

6.1.1 Discretized Flows

To study the discretization of the previous flow, let $t = 1, \dots, T$ be the discretized time steps and from now we consider the flow defined by $F_t(x) = -\nabla f_t(x) + A_t x$, with $(f_t)_{t=1,\dots,T}$ a family of convex differentiable functions on $\mathbb{R}^d$ and $(A_t)_{t=1,\dots,T}$ a family of skew symmetric matrices. The most basic method the explicit Euler scheme as defined by:

$$x_{t+1} = x_t + F_t(x_t)$$

However, if $A_t \neq 0$, this discretized system might not satisfy $\|x_t - z_t\| \leq \|x_0 - z_0\|$. Indeed, consider the simple example where $f_t = 0$. We then have:

$$\|x_{t+1} - z_{t+1}\|^2 - \|x_t - z_t\|^2 = \|A_t (x_t - z_t)\|^2.$$

Thus explicit Euler scheme cannot guarantee Lipschitzness when $A_t \neq 0$. To overcome this difficulty, the discretization step can be split in two parts, one for $\nabla_x f_t$ and one for A_t:

$$\begin{cases} x_{t+\frac{1}{2}} &= \text{STEP1}(x_t, \nabla_x f_t) \\ x_{t+1} &= \text{STEP2}(x_{t+\frac{1}{2}}, A_t) \end{cases}$$

This type of discretization scheme can be found for instance from Proximal Gradient methods where one step is explicit and the other is implicit. Then, we dissociate the Lipschitzness study of both terms of the flow.

6.1.2 Discretization scheme for $\nabla_x f_t$

To apply the explicit Euler scheme to $\nabla_x f_t$, an additional smoothness property on the potential functions is required to generalize the Lipschitzness guarantee to the discretized flows. Recall that a function f is said to be *L-smooth* if it is differentiable and if $x \mapsto \nabla_x f(x)$ is L-Lipschitz.

Proposition 16. *Let $t \in \{1, \cdots, T\}$ Let us assume that f_t is L_t-smooth. We define the following discretized ResNet gradient flow using h_t as a step size:*

$$x_{t+\frac{1}{2}} = x_t - h_t \nabla_x f_t(x_t)$$

Consider now two trajectories x_t and z_t with initial points $x_0 = x$ and $z_0 = z$ respectively, if $0 \leq h_t \leq \frac{2}{L_t}$, then

$$\|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}}\|_2 \leq \|x_t - z_t\|_2$$

Proof. With $c_t = \|x_t - z_t\|_2^2$, we can write:

$$c_{t+\frac{1}{2}} - c_t = -2h_t \big\langle x_t - z_t, \nabla_x F_{\theta_t}(x_t) - \nabla_x F_{\theta_t}(z_t) \big\rangle + h_t^2 \|\nabla_x F_{\theta_t}(z_t) - \nabla_x F_{\theta_t}(z_t)\|_2^2$$

This equality allows us to derive the equivalence between $c_{t+1} \leq c_t$ and:

$$\frac{h_t}{2} \|\nabla F_{\theta_t}(x_t) - \nabla F_{\theta_t}(z_t)\|_2^2 \leq \langle x_t - z_t, \nabla F_{\theta_t}(z_t) - \nabla F_{\theta_t}(z_t) \rangle$$

Moreover, assuming that F_{θ_t} being that:

$$\frac{1}{L_t} \|\nabla_x F_{\theta_t}(x_t) - \nabla_x F_{\theta_t}(z_t)\|_2^2 \leq \big\langle x_t - z_t, \nabla_x F_{\theta_t}(x_t) - \nabla_x F_{\theta_t}(z_t) \big\rangle$$

We can see with this last inequality that if we enforce $h_t \leq \frac{2}{L_t}$, we get $c_{t+\frac{1}{2}} \leq c_t$ which concludes the proof. □

In Section 6.2, we describe how to parametrize a neural network layer to implement such a discretization step by leveraging the recent work on Input Convex Neural Networks Amos et al. [2017].

Remark 7. *Another solution relies on the implicit Euler scheme: $x_{t+\frac{1}{2}} = x_t - \nabla_x f_t(x_{t+\frac{1}{2}})$. Let us remark that $x_{t+\frac{1}{2}}$ is uniquely defined as:*

$$x_{t+\frac{1}{2}} = \underset{x \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2} \|x - x_t\|^2 + f_t(x)$$

We recognized here the proximal operator of f_t that is uniquely defined since f_t is convex. Moreover we have for two trajectories x_t and z_t:

$$\begin{aligned}\|x_t - z_t\|_2^2 &= \|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}} + \nabla_x f_t(x_{t+\frac{1}{2}}) - \nabla_x f_t(z_{t+\frac{1}{2}})\|_2^2 \\ &= \|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}}\|^2 + 2\langle x_t - z_t, \nabla_x f_t(x_{t+\frac{1}{2}}) - \nabla_x f_t(z_{t+\frac{1}{2}})\rangle \\ &+ \|\nabla_x f_t(x_{t+\frac{1}{2}}) - \nabla_x f_t(z_{t+\frac{1}{2}})\|_2^2 \\ &\geq \|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}}\|^2\end{aligned}$$

where the last inequality is deduced from the convexity of f_t. So, without any further assumption on f_t, the discretized implicit convex potential flow is 1-Lipschitz. Then, this strategy defines a 1-Lipschitz flow without further assumption on f_t than convexity. To compute such a layer, one could solve the proximal operator strongly convex-minimization optimization problem. However, This strategy is not computationally efficient and not scalable and preliminary experiments did not show competitive results and the training time is prohibitive. We leave this solution for future work.

6.1.3 Discretization scheme for A_t

The second step of discretization involves the term with skew-symmetric matrix A_t. As mentioned earlier, the challenge is that the *explicit Euler discretization* is not contractive. More precisely, the following property

$$\|x_{t+1} - z_{t+1}\| \geq \|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}}\|$$

is satisfied with equality only in the special and useless case of $x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}} \in \ker(A_t)$. Moreover, the implicit Euler discretization induces an increasing norm and hence does not satisfy the desired property of norm preservation neither.

Midpoint Euler method. We thus propose to use *Midpoint Euler* method, defined as follows:

$$\begin{aligned}x_{t+1} &= x_{t+\frac{1}{2}} + A_t \frac{x_{t+1} + x_{t+\frac{1}{2}}}{2} \\ \iff x_{t+1} &= \left(I - \frac{A_t}{2}\right)^{-1} \left(I + \frac{A_t}{2}\right) x_{t+\frac{1}{2}}.\end{aligned}$$

Since A_t is skew-symmetric, $I - \frac{A_t}{2}$ is invertible. This update corresponds to the Cayley Transform of $\frac{A_t}{2}$ that induces an orthogonal mapping. This kind of layers was introduced and extensively studied in Trockman et al. [2021].

Exact Flow. One can define the simple differential equation corresponding to the flow associated with A_t

$$\frac{du_t}{ds} = A_t u_s, \quad u_0 = x_{t+\frac{1}{2}},$$

There exists an exact solution exists since A_t is linear. By taking the value at $s = \frac{1}{2}$, we obtained the following transformation:

$$x_{t+1} := u_{\frac{1}{2}} = e^{\frac{A}{2}} x_{t+\frac{1}{2}}.$$

This step is therefore clearly norm preserving but the matrix exponentiation is challenging and it requires efficient approximations. This trend was recently investigated under the name of Skew Orthogonal Convolution (SOC) Singla and Feizi [2021].

6.2 Parametrizing Convex Potentials Layers

As presented in the previous section, parametrizing the skew symmetric updates has been extensively studied by Singla and Feizi [2021], Trockman et al. [2021]. In this chapter, we focus on the parametrization of symmetric update with the convex potentials proposed in 16. For that purpose, the Input Convex Neural Network (ICNN) [Amos et al., 2017] provide a relevant starting point that we will extend.

6.2.1 Gradient of ICNN

We use 1-layer ICNN [Amos et al., 2017] to define an efficient computation of Convex Potentials Flows. For any vectors $w_1, \dots w_k \in \mathbb{R}^d$, and bias terms $b_1, \dots, b_k \in \mathbb{R}$, and for ϕ a convex function, the potential F defined as:

$$F_{w,b} : x \in \mathbb{R}^d \mapsto \sum_{i=1}^{k} \phi(w_i^\top x + b_i)$$

defines a convex function in x as the composition of a linear and a convex function. Its gradient with respect to its input x is then:

$$x \mapsto \sum_{i=1}^{k} w_i \phi'(w_i^\top x + b_i) = \mathbf{W}^\top \phi'(\mathbf{W}x + \mathbf{b})$$

with $\mathbf{W} \in \mathbb{R}^{k \times d}$ and $\mathbf{b} \in \mathbb{R}^k$ are respectively the matrix and vector obtained by the concatenation of, respectively, $w_i^\top$ and b_i, and ϕ' is applied element-wise. Moreover, assuming ϕ' is L-Lipschitz, we have that $F_{w,b}$ is $L\|\mathbf{W}\|_2^2$-smooth. $\|\mathbf{W}\|_2$ denotes the spectral norm of $\mathbf{W}$. The reciprocal also holds: if $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-decreasing L-Lipschitz function, $\mathbf{W} \in \mathbb{R}^{k \times d}$ and $b \in \mathbb{R}^k$, there exists a convex $L\|\mathbf{W}\|_2^2$-smooth function $F_{w,b}$ such that

$$\nabla_x F_{w,b}(x) = \mathbf{W}^\top \sigma(\mathbf{W}x + \mathbf{b}),$$

where σ is applied element-wise. The next section shows how this property can be used to implement the building block and training of such layers.

6.2.2 Convex Potential layers

From the previous section, we derive the following *Convex Potential Layer*:

$$z = x - \frac{2}{\|\mathbf{W}\|_2^2} \mathbf{W}^\top \sigma(\mathbf{W}x + b)$$

Written in a matrix form, this layer can be implemented with every linear operation $\mathbf{W}$. In the context of image classification, it is beneficial to use convolutions[1] instead of generic linear transforms represented by a dense matrix.

Remark 8. *When* $\mathbf{W} \in \mathbb{R}^{1 \times d}$, $b = 0$ *and* $\sigma = \text{ReLU}$, *the* Convex Potential Layer *is equivalent to the HouseHolder activation function introduced in Singla et al. [2021a].*

Residual Networks [He et al., 2016] are also composed of other types of layers which increase or decrease the dimensionality of the flow. Typically, in a classical setting, the number of input channels is gradually increased, while the size of the image is reduced with pooling layers. In order to build a 1-Lipschitz Residual Network, all operations need to be properly scale or normalize in order to maintain the Lipschitz constant.

[1] For instance, one can leverage the `Conv2D` and `Conv2D_transpose` functions of the PyTorch framework [Paszke et al., 2019]

Algorithm 5: Computation of a Convex Potential Layer

Require: **Input:** x**, vector:** u**, weights:** $\mathbf{W}$**,** b
Ensure: Compute the layer z and return u
$v \leftarrow \mathbf{W}u / \|\mathbf{W}u\|_2$
$u \leftarrow \mathbf{W}^\top v / \|\mathbf{W}^\top v\|_2$
$h \leftarrow 2 / \left(\sum_i (\mathbf{W}u \cdot v)_i\right)^2$
} 1 iter. for training, 100 iter. for inference
return $x - h\left[\mathbf{W}^\top \sigma(\mathbf{W}x + b)\right], u$

Increasing dimensionsionality. To increase the number of channels in a convolutional Convex Potential Layer, a zero-padding operation can be easily performed: an input x of size $c \times h \times w$ can be extended to some x' of size $c' \times h \times w$, where $c' > c$, which equals x on the c first channels and 0 on the $c' - c$ other channels.

Reducing dimensionsionality. Dimensionality reduction is another essential operation in neural networks. On one hand, its goal is to reduce the number of parameters and thus the amount of computation required to build the network. On the other hand it allows the model to progressively map the input space on the output dimension, which corresponds in many cases to the number of different labels K. In this context, several operations exist: pooling layers are used to extract information present in a region of the feature map generated by a convolution layer. One can easily adapt pooling layers (*e.g.* max and average) to make them 1-Lipschitz [Bartlett et al., 2017]. Finally, a simple method to reduce the dimension is the product with a non-square matrix. We simply implement it as the truncation of the output. This obviously maintains the Lipschitz constant.

6.2.3 Computing spectral norms

Our Convex Potential Layer, described in Equation 6.2.2, can be adapted to any kind of linear transformations (*i.e.* Dense or Convolutional) but requires the computation of the spectral norm for these transformations. Given that computation of the spectral norm of a linear operator is known to be NP-hard [Steinberg, 2005], an efficient approximate method is required during training to keep the complexity tractable.

Many techniques exist to approximate the spectral norm (or the largest singular value), and most of them exhibit a trade-off between efficiency and accuracy. Several methods exploit the structure of convolutional layers to build an upper bound on the spectral norm of the linear transform performed by the convolution [Araujo et al., 2021, Jia et al., 2017, Singla et al., 2021b]. While these methods are generally efficient, they can less relevant and adapted to certain settings. For instance in our context, using a loose upper bound of the spectral norm will hinder the expressive power of the layer and make it too contracting.

For these reasons we rely on the Power Iteration Method (PM). This method converges at a geometric rate towards the largest singular value of a matrix. More precisely the convergence rate for a given matrix $\mathbf{W}$ is $O((\frac{\lambda_2}{\lambda_1})^k)$ after k iterations, independently from the choice for the starting vector, where $\lambda_1 > \lambda_2$ are the two largest singular values of $\mathbf{W}$. While it can appear to be computationally expensive due to the large number of required iterations for convergence, it is possible to drastically reduce the number of iterations during training. Indeed, as in [Miyato et al., 2018], by considering that the weights' matrices $\mathbf{W}$ change slowly during training, one can perform only one iteration of the PM for each step of the training and let the algorithm slowly

#	S	M	L	XL
Conv. Layers	20	30	50	70
Channels	45	60	90	120
Lin. Layers	7	10	15	15
Lin. Features	2048	2048	4096	4096

Table 6.1: Architectures description for our Convex Potential Layers (CPL) neural networks with different capacities. We vary the number of Convolutional Convex Potential Layers, the number of Linear Convex Potential Layers, the number of channels in the convolutional layers and the width of fully connected layers. They will be reported respectively as CPL-S, CPL-M, CPL-L and CPL-XL.

converges along with the training process[2]. We describe with more details in Algorithm 5, the operations performed during a forward pass with a Convex Potential Layer.

However for evaluation purpose, we need to compute the certified adversarial robustness, and this requires to ensure the convergence of the PM. Therefore, we perform 100 iterations for each layer[3] at inference time. Also note that at inference time, the computation of the spectral norm only needs to be performed once for each layer.

6.3 Experiments

To evaluate our new 1-Lipschitz Convex Potential Layers, we carry out an extensive set of experiments. In this section, we first describe the details of our experimental setup. We then recall the concurrent approaches that build 1-Lipschitz Neural Networks and stress their limitations. Our experimental results are finally summarized in ection 6.3.1. By computing the certified and empirical adversarial accuracy of our networks on CIFAR10 and CIFAR100 classification tasks [Krizhevsky and Hinton, 2009], we show that our architecture is competitive with state-of-the-art methods (Sections 6.3.3). We also study the influence of some hyperparameters and demonstrate the stability and the scalability of our approach by training very deep neural networks up to 1000 layers without normalization tricks or gradient clipping.

6.3.1 Training and Architectural Details

We demonstrate the effectiveness of our approach on a classification task with CIFAR10 and CIFAR100 datasets [Krizhevsky and Hinton, 2009]. We use a similar training configuration to the one proposed in [Trockman et al., 2021] We trained our networks with a batch size of 256 over 200 epochs. We use standard data augmentation (i.e.,, random cropping and flipping), a learning rate of 0.001 with Adam optimizer [Diederik P. Kingma, 2014] without weight decay and a piecewise triangular learning rate scheduler. We used a margin parameter in the loss set to 0.7.

As other usual convolutional neural networks, we first stack few Convolutional CPLs and then stack some Linear CPLs for classification tasks. To validate the performance and the scalability of our layers, we will evaluate four different variations of different hyperparameters as described in Table 6.1, respectively named CPL-S, CPL-M, CPL-L and CPL-XL, ranked according to the

[2]Note that a typical training requires approximately 200K steps where 100 steps of PM is usually enough for convergence

[3]100 iterations of Power Method is sufficient to converge with a geometric rate.

	Clean Accuracy	Provable Accuracy (ε)			Time per epoch (s)
		36/255	72/255	108/255	
CPL-S	75.6	62.3	46.9	32.2	21.9
CPL-M	76.8	63.3	47.5	32.5	40.0
CPL-L	77.7	63.9	48.1	32.9	93.4
CPL-XL	78.5	64.4	48.0	33.0	163
Cayley (KW3)	74.6	61.4	46.4	32.1	30.8
SOC-10	77.6	62.0	45.0	29.5	33.4
SOC-20	78.0	62.7	46.0	30.3	52.2
SOC+-10	76.2	62.6	47.7	34.2	N/A
SOC+-20	76.3	62.6	48.7	36.0	N/A

Table 6.2: Results on the CIFAR10 dataset on standard and provably certifiable accuracies for different values of perturbations ε on CPL (ours), SOC and Cayley models. The average time per epoch in seconds is also reported in the last column. None of these networks uses Last Layer Normalization.

number of parameters they have. In all our experiments, we made 3 independent trainings to evaluate accurately the models. All reported results are the average of these 3 runs.

6.3.2 Concurrent Approaches

We compare our networks with SOC [Singla and Feizi, 2021] and Cayley Trockman et al. [2021] networks which are to our knowledge the best performing approaches for deterministic 1-Lipschitz Neural Networks. Since our layers are fundamentally different from these ones, we cannot compare with the same architectures. We reproduced SOC results for with 10 and 20 layers, that we call respectively SOC-10 and SOC-20 in the same training setting, *i.e.* normalized inputs, cross entropy loss, SGD optimizer with learning rate 0.1 and multi-step learning rate scheduler. For Cayley layers networks, we reproduced their best reported model, *i.e.* KWLarge with width factor of 3.

The work of Singla et al. [2021a] propose three methods to improve certifiable accuracies from SOC layers: a new HouseHolder activation function (HH), last layer normalization (LLN), and certificate regularization (CR). The code associated with this approach is not open-sourced yet, so we just reported the results from their paper in ours results (Tables 6.2 and 6.3) under the name SOC+. We were being able to implement the LLN method in all models. This method largely improve the result of all methods on CIFAR100, so we used it for all networks we compared on CIFAR100 (ours and concurrent approaches).

6.3.3 Results

In this section, we present our results on adversarial robustness. We provide results on provable ℓ_2 robustness as well as empirical robustness on CIFAR10 and CIFAR100 datasets for all our models and the concurrent ones

Certified Adversarial Robustness. Results on CIFAR10 and CIFAR100 dataset are reported respectively in Tables 6.2 and 6.3. We also plotted certified accuracy in function of ε

	Clean Accuracy	Provable Accuracy (ε)			Time per epoch (s)
		36/255	72/255	108/255	
CPL-S	44.0	29.9	19.1	11.0	22.4
CPL-M	45.6	31.1	19.3	11.3	40.7
CPL-L	46.7	31.8	20.1	11.7	93.8
CPL-XL	47.8	33.4	20.9	12.6	164
Cayley (KW3)	43.3	29.2	18.8	11.0	31.3
SOC-10	48.2	34.3	22.7	14.0	33.8
SOC-20	48.3	34.4	22.7	14.2	52.7
SOC+-10	47.1	34.5	23.5	15.7	N/A
SOC+-20	47.8	34.8	23.7	15.8	N/A

Table 6.3: Results on the CIFAR100 dataset on standard and provably certifiable accuracies for different values of perturbations ε on CPL (ours), SOC and Cayley models. The average time per epoch in seconds is also reported in the last column. All the reported networks use Last Layer Normalization.

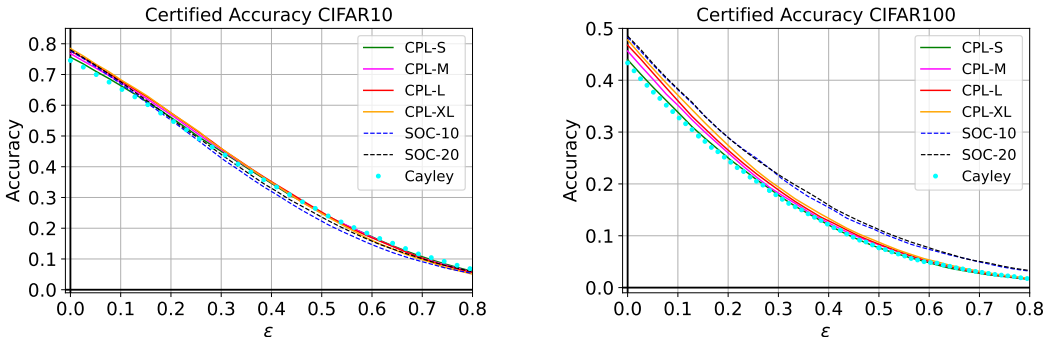


Figure 6.1: Certifiably robust accuracy in function of the perturbation ε for our CPL networks and its concurrent approaches (SOC and Cayley models) on CIFAR10 and CIFAR100 datasets.

on Figure 6.1. On CIFAR10, our method outperforms the concurrent approaches in terms of standard and certified accuracies for every level of ε except SOC+ that uses additional tricks we did not use. On CIFAR100, our method performs slightly under the SOC networks but better than Cayley networks. Overall, our methods reach competitive results with SOC and Cayley layers.

Note that we observe a small gain using larger and deeper architectures for our models. This gain is less important as ε increases but the gain is non negligible for standard accuracies. In term of training time, our small architecture (CPL-S) trains very fast compared to other methods, while larger ones are longer to train.

Empirical Adversarial Robustness. We also reported in Figure 6.2 the accuracy of all the models against PGD ℓ_2-attack [Kurakin et al., 2016, Madry et al., 2018] for various levels of ϵ. We used 10 iterations for this attack. We remark here that our methods brings a large gain of robust accuracy over all other methods. On CIFAR10 for $\varepsilon = 0.8$, the gain of CPL-S over

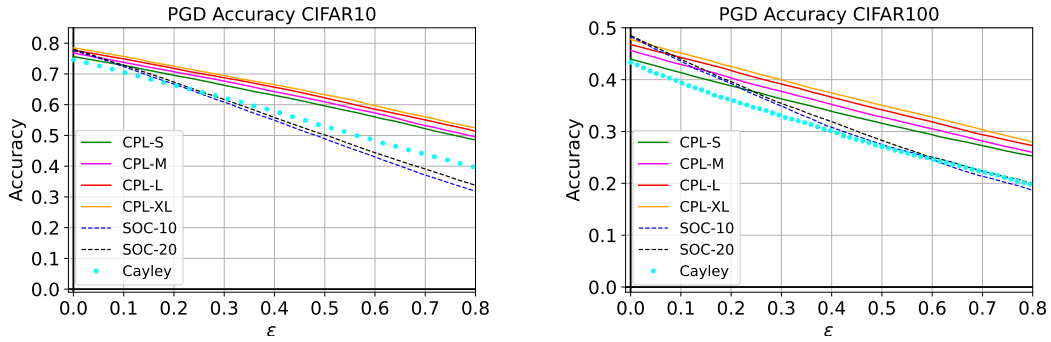


Figure 6.2: Accuracy against PGD attack with 10 iterations in function of the perturbation ε for our CPL networks and its concurrent approaches on CIFAR10 and CIFAR100 datasets.

SOC-10 approach is more than 10%. For CIFAR100, the gain is about 10% too for $\varepsilon = 0.6$. We remark that using larger architectures lead in a more substantial gain in empirical robustness.

Our layers only provide an upper bound on the Lipschitz constant, while orthonormal layers as Cayley and SOC are built to exactly preserve the norms. This might negatively influence the certified accuracy since the effective Lipschitz constant is smaller than the theoretical one, hence leading to suboptimal certificates. This might explain why our method performs so well of empirical robustness task.

	Batch size	**Clean Accuracy**	**Provable Accuracy (ε)**			**Time per epoch (s)**
			36/255	72/255	108/255	
CPL-S	64	76.5	62.9	47.3	32.0	48
	128	76.1	62.8	47.1	32.3	31
	256	75.6	62.3	46.9	32.2	22
CPL-M	64	77.4	63.6	47.4	32.1	77
	128	77.2	63.5	47.5	32.1	50
	256	76.8	63.2	47.4	32.4	40
CPL-L	64	78.4	64.2	47.8	32.2	162
	128	78.2	64.3	47.9	32.5	109
	256	77.6	63.9	48.1	32.7	93
CPL-XL	64	78.9	64.2	47.2	31.2	271
	128	78.9	64.2	47.5	31.8	198
	256	78.5	64.4	47.8	32.4	163

Table 6.4: Results on the CIFAR10 dataset on standard and provably certifiable accuracies for different values of perturbations ε on CPL (ours) models with various batch sizes. The average time per epoch in seconds is also reported in the last column. All the reported networks use Last Layer Normalization.

Effect of Batch Size in Training. In Tables 6.4 and 6.5, we tried three different batch sizes (64, 128 and 256) for training our networks on CIFAR10 and CIFAR100 datasets, we remark

	Batch size	Clean Acc.	Provable Acc. (ε)			Time per epoch (s)
			36/255	72/255	108/255	
CPL-S	64	45.6	30.8	19.3	11.2	47
	128	44.9	30.7	19.2	11.0	31
	256	44.0	29.9	19.1	10.9	23
CPL-M	64	46.6	31.6	19.6	11.6	78
	128	46.3	31.1	19.7	11.5	55
	256	45.6	31.1	19.3	11.3	41
CPL-L	64	48.1	32.7	20.3	11.7	163
	128	47.4	32.3	20.0	11.8	116
	256	46.8	31.8	20.1	11.7	95
CPL-XL	64	49.0	33.7	21.1	12.0	293
	128	48.0	33.7	21.0	12.1	209
	256	47.8	33.4	20.9	12.6	164

Table 6.5: Results on the CIFAR100 dataset on standard and provably certifiable accuracies for different values of perturbations ε on CPL (ours) models with various batch sizes. The average time per epoch in seconds is also reported in the last column. All the reported networks use Last Layer Normalization.

a gain in standard accuracy in reducing the batch size for all settings. As the perturbation becomes larger, the gain in accuracy is reduced and can even in some cases we may loose some points in robustness.

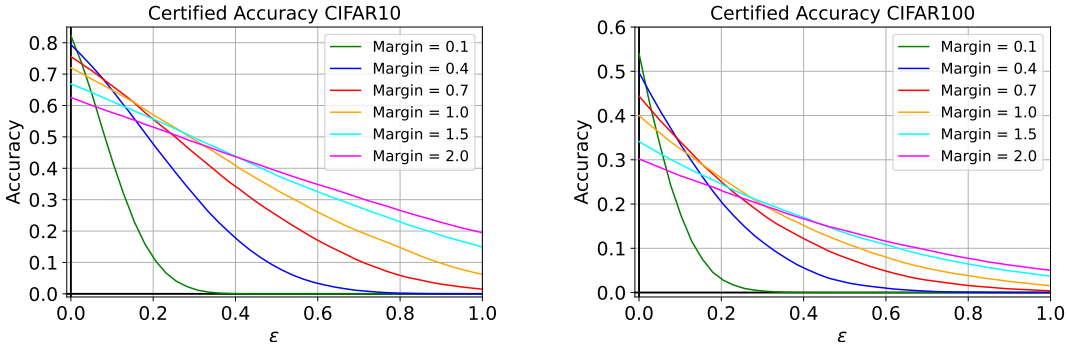


Figure 6.3: Certifiably robust accuracy in function of the perturbation ε for our CPL-S network with different margin parameters on CIFAR10 and CIFAR100 datasets.

Effect of the Margin Parameter. In these experiments we varied the margin parameter in the margin loss in Figures 6.3 and 6.4. It clearly exhibits a tradeoff between standard and robust accuracy. When the margin is large, the standard accuracy is low, but the level of robustness remain high even for "large" perturbations. On the opposite, when the margin is small, we get a high standard accuracy but we are unable to keep a good robustness level as the perturbation increases. It is verified both on certified and empirical robustness.

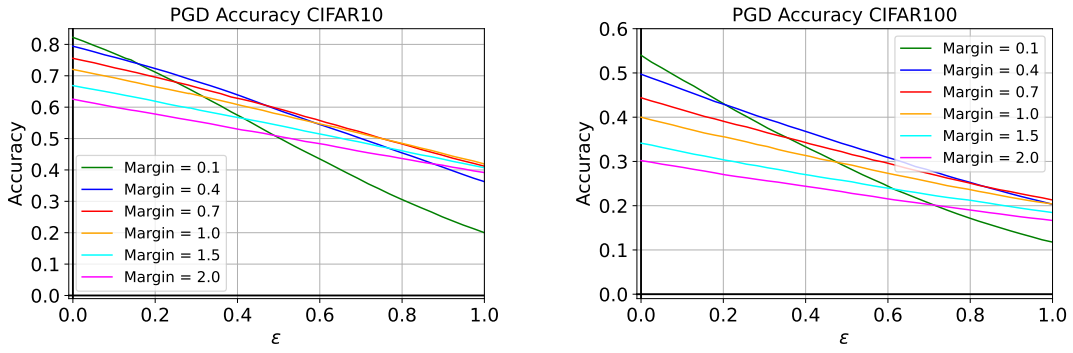


Figure 6.4: Certifiably robust accuracy in function of the perturbation ε for our CPL-S network with different margin parameters on CIFAR10 and CIFAR100 datasets.

6.3.4 Training stability: scaling up to 1000 layers

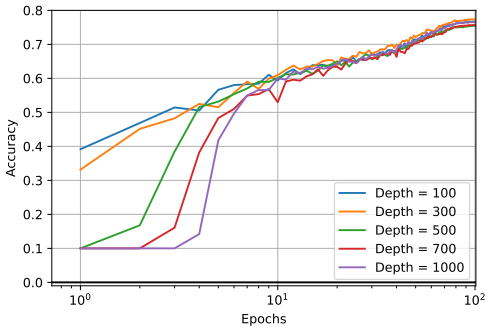


Figure 6.5: Standard test accuracy in function of the number of epochs (log-scale) for various depths for our neural networks $(100, 300, 500, 700, 1000)$.

While the Residual Network architecture limits, by design, gradient vanishing issues, it still suffers from exploding gradients in many cases [Hayou et al., 2021]. To prevent such scenarii, batch normalization layers [Ioffe and Szegedy, 2015] are used in most Residual Networks to stabilize the training.

Recently, several works [Farnia et al., 2019, Miyato et al., 2018] have proposed to normalize the linear transformation of each layer by their spectral norm. Such a method would limit exploding gradients but would again suffer from gradient vanishing issues. Indeed, spectral normalization might be too restrictive: dividing by the spectral norm can make other singular values vanishingly small. While more computationally expensive (spectral normalization can be done with 1 Power Method iteration), orthogonal projections prevent both exploding and vanishing issues.

On the contrary the architecture proposed in this paper has the advantage to naturally control the gradient norm of the output with respect to a given layer. Therefore, our architecture can get the best of both worlds: limiting exploding and vanishing issues while maintaining scalability. To demonstrate the scalability of our approach, we experiment the ability to scale our architecture

to very high depth (up to 1000 layers) without any additional normalization/regularization tricks, such as Dropout [Srivastava et al., 2014], Batch Normalization [Ioffe and Szegedy, 2015] or gradient clipping [Pascanu et al., 2013]. With the work done by Xiao et al. [2018], which leverage Dynamical Isometry and a Mean Field Theory to train a 10000 layers neural network, we believe, to the best of our knowledge, to be the second to perform such training. For sake of computation efficiency, we limit this experiment to architecture with 30 feature maps. We report the accuracy in terms of epochs for our architecture in Figure 6.5 for a varying number of convolutional layers. It is worth noting that for the deepest networks, it may take a few epochs before the start of convergence. As Xiao et al. [2018], we remark there is no gain in using very deep architecture for this task.

6.3.5 Relaxing linear layers

	h = 1.0	**h = 0.1**	**h = 0.01**
Clean	85.10	82.23	78.53
PGD ($\varepsilon = 36/255$)	61.45	62.99	60.98

The table above shows the result of the relaxed training of our StableBlock architecture, i.e. we fixed the step h_t in the discretized convex potential flow of Proposition 16. Increasing the constant h allows for an important improvement in the clean accuracy, but we loose in robust empirical accuracy. While computing the certified accuracy is not possible in this case due to the unknown value of the Lipschitz constant, we can still notice that the training of the network are still stable without normalization tricks, and offer a non-negligible level of robustness.

6.4 Discussions and Open questions

In this chapter, we presented a new generic method to build 1-Lipschitz layers. We leverage the continuous time dynamical system interpretation of Residual Networks and show that using convex potential flows naturally defines 1-Lipschitz neural networks. After proposing a parametrization based on Input Convex Neural Networks [Amos et al., 2017], we show that our models reach competitive results in classification and robustness in comparison which other existing 1-Lipschitz approaches. We also experimentally show that our layers provide scalable approaches without further regularization tricks to train very deep architectures.

Exploiting the ResNet architecture for devising flows have been an important research topic. For example, in the context of generative modeling, Invertible Neural Networks [Behrmann et al., 2019] and Normalizing Flows [Rezende and Mohamed, 2015, Verine et al., 2021] are both import research topic. More recently, Sylvester Normalizing Flows [van den Berg et al., 2018] or Convex Potential Flows [Huang et al., 2021a] have had similar ideas to this present work but for a very different setting and applications. In particular, they did not have interest in the contraction property of convex flows and the link with adversarial robustness have been under-exploited.

Expressivity of discretized convex potential flows. Propoisition 15 suggests to constraint the symmetric part of the Jacobian of F_t. We proposed to decompose F_t as a sum of potential gradient and skew symmetric matrix. Finding other parametrizations is an open challenge. Our models may not express all 1-Lipschitz functions. Knowing which functions can be approximated by our CPL layers is difficult even in the linear case. Indeed, let us define $\mathcal{S}_1(\mathbb{R}^{d\times d})$ the space of real symmetric matrices with singular values bounded by 1. Let us also

define $\mathcal{M}_1(\mathbb{R}^{d\times d})$ the space of real matrices with singular values bounded by 1 in absolute value. Let $\mathcal{P}(\mathbb{R}^{d\times d}) = \{A \in \mathbb{R}^{d\times d} | \exists n \in \mathbb{N}, S_1, \dots, S_n \in \mathcal{S}_1(\mathbb{R}^d \times d) \text{ s.t. } A = S_1 \dots S_n\}$. Then one can prove[4] that $\mathcal{P}(\mathbb{R}^{d\times d}) \neq \mathcal{M}_1(\mathbb{R}^{d\times d})$. Thus there exists $A \in \mathcal{M}_1(\mathbb{R}^{d\times d})$ such that for all matrices n, for all matrices $S_1, \dots, S_n \in \mathcal{S}_1(\mathbb{R}^{d\times d})$ such that $M \neq S_1, \dots, S_n$. Applied to the expressivity of discretized convex potential flows, the previous result means that there exists a 1-Lipschitz linear function that cannot be approximated as a discretized flow of any depth of convex linear 1-smooth potential flows as in Proposition 16. Indeed such a flow would write: $x \mapsto \prod_i (1 - 2S_i)x$ where S_i are symmetric matrices whose eigenvalues are in $[0, 1]$, in other words such transformations are exactly described by $x \mapsto Mx$ for some $M \in \mathcal{P}(\mathbb{R}^{d\times d})$. This is an important question that requires further investigation.

Going beyond ResNets One can also think of extending our work by the study of other dynamical systems. Recent architectures such as Hamiltonian Networks [Greydanus et al., 2019] and Momentum Networks [Sander et al., 2021a] exhibit interesting properties and it worth digging into these architectures to build Lipschitz layers. Finally, we hope to use similar approaches to build robust Recurrent Neural Networks [Sherstinsky, 2020] and Transformers [Vaswani et al., 2017]. For Transformers, Sander et al. [2021b], Vuckovic et al. [2020] has proposed a dynamical system interpretation of a flow on particles (i.e. the words in the the initial sentence). This can be seen as an interacting flow over a distributions. The question of robustness and Lipschitzness is way more technical since it implies Lipshitzness in the space of a distribution. One could imagine to use optimal transport [Villani, 2003] and Wasserstein Gradient flows [Ambrosio et al., 2005] as tools for deriving Lipschitz guarantees for Transformers.

[4] A proof and justification of this result can be found here: `https://mathoverflow.net/questions/60174/factorization-of-a-real-matrix-into-hermitian-x-hermitian-is-it-stable`

Chapter 7

Conclusion

Contents

7.1 Summary of the thesis

In this thesis, we studied the problem of classication in presernce of adversaries from different point of views for theoretical and practical finalities. We have tried to analyze the problem using both a high level and a more precise analysis. We summarize our findings as follows.

1. We provide a better understanding of the adversarial problem studying the nature of equilibria in this game. We proved the existence of mixed Nash equilibria for very general assumptions. We hope this research directions will lead to principled results that can be used in practice for better defending against adversarial examples.
2. We studied and closed the problem of calibration in the adversarial binary-classification setting providing necessary and sufficient conditions. We paved a way to prove consistency results, and hope being able to conclude on consistency of shifted odd losses. It remains to find necessary and sufficient conditions for consistency.
3. We derived a principled way based on dynamical system to build 1-Lipschitz layers. Interestingly, we recovered some existing methods from the literature, but we were also able to build new interesting layers, namely the Convex Potential Layers. We hope this work would lead to study other possible dynamical systems and provide new provably robust neural networks.

7.2 Open Questions

7.2.1 Understanding Randomization in Adversarial Classification

- Statistical Bounds for Adversarial Robustness in the Case of Randomized Classifiers

- Designing an Algorithm for computing Nash Equilibria in the General Case

7.2.2 Loss Calibration General Results

- The non realisable case is difficult: showing either negative/positive general results
- Further developing the margin loss analysis

7.2.3 Exploiting the architecture of Neural Networks to get Guarantees

- Exploiting Helmoltz decomposition of flows
- Exploiting other flows (Hamiltonian, Momentum, etc.)

Bibliography

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Marc Abeille, Alessandro Lazaric, et al. Linear thompson sampling revisited. *Electronic Journal of Statistics*, 11(2):5165–5197, 2017.

Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

Abdullah Al-Dujaili and Una-May O'Reilly. There are no bit parts for sign bits in black-box attacks, 2019.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, 2017.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. *arXiv preprint arXiv:1912.00049*, 2019.

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, 2019.

Alexandre Araujo, Rafael Pinot, Benjamin Negrevergne, Laurent Meunier, Yann Chevaleyre, Florian Yger, and Jamal Atif. Robust neural networks using randomized adversarial training. *arXiv preprint arXiv:1903.10219*, 2019.

Alexandre Araujo, Benjamin Negrevergne, Yann Chevaleyre, and Jamal Atif. On lipschitz regularization of convolutional layers using toeplitz matrix theory. *Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018a. PMLR.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018b. PMLR. URL `http://proceedings.mlr.press/v80/athalye18b.html`.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. Adversarial learning guarantees for linear hypotheses and neural networks. *International Conference on Machine Learning*, 2020.

Pranjal Awasthi, Natalie Frank, Anqi Mao, Mehryar Mohri, and Yutao Zhong. Calibration and consistency of adversarial surrogate losses. *arXiv preprint arXiv:2104.09658*, 2021a.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. On the existence of the adversarial bayes classifier. *Advances in Neural Information Processing Systems*, 34, 2021b.

Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. A finer calibration analysis for adversarial robustness. *arXiv preprint arXiv:2105.01550*, 2021c.

Han Bao, Clay Scott, and Masashi Sugiyama. Calibrated surrogate losses for adversarially robust classification. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 408–451. PMLR, 09–12 Jul 2020. URL `http://proceedings.mlr.press/v125/bao20a.html`.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, 2019.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Dimitir P Bertsekas and Steven Shreve. *Stochastic optimal control: the discrete-time case*. 2004.

Louis Béthune, Alberto González-Sanz, Franck Mamalet, and Mathieu Serrurier. The many faces of 1-lipschitz neural networks. *arXiv preprint arXiv:2104.05097*, 2021.

H.-G. Beyer. *The Theory of Evolution Strategies*. Natural Computing Series. Springer, Heideberg, Berlin, 2001. URL `http://books.google.com/books?id=8tbInLufkTMC`.

Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. *Advances in Neural Information Processing Systems*, 32:7496–7508, 2019.

Battista Biggio, Giorgio Fumera, and Fabio Roli. Evade hard multiple classifier systems. In *Applications of Supervised and Unsupervised Ensemble Methods*, pages 15–38. Springer, 2009.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

Åke Björck et al. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 1971.

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

Avishek Joey Bose, Gauthier Gidel, Hugo Berard, Andre Cianflone, Pascal Vincent, Simon Lacoste-Julien, and William L. Hamilton. Adversarial example games, 2021.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Stephen Boyd. Subgradient methods. 2003.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 547–555, New York, NY, USA, 2011. Association for Computing Machinery. doi: 10.1145/2020408.2020495.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 42–1, 2012.

Leon Bungert, Nicolás García Trillos, and Ryan Murray. The geometry of adversarial training in binary classification. *arXiv preprint arXiv:2111.13613*, 2021.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

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

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

Tony F Chan and Selim Esedoglu. Aspects of total variation regularized l 1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence*, 2018a.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018b.

Lénaïc Chizat. Convergence rates of gradient methods for convex optimization in the space of measures. *arXiv preprint arXiv:2105.08368*, 2021a.

Lenaic Chizat. Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming*, pages 1–46, 2021b.

Alexandre Chotard, Anne Auger, and Nikolaus Hansen. Cumulative step-size adaptation on linear functions. In Carlos A. Coello Coello, Vincenzo Cutello, Kalyanmoy Deb, Stephanie Forrest, Giuseppe Nicosia, and Mario Pavone, editors, *Parallel Problem Solving from Nature - PPSN XII*, pages 72–81, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-32937-1.

Konstantina Christakopoulou and Arindam Banerjee. Adversarial attacks on an oblivious recommender. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 322–330, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362436. doi: 10.1145/3298689.3347031. URL `https://doi.org/10.1145/3298689.3347031`.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 854–863, 2017.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. *CoRR*, abs/1902.02918, 2019. URL `http://arxiv.org/abs/1902.02918`.

Patrick L Combettes and Jean-Christophe Pesquet. Lipschitz certificates for layered network structures driven by averaged activation operators. *SIAM Journal on Mathematics of Data Science*, 2020.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020a.

Francesco Croce et al. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, 2020b.

Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of adversaries. *Advances in Neural Information Processing Systems*, 31, 2018.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004.

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296, 2010.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Jimmy Ba Diederik P. Kingma. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2014.

Elvis Dohmatob. Generalized no free lunch theorem for adversarial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1646–1654, Long Beach, California, USA, 2019. PMLR. URL `http://proceedings.mlr.press/v97/dohmatob19a.html`.

Carles Domingo-Enrich, Samy Jelassi, Arthur Mensch, Grant Rotskoff, and Joan Bruna. A mean-field analysis of two-player zero-sum games. *Advances in Neural Information Processing Systems*, 33:20215–20226, 2020.

Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE, 2018.

John Duchi, Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 2017.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811. PMLR, 2019.

Ky Fan. Minimax theorems. *Proceedings of the National Academy of Sciences of the United States of America*, 39(1):42, 1953.

Farzan Farnia, Jesse Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *International Conference on Learning Representations*, 2019.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pages 1632–1640, 2016.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Evrard Garcelon, Baptiste Roziere, Laurent Meunier, Olivier Teytaud, Alessandro Lazaric, and Matteo Pirotta. Adversarial attacks on linear contextual bandits. *arXiv preprint arXiv:2002.03839*, 2020.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

Amir Globerson et al. Nightmare at test time: Robust learning by feature deletion. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

Gene H Golub et al. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 2000.

Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6 (4):1–19, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

Samuel J Greydanus, Misko Dzumba, and Jason Yosinski. Hamiltonian neural networks. 2019.

Ziwei Guan, Kaiyi Ji, Donald J Bucci Jr, Timothy Y Hu, Joseph Palombo, Michael Liston, and Yingbin Liang. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. *arXiv preprint arXiv:2002.07214*, 2020.

Benjamin Guedj. A primer on pac-bayesian learning. *arXiv preprint arXiv:1901.05353*, 2019.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. *arXiv preprint arXiv:2104.13733*, 2021.

Anupam Gupta, Tomer Koren, and Kunal Talwar. Better algorithms for stochastic bandits with adversarial corruptions. *arXiv preprint arXiv:1902.08647*, 2019.

Eldad Haber et al. Stable architectures for deep neural networks. *Inverse problems*, 2017.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 11(1), 2003.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis) DOI=http://dx.doi.org/10.1145/2827872*, 5(4):1–19, 2015.

Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable resnet. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

Chin-Wei Huang, Ricky T. Q. Chen, Christos Tsirigotis, and Aaron Courville. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. In *International Conference on Learning Representations*, 2021a.

Lei Huang, Li Liu, Fan Zhu, Diwen Wan, Zehuan Yuan, Bo Li, and Ling Shao. Controllable orthogonalization in training dnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020a.

Yifei Huang, Yaodong Yu, Hongyang Zhang, Yi Ma, and Yuan Yao. Adversarial robustness of stabilized neuralodes might be from obfuscated gradients. *Mathematical and Scientific Machine Learning*, 2020b.

Yujia Huang, Huan Zhang, Yuanyuan Shi, J Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. In *Advances in Neural Information Processing Systems*, 2021b.

Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. Targeted attacks on deep reinforcement learning agents through adversarial observations. *arXiv preprint arXiv:1905.12282*, 2019.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, pages 2137–2146, 2018a.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018b.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

Nicole Immorlica, Karthik Abinav Sankararaman, Robert E. Schapire, and Aleksandrs Slivkins. Adversarial bandits with knapsacks. *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 202–219, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

Kui Jia, Dacheng Tao, Shenghua Gao, and Xiangmin Xu. Improving training of deep neural networks via singular value bounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 864–872, 2019.

Ruiwei Jiang and Yongpei Guan. Risk-averse two-stage stochastic program with distributional ambiguity. *Operations Research*, 66(5):1390–1405, 2018.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. arxiv, pages arxiv–1907. 2019.

Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3640–3649, 2018.

Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Geometric robustness of deep networks: analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4441–4449, 2018.

Sayash Kapoor, Kumar Kshitij Patel, and Purushottam Kar. Corruption-tolerant bandit learning. *Machine Learning*, 108(4):687–715, 2019.

Michael Kearns and Ming Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

Carson Kent, Jose Blanchet, and Peter Glynn. Frank-wolfe methods in probability space. *arXiv preprint arXiv:2105.05352*, 2021.

Justin Khim and Po-Ling Loh. Adversarial risk bounds for binary classification via function transformation. *arXiv preprint arXiv:1810.09519*, 2, 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. In *International Conference on Machine Learning*, 2020a.

Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 69–75. IEEE, 2020b.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Tor Lattimore and Csaba Szepesvári. Bandit algorithms. Pre-publication version, 2018. URL `http://downloads.tor-lattimore.com/banditbook/book.pdf`.

Mathias Lechner, Ramin Hasani, Radu Grosu, Daniela Rus, and Thomas A Henzinger. Adversarial training is not ready for robot learning. *arXiv preprint arXiv:2103.08187*, 2021.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

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

M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 727–743, 2018.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in neural information processing systems*, pages 1885–1893, 2016.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.500. URL `https://aclanthology.org/2020.emnlp-main.500`.

Mingjie Li, Lingshen He, and Zhouchen Lin. Implicit euler skip connections: Enhancing adversarial robustness via numerical stability. In *International Conference on Machine Learning*, pages 5874–5883. PMLR, 2020b.

Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Joern-Henrik Jacobsen. Preventing gradient attenuation in lipschitz constrained convolutional networks. In *Advances in Neural Information Processing Systems*, 2019a.

Yingkai Li, Edmund Y Lou, and Liren Shan. Stochastic linear optimization with adversarial corruption. *arXiv preprint arXiv:1909.02109*, 2019b.

Chang Liu, Bo Li, Yevgeniy Vorobeychik, and Alina Oprea. Robust linear regression against training data poisoning. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 91–102, 2017.

Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. *arXiv preprint arXiv:1905.06494*, 2019.

Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. signSGD via zeroth-order oracle. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=BJe-DsC5Fm`.

Phil Long and Rocco Servedio. Consistency versus realizable h-consistency for multiclass classification. In *International Conference on Machine Learning*, pages 801–809. PMLR, 2013.

Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 114–122, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355599. doi: 10.1145/3188745.3188918. URL `https://doi.org/10.1145/3188745.3188918`.

Thodoris Lykouris, Max Simchowitz, Aleksandrs Slivkins, and Weidong Sun. Corruption robust exploration in episodic reinforcement learning. *ArXiv*, abs/1911.08689, 2019.

Yuzhe Ma, Kwang-Sung Jun, Lihong Li, and Xiaojin Zhu. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, pages 186–204. Springer, 2018.

Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. Policy poisoning in batch reinforcement learning and control. In *Advances in Neural Information Processing Systems*, pages 14543–14553, 2019.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. ACL. URL `http://www.aclweb.org/anthology/P11-1015`.

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

J. Matyas. Random optimization. *Automation and Remote control*, 26:246–253, 1965.

John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. A proposal for the dartmouth summer research project on artificial intelligence. *AI magazine*, 27(4):12–12, 1955.

Bhaskar Mehta and Wolfgang Nejdl. Attack resistant collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 75–82, 2008.

Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4636–4645, Long Beach, California, USA, 09–15 Jun 2019. PMLR, PMLR. URL `http://proceedings.mlr.press/v97/moon19a.html`.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94. Ieee, 2017.

Herve Moulin. *Game theory for the social sciences*. NYU press, 1986.

Ronghui Mu, Wenjie Ruan, Leandro Soriano Marcolino, and Qiang Ni. Sparse adversarial video attacks with spatial transformations. *arXiv preprint arXiv:2111.05468*, 2021.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29, 2016.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *J. Mach. Learn. Res.*, 18:18:1–18:65, 2017. URL `http://jmlr.org/papers/v18/14-467.html`.

N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 372–387, March 2016. doi: 10.1109/EuroSP.2016.36.

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

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016b.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pages 506–519, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4944-4. doi: 10.1145/3052973.3053009. URL `http://doi.acm.org/10.1145/3052973.3053009`.

Haekyu Park, Jinhong Jung, and U Kang. A comparative study of matrix factorization and random walk with restart in recommender systems. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 756–765. IEEE, 2017.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, 2013.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (NeurIPS), 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Juan C. Perdomo and Yaron Singer. Robust attacks against multiple classifiers. *arXiv preprint arXiv:1906.02816*, 2019.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization: the case of the exponential family. *arXiv preprint arXiv:1902.01148*, 2019.

Rafael Pinot, Raphael Ettedgui, Geovani Rizk, Yann Chevaleyre, and Jamal Atif. Randomization matters how to defend against strong adversarial attacks. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Muni Sreenivas Pydi and Varun Jog. Adversarial risk via optimal transport and optimal couplings. *IEEE Transactions on Information Theory*, 67(9):6031–6052, 2021a. doi: 10.1109/TIT.2021.3100107.

Muni Sreenivas Pydi and Varun Jog. The many faces of adversarial risk. *Advances in Neural Information Processing Systems*, 34, 2021b.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAi, 2018.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

Hamed Rahimian, Güzin Bayraksan, and Tito Homem-de Mello. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. *Mathematical Programming*, 173(1):393–430, 2019.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

J. Rapin and O. Teytaud. Nevergrad - A gradient-free optimization platform. `https://GitHub.com/FacebookResearch/Nevergrad`, 2018.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, 2015.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

Raymond Ros and Nikolaus Hansen. A simple modification in cma-es achieving linear time and space complexity. In Günter Rudolph, Thomas Jansen, Nicola Beume, Simon Lucas, and Carlo Poloni, editors, *Parallel Problem Solving from Nature – PPSN X*, pages 296–305, Berlin, Heidelberg, 2008. Springer, Springer Berlin Heidelberg. ISBN 978-3-540-87700-4.

S. Rota Bulò, B. Biggio, I. Pillai, M. Pelillo, and F. Roli. Randomized prediction games for adversarial machine learning. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2466–2478, 2017.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, 2019.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

Michael E. Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Momentum residual neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, 2021a.

Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. *arXiv preprint arXiv:2110.11773*, 2021b.

M. Schumer and K. Steiglitz. Adaptive step size random search. *Automatic Control, IEEE Transactions on*, 13:270–276, 1968.

Hanie Sedghi, Vineet Gupta, and Philip Long. The singular values of convolutional layers. In *International Conference on Learning Representations*, 2018.

Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *International Conference on Learning Representtation*, 2018.

Soroosh Shafieezadeh Abadeh, Peyman M Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. *Advances in Neural Information Processing Systems*, 28, 2015.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

Carl-Johann Simon-Gabriel, Yann Ollivier, Leon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In *International Conference on Machine Learning*, pages 5809–5817, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sahil Singla and Soheil Feizi. Skew orthogonal convolutions. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Sahil Singla, Surbhi Singla, and Soheil Feizi. Householder activations for provable robustness against adversarial attacks. *arXiv preprint arXiv:2108.04062*, 2021a.

Sahil Singla et al. Fantastic four: Differentiable and efficient bounds on singular values of convolution layers. In *International Conference on Learning Representations*, 2021b.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Maurice Sion. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958. URL `https://projecteuclid.org:443/euclid.pjm/1103040253`.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

Daureen Steinberg. Computation of matrix norms with applications to robust optimization. *Research thesis, Technion-Israel University of Technology*, 2005.

Ingo Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26(2):225–287, 2007.

Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. Stealthy and efficient adversarial attacks against deep reinforcement learning. *To appear in Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

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

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems*, pages 5866–5876, 2019.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Nicolás García Trillos and Ryan Murray. Adversarial classification: Necessary conditions and geometric flows. *arXiv preprint arXiv:2011.10797*, 2020.

Asher Trockman et al. Orthogonalizing convolutional layers with the cayley transform. In *International Conference on Learning Representations*, 2021.

Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 1, 2008.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

AM Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

Hoang Tuy. Dc optimization: theory, methods and algorithms. In *Handbook of global optimization*, pages 149–216. Springer, 1995.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Rianne van den Berg, Leonard Hasenclever, Jakub Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. In *proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Vladimir N. Vapnik. *Statistical Learning Theory*. 1998.

J.M. Varah. A lower bound for the smallest singular value of a matrix. *Linear Algebra and its Applications*, 11(1):3–5, 1975. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(75)90112-3. URL `http://www.sciencedirect.com/science/article/pii/0024379575901123`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Alexandre Verine, Yann Chevaleyre, Fabrice Rossi, and benjamin negrevergne. On the expressivity of bi-lipschitz normalizing flows. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021.

Paul Viallard, Eric Guillaume VIDOT, Amaury Habrard, and Emilie Morvant. A pac-bayes analysis of adversarial robustness. *Advances in Neural Information Processing Systems*, 34, 2021.

Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, 2018.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, art. arXiv:1907.10121, Jul 2019.

John Von Neumann. Uber ein okonomsiches gleichungssystem und eine verallgemeinering des browerschen fixpunktsatzes. In *Erge. Math. Kolloq.*, volume 8, pages 73–83, 1937.

James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. A mathematical theory of attention. *arXiv preprint arXiv:2007.02876*, 2020.

Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X. Yu. Orthogonal convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *International Conference on Machine Learning*, pages 5133–5142. PMLR, 2018.

Andre Wibisono. Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In *Conference on Learning Theory*, pages 2093–3027. PMLR, 2018.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Rey Reza Wiyatno and Anqi Xu. Physical adversarial textures that fool visual object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4822–4831, 2019.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295, 2018.

Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J. Zico Kolter. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, 2018.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, 2018.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision*, pages 665–681. Springer, 2020.

Xin Yao and Yong Liu. Fast evolutionary programming. In *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 451–460. MIT Press, 1996.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International Conference on Machine Learning*, pages 7085–7094, 2019.

Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

Man-Chung Yue, Daniel Kuhn, and Wolfram Wiesemann. On linear optimization over wasserstein balls. *arXiv preprint arXiv:2004.07162*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *International conference on Machine Learning*, 2019a.

Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1381–1396. IEEE, 2019b.

Tong Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5(Oct):1225–1251, 2004a.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004b.

Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

Appendices

Appendix A

Black-box adversarial attacks: tiling and evolution strategies

We introduce a new black-box attack achieving state of the art performances. Our approach is based on a new objective function, borrowing ideas from ℓ_∞-white box attacks, and particularly designed to fit derivative-free optimization requirements. It only requires to have access to the logits of the classifier without any other information which is a more realistic scenario. Not only we introduce a new objective function, we extend previous works on black box adversarial attacks to a larger spectrum of evolution strategies and other derivative-free optimization methods. We also highlight a new intriguing property that deep neural networks are not robust to single shot tiled attacks. Our models achieve, with a budget limited to $10,000$ queries, results up to 99.2% of success rate against InceptionV3 classifier with 630 queries to the network on average in the untargeted attacks setting, which is an improvement by 90 queries of the current state of the art. In the targeted setting, we are able to reach, with a limited budget of $100,000$, 100% of success rate with a budget of $6,662$ queries on average, i.e. we need 800 queries less than the current state of the art.

A.1 Introduction

Despite their success, deep learning algorithms have shown vulnerability to adversarial attacks [Biggio et al., 2013, Szegedy et al., 2014], *i.e.* small imperceptible perturbations of the inputs, that lead the networks to misclassify the generated adversarial examples. Since their discovery, adversarial attacks and defenses have become one of the hottest research topics in the machine learning community as serious security issues are raised in many critical fields. They also question our understanding of deep learning behaviors. Although some advances have been made to explain theoretically [Cohen et al., 2019, Fawzi et al., 2016, Pinot et al., 2019, Sinha et al., 2017] and experimentally [Araujo et al., 2019, Goodfellow et al., 2015, Meng and Chen, 2017, Samangouei et al., 2018, Xie et al., 2018] adversarial attacks, the phenomenon remains misunderstood and there is still a gap to come up with principled guarantees on the robustness of neural networks against maliciously crafted attacks. Designing new and stronger attacks helps building better defenses, hence the motivation of our work.

First attacks were generated in a setting where the attacker knows all the information of the network (architecture and parameters). In this *white box* setting, the main idea is to perturb

the input in the direction of the gradient of the loss w.r.t. the input [Carlini and Wagner, 2017, Goodfellow et al., 2015, Kurakin et al., 2016, Moosavi-Dezfooli et al., 2016]. This case is unrealistic because the attacker has only limited access to the network in practice. For instance, web services that propose commercial recognition systems such as Amazon or Google are backed by pretrained neural networks. A user can *query* this system by sending an image to classify. For such a query, the user only has access to the inference results of the classifier which might be either the label, probabilities or logits. Such a setting is coined in the literature as the *black box* setting. It is more realistic but also more challenging from the attacker's standpoint.

As a consequence, several works proposed black box attacks by just querying the inference results of a given classifier. A natural way consists in exploiting the transferability of an adversarial attack, based on the idea that if an example fools a classifier, it is more likely that it fools another one [Papernot et al., 2016a]. In this case, a white box attack is crafted on a fully known classifier. Papernot et al. [2017] exploited this property to derive practical black box attacks. Another approach within the black box setting consists in estimating the gradient of the loss by querying the classifier [Chen et al., 2017, Ilyas et al., 2018a,b]. For these attacks, the PGD attack [Kurakin et al., 2016, Madry et al., 2018] algorithm is used and the gradient is replaced by its estimation.

In this paper, we propose efficient black box adversarial attacks using stochastic derivative free optimization (DFO) methods with only access to the logits of the classifier. By efficient, we mean that our model requires a limited number of queries while outperforming the state of the art in terms of attack success rate. At the very core of our approach is a new objective function particularly designed to suit classical derivative free optimization. We also highlight a new intriguing property that deep neural networks are not robust to single shot tiled attacks. It leverages results and ideas from ℓ_∞-attacks. We also explore a large spectrum of evolution strategies and other derivative-free optimization methods thanks to the Nevergrad framework [Rapin and Teytaud, 2018].

Outline of the paper. We present in Section A.2 the related work on adversarial attacks. Section A.3 presents the core of our approach. We introduce a new generic objective function and discuss two practical instantiations leading to a discrete and a continuous optimization problems. We then give more details on the best performing derivative-free optimization methods, and provide some insights on our models and optimization strategies. Section A.4 is dedicated to a thorough experimental analysis, where we show we reach state of the art performances by comparing our models with the most powerful black-box approaches on both targeted and untargeted attacks. We also assess our models against the most efficient so far defense strategy based on adversarial training. We finally conclude our paper in Section A.5.

A.2 Related work

Adversarial attacks have a long standing history in the machine learning community. Early works appeared in the mid 2000's where the authors were concerned about Spam classification [Biggio et al., 2009]. Szegedy et al. [2014] revives this research topic by highlighting that deep convolutional networks can be easily fooled. Many adversarial attacks against deep neural networks have been proposed since then. One can distinguish two classes of attacks: white box and black box attacks. In the white box setting, the adversary is supposed to have full knowledge of the network (architecture and parameters), while in the black box one, the adversary only has limited access to the network: she does not know the architecture, and can only query the network and gets labels, logits or probabilities from her queries. An attack is said to have

suceeded (we also talk about Attack Success Rate), if the input was originally well classified and the generated example is classified to the targeted label.

The white box setting attracted more attention even if it is the more unrealistic between the two. The attacks are crafted by by back-propagating the gradient of the loss function w.r.t. the input. The problem writes as a non-convex optimization procedure that either constraints the perturbation or aims at minimizing its norm. Among the most popular ones, one can cite FGSM [Goodfellow et al., 2015], PGD [Kurakin et al., 2016, Madry et al., 2018], Deepfool [Moosavi-Dezfooli et al., 2016], JSMA [Papernot et al., 2016b], Carlini&Wagner attack [Carlini and Wagner, 2017] and EAD [Chen et al., 2018a].

The black box setting is more realistic, but also more challenging. Two strategies emerged in the literature to craft attacks within this setting: transferability from a substitute network, and gradient estimation algorithms. Transferability has been pointed out by Papernot et al. [2017]. It consists in generating a white-box adversarial example on a fully known substitute neural network, i.e. a network trained on the same classification task. This crafted adversarial example can be *transferred* to the targeted unknown network. Leveraging this property, Moosavi-Dezfooli et al. [2017] proposed an algorithm to craft a single adversarial attack that is the same for all examples and all networks. Despite the popularity of these methods, gradient estimation algorithms outperform transferability methods. Chen et al. [2017] proposed a variant of the powerful white-box attack introduced in [Carlini and Wagner, 2017], based on gradient estimation with finite differences. This method achieves good results in practice but requires a high number of queries to the network. To reduce the number of queries, Ilyas et al. [2018a] proposed to rely rather on Natural Evolution Strategies (NES). These derivative-free optimization approaches consist in estimating the parametric distribution of the minima of a given objective function. This amounts for most of NES algorithms to perform a natural gradient descent in the space of distributions [Ollivier et al., 2017]. In [Al-Dujaili and O'Reilly, 2019], the authors propose to rather estimate the sign of the gradient instead of estimating the its magnitude suing zeroth-order optimization techniques. They show further how to reduce the search space from exponential to linear. The achieved results were state of the art at the publication date. In Liu et al. [2019], the authors introduced a zeroth-order version of the signSGD algorithm, studied its convergence properties and showed its efficiency in crafting adversarial black-box attacks. The results are promising but fail to beat the state of the art. In Tu et al. [2019], the authors introduce the AutoZOOM framework combining gradient estimation and an auto-encoder trained offline with unlabeled data. The idea is appealing but requires training an auto-encoder with an available dataset, which an additional effort for the attacker. Besides, this may be unrealistic for several use cases. More recently, Moon et al. [2019] proposed a method based on discrete and combinatorial optimization where the perturbations are pushed towards the corners of the ℓ_∞ ball. This method is to the best of our knowledge the state of the art in the black box setting in terms of queries budget and success rate. We will focus in our experiments on this method and show how our approaches achieve better results.

Several defense strategies have been proposed to diminish the impact of adversarial attacks on networks accuracies. A basic workaround, introduced in [Goodfellow et al., 2015], is to augment the learning set with adversarial attacks examples. Such an approach is called adversarial training in the literature. It helps recovering some accuracy but fails to fully defend the network, and lacks theoretical guarantees, in particular principled certificates. Defenses based on randomization at inference time were also proposed [Cohen et al., 2019, Lecuyer et al., 2018, Pinot et al., 2019]. These methods are grounded theoretically, but the guarantees cannot ensure full protection against adversarial examples. The question of defenses and attacks is still widely open since our understanding of this phenomenon is still in its infancy. We evaluate our approach

against adversarial training, the most powerful defense method so far.

A.3 Methods

A.3.1 General framework

Let us consider a classification task $\mathcal{X} \mapsto [K]$ where $\mathcal{X} \subseteq \mathbb{R}^d$ is the input space and $[K] = \{1, ..., K\}$ is the corresponding label set. Let $f : \mathbb{R}^d \to \mathbb{R}^K$ be a classifier (a feed forward neural network in our paper) from an input space $\mathcal{X}$ returning the logits of each label in $[K]$ such that the predicted label for a given input is $\arg\max_{i \in [K]} f_i(x)$. The aim of $||.||_\infty$-bounded untargeted adversarial attacks is, for some input x with label y, to find a perturbation τ such that $\arg\max_{i \in [K]} f_i(x) \neq y$. Classically, $||.||_\infty$-bounded untargeted adversarial attacks aims at optimizing the following objective:

$$\max_{\tau : ||\tau||_\infty \leq \epsilon} L(f(x + \tau), y) \tag{A.1}$$

where L is a loss function (typically the cross entropy) and y the true label. For targeted attacks, the attacker targets a label y_t by maximizing $-L(f(x + \tau), y_t)$. With access to the gradients of the network, gradient descent methods have proved their efficiency [Kurakin et al., 2016, Madry et al., 2018]. So far, the outline of most black box attacks was to estimate the gradient using either finite differences or natural evolution strategies. Here using evolutionary strategies heuristics, we do not want to take care of the gradient estimation problem.

A.3.2 Two optimization problems

In some DFO approaches, the default search space is $\mathbb{R}^d$. In the ℓ_∞ bounded adversarial attacks setting, the search space is $B_\infty(\epsilon) = \{\tau : ||\tau||_\infty \leq \epsilon\}$. It requires to adapt the problem in Eq A.1. Two variants are proposed in the sequel leading to continuous and discretized versions of the problem.

The continuous problem. As in Carlini and Wagner [2017], we use the hyperbolic tangent transformation to restate our problem since $B_\infty(\epsilon) = \epsilon \tanh(\mathbb{R}^d)$. This leads to a continuous search space on which evolutionary strategies apply. Hence our optimization problem writes:

$$\max_{\tau \in \mathbb{R}^d} L(f(x + \epsilon \tanh(\tau)), y). \tag{A.2}$$

We will call this problem $\text{DFO}_c - \text{optimizer}$ where optimizer is the used black box derivative free optimization strategy.

The discretized problem. Moon et al. [2019] pointed out that PGD attacks [Kurakin et al., 2016, Madry et al., 2018] are mainly located on the corners of the ℓ_∞-ball. They consider optimizing the following

$$\max_{\tau \in \{-\epsilon, +\epsilon\}^d} L(f(x + \tau), y). \tag{A.3}$$

The author in [Moon et al., 2019] proposed a purely discrete combinatorial optimization to solve this problem (Eq. A.3). As in Zoph and Le [2017], we here consider how to automatically convert an algorithm designed for continuous optimization to discrete optimization. To make the problem in Eq. A.3 compliant with our evolutionary strategies setting, we rewrite our problem by considering a stochastic function $f(x + \epsilon\tau)$ where, for all i, $\tau_i \in \{-1, +1\}$ and

$\mathbb{P}(\tau_i = 1) = \text{Softmax}(a_i, b_i) = \frac{e^{a_i}}{e^{a_i}+e^{b_i}}$. Hence our problem amounts to find the best parameters a_i and b_i that optimize:

$$\min_{a,b} \mathbb{E}_{\tau \sim \mathbb{P}_{a,b}}(L(f(x + \epsilon\tau), y) \tag{A.4}$$

We then rely on evolutionary strategies to find the parameters a and b. As the optima are deterministic, the optimal values for a and b are at infinity. Some ES algorithms are well suited to such setting as will be discussed in the sequel. We will call this problem $\text{DFO}_d - \text{optimizer}$ where optimizer is the used black box derivative free optimization strategy for a and b. In this case, one could reduce the problem to one variale a_i with $\mathbb{P}(\tau_i = 1) = \frac{1}{1+e^{-a_i}}$, but experimentally the results are comparable, so we concentrate on Problem A.4.

A.3.3 Derivative-free optimization methods

Derivative-free optimization methods are aimed at optimizing an objective function without access to the gradient. There exists a large and wide literature around derivative free optimisation. In this setting, one algorithm aims to minimize some function f on some space $\mathcal{X}$. The only thing that could be done by this algorithm is to query for some points x the value of $f(x)$. As evaluating f can be computationally expensive, the purpose of DFO methods is to get a good approximation of the optima using a moderate number of queries. We tested several evolution strategies [Beyer, 2001, Rechenberg, 1973]: the simple $(1 + 1)$-algorithm [Matyas, 1965, Schumer and Steiglitz, 1968], Covariance Matrix Adaptation (CMA [Hansen and Ostermeier, 2003]). For these methods, the underlying algorithm is to iteratively update some distribution P_θ defined on $\mathcal{X}$. Roughly speaking, the current distribution $\mathbb{P}_\theta$ represents the current belief of the localization of the optimas of the goal function. The parameters are updated using objective function values at different points. It turns out that this family of algorithms, than can be reinterpreted as natural evolution strategies, perform best. The two best performing methods will be detailed in Section A.3.3; we refer to references above for other tested methods.

Our best performing methods: evolution strategies

The $(1 + 1)$-ES algorithm. The $(1 + 1)$-evolution strategy with one-fifth rule [Matyas, 1965, Schumer and Steiglitz, 1968] is a simple but effective derivative-free optimization algorithm (in supplementary material, Alg. 6). Compared to random search, this algorithm moves the center of the Gaussian sampling according to the best candidate and adapts its scale by taking into account their frequency. Yao and Liu [1996] proposed the use of Cauchy distributions instead of classical Gaussian sampling. This favors large steps, and improves the results in case of (possibly partial) separability of the problem, i.e. when it is meaningful to perform large steps in some directions and very moderate ones in the other directions.

CMA-ES algorithm. The Covariance Matrix Adaptation Evolution Strategy [Hansen and Ostermeier, 2003] combines evolution strategies [Beyer, 2001], Cumulative Step-Size Adaptation [Chotard et al., 2012], and a specific method for adaptating the covariance matrix. An outline is provided in supplementary material, Alg. 7. CMA-ES is an effective and robust algorithm, but it becomes catastrophically slow in high dimension due to the expensive computation of the square root of the matrix. As a workaround, Ros and Hansen [2008] propose to approximate the covariance matrix by a diagonal one. This leads to a computational cost linear in the dimension, rather than the original quadratic one.

Link with Natural Evolution Strategy (NES) attacks. Both (1+1)-ES and CMA-ES can be seen as an instantiation of a natural evolution strategy (see for instance Ollivier et al.

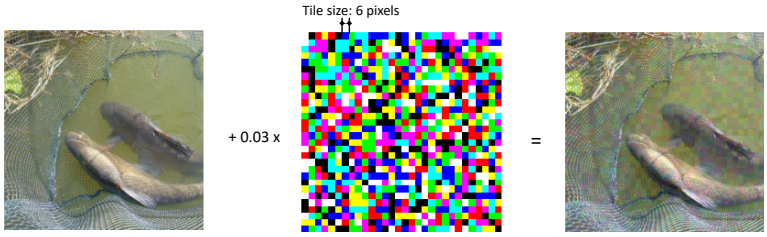


Figure A.1: Illustration of the tiling trick: the same noise is applied on small tile squares.

[2017], Wierstra et al. [2014]). A natural evolution strategy consists in estimating iteratively the distribution of the optima. For most NES approaches, a fortiori CMA-ES, the iterative estimation consists in a second-order gradient descent (also known as natural gradient) in the space of distributions (e.g. Gaussians). (1+1)-ES can also be seen as a NES, where the covariance matrix is restricted to be proportional to the identity. Note however that from an algorithmic perspective, both CME-ES and (1+1)-ES optimize the quantile of the objective function.

Hypotheses for DFO methods in the adversarial attacks context

The state of the art in DFO and intuition suggest the followings. Using softmax for exploring only points in the corner (Eq. A.3) is better for moderate budget, as corners are known to be good adversarial candidates; however, for high precision attacks (with small τ) a smooth continuous precision (Eq A.2) is more relevant. With or without softmax, the optimum is at infinity [1], which is in favor of methods having fast step-size adaptation or samplings with heavy-tail distributions. With an optimum at infinity, [Chotard et al., 2012] has shown how fast is the adaptation of the step-size when using cumulative step-size adaptation (as in CMA-ES), as opposed to slower rates for most methods. Cauchy sampling [Yao and Liu, 1996] in the $(1+1)$-ES is known for favoring fast changes; this is consistent with the superiority of Cauchy sampling in our setting compared to Gaussian sampling.

Newuoa, Powell, SQP, Bayesian Optimization, Bayesian optimization are present in Nevergrad but they have an expensive (budget consumption linear is linear w.r.t. the dimension) initial sampling stage which is not possible in our high-dimensional / moderate budget context. The targeted case needs more precision and favors algorithms such as Diagonal CMA-ES which adapt a step-size per coordinate whereas the untargeted case is more in favor of fast random exploration such as the $(1+1)$-ES. Compared to Diagonal-CMA, CMA with full covariance might be too slow; given a number of queries (rather than a time budget) it is however optimal for high precision.

A.3.4 The tiling trick

Ilyas et al. [2018b] suggested to tile the attack to lower the number of queries necessary to fool the network. Concretely, they observe that the gradient coordinates are correlated for close pixels in the images, so they suggested to add the same noise for small square tiles in the image

[1] i.e. the optima of the ball constrained problem A.1, would be close to the boundary or on the boundary of the ℓ_∞ ball. In that case, the optimum of the continuous problem A.2 will be at ∞ or "close" to it. On the discrete case A.4 it is easy to see that the optimum is when a_i or $b_i \to \infty$.

(see Fig. A.1). We exploit the same trick since it reduces the dimensionality of the search space, and makes hence evolutionary strategies suited to the problem at hand. Besides breaking the curse of dimensionality, tiling leads surprisingly to a new property that we discovered during our experiments. At a given tiling scale, convolutional neural networks are not robust to random noise. Section A.4.2 is devoted to this intriguing property. Interestingly enough, initializing our optimization algorithms with a tiled noise at the appropriate scale drastically speeds up the convergence, leading to a reduced number of queries.

A.4 Experiments

A.4.1 General setting and implementation details

We compare our approach to the "bandits" method [Ilyas et al., 2018b] and the parsimonious attack [Moon et al., 2019]. The latter (parsimonious attack) is, to the best of our knowledge, the state of the art in the black-box setting from the literature; bandits method is also considered in our benchmark given its ties to our models. We reproduced the results from [Moon et al., 2019] in our setting for fair comparison. As explained in section A.3.2, our attacks can be interpreted as ℓ_∞ ones. We use the large-scale ImageNet dataset [Deng et al., 2009]. As usually done in most frameworks, we quantify our success in terms of attack success rate, median queries and average queries. Here, the number of queries refers to the number of requests to the output logits of a classifier for a given image. For the success rate, we only consider the images that were correctly classified by our model. We use InceptionV3 [Szegedy et al., 2017] , VGG16 [Simonyan and Zisserman, 2014] with batch normalization (VGG16bn) and ResNet50 [He et al., 2016] architectures to measure the performance of our algorithm on the ImageNet dataset. These models reach accuracy close to the the state of the art with around $75-80\%$ for the Top-1 accuracy and 95% for the Top-5 accuracy. We use pretrained models from PyTorch [Paszke et al., 2017]. All images are normalized to $[0,1]$. Results on VGG16bn and ResNet50 are deferred in supplementary material A.E. The images to be attacked are selected at random.

We first show that convolutional networks are not robust to tiled random noise, and more surprisingly that there exists an optimal tile size that is the same for all architectures and noise intensities. Then, we evaluate our methods on both targeted and untargeted objectives. We considered the following losses: the cross entropy $L(f(x),y) = -\log(\mathbb{P}(y|x))$ and a loss inspired from the "Carlini&Wagner" attack: $L(f(x),y) = -\mathbb{P}(y|x) + \max_{y'\neq y}\mathbb{P}(y'|x)$ where $\mathbb{P}(y|x) = [\text{Softmax}(f(x))]_y$, the probability for the classifier to classify the input x to label y. The results for the second loss are deferred in supplementary material A.C. For all our attacks, we use the Nevergrad [Rapin and Teytaud, 2018] implementation of evolution strategies. We did not change the default parameters of the optimization strategies.

A.4.2 Convolutional neural networks are not robust to tiled random noise

In this section, we highlight that neural neural networks are not robust to ℓ_∞ tiled random noise. A noise on an image is said to be tiled if the added noise on the image is the same on small squares of pixels (see Figure A.2). In practice, we divide our image in equally sized tiles. For each tile, we add to the image a randomly chosen constant noise: $+\epsilon$ with probability $\frac{1}{2}$ and $-\epsilon$ with probability $\frac{1}{2}$, uniformly on the tile. The tile trick has been introduced inIlyas et al. [2018a] for dimensionality reduction. Here we exhibit a new behavior that we discovered during our experiments. As shown in Fig. A.1 for reasonable noise intensity ($\epsilon = 0.05$), the success rate of a one shot randomly tiled attack is quite high. This fact is observed on many neural network

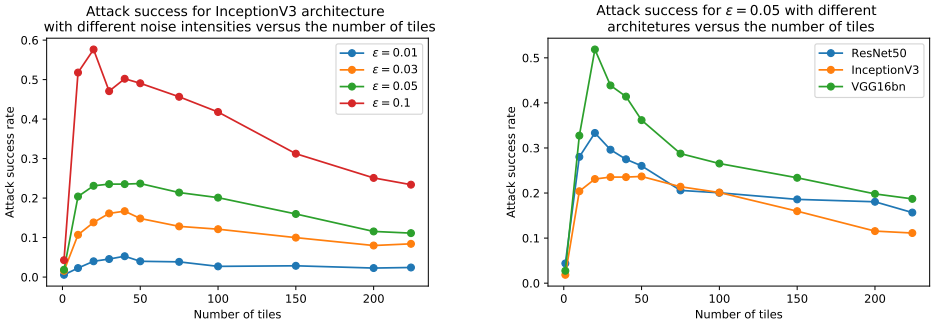


Figure A.2: Success rate of a single shot random attacks on ImageNet vs. the number of tiles used to craft the attack. On the left, attacks are plotted against InceptionV3 classifier with different noise intensities ($\epsilon \in \{0.01, 0.03, 0.05, 0.1\}$). On the right, ϵ is fixed to 0.05 and the single shot attack is evaluated on InceptionV3, ResNet50 and VGG16bn.

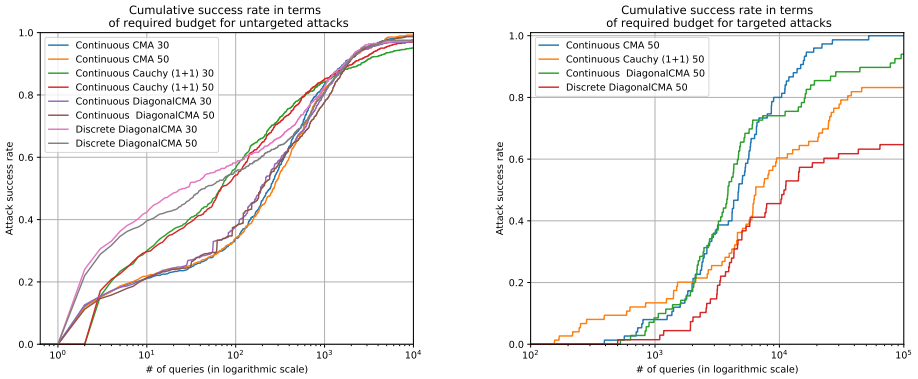


Figure A.3: The cumulative success rate in terms the number of queries for the number of queries required for attacks on ImageNet with $\epsilon = 0.05$ in the untargeted (left) and targeted setting (right). The number of queries (x-axis) is plotted with a logarithmic scale.

architectures. We compared the number of tiles since the images input size are not the same for all architectures ($299 \times 299 \times 3$ for InceptionV3 and $224 \times 224 \times 3$ for VGG16bn and ResNet50). The optimal number of tiles (in the sense of attack success rate) is, surprisingly, independent from the architecture and the noise intensity. We also note that the InceptionV3 architecture is more robust to random tiled noise than VGG16bn and ResNet50 architectures. InceptionV3 blocks are parallel convolutions with different filter sizes that are concatenated. Using different filter sizes may attenuate the effect of the tiled noise since some convolution sizes might be less sensitive. We test this with a single random attack with various numbers of tiles (cf. Figure A.1, A.2). We plotted additional graphs in supplementary material A.B.

A.4.3 Untargeted adversarial attacks

We first evaluate our attacks in the untargeted setting. The aim is to change the predicted label of the classifier. Following [Ilyas et al., 2018b, Moon et al., 2019], we use $10,000$ images that are initially correctly classified and we limit the budget to $10,000$ queries. We experimented with 30

Table A.1: Comparison of our method with the parsimonious and bandits attacks in the untargeted setting on ImageNet on InceptionV3 pretrained network for $\epsilon = 0.05$ and $10,000$ as budget limit.

Method	**# of tiles**	**Average queries**	**Median queries**	**Success rate**
Parsimonious	-	702	222	98.4%
Bandits	30	1007	269	95.3%
Bandits	50	995	249	95.1%
$\text{DFO}_c - \text{Cauchy}(1+1)$-ES	30	466	60	95.2%
$\text{DFO}_c - \text{Cauchy}(1+1)$-ES	50	510	63	97.3%
$\text{DFO}_c - \text{DiagonalCMA}$	30	533	189	97.2%
$\text{DFO}_c - \text{DiagonalCMA}$	50	623	191	98.7%
$\text{DFO}_c - \text{CMA}$	30	589	232	98.9%
$\text{DFO}_c - \text{CMA}$	50	630	259	**99.2%**
$\text{DFO}_d - \text{DiagonalCMA}$	30	**424**	**20**	97.7%
$\text{DFO}_d - \text{DiagonalCMA}$	50	485	38	97.4%

and 50 tiles on the images. Only the best performing methods are reported in Table A.1. We compare our results with [Moon et al., 2019] and [Ilyas et al., 2018b] on InceptionV3 (cf. Table A.1). We also plotted the cumulative success rate in terms of required budget in Figure A.3. We also evaluated our attacks for smaller noise in supplementary material A.D. We achieve results outperforming or at least equal to the state of the art in all cases. More remarkably, We improve by far the number of necessary queries to fool the classifiers. The tiling trick partially explains why the average and the median number of queries are low. Indeed, the first queries of our evolution strategies is in general close to random search and hence, according to the observation of Figs A.1-A.2, the first steps are more likely to fool the network, which explains why the queries budget remains low. This Discrete strategies reach better median numbers of queries - which is consistent as we directly search on the limits of the ℓ_∞-ball; however, given the restricted search space (only corners of the search space are considered), the success rate is lower and on average the number of queries increases due to hard cases.

A.4.4 Targeted adversarial attacks

We also evaluate our methods in the targeted case on ImageNet dataset. We selected $1,000$ images, correctly classified. Since the targeted task is harder than the untargeted case, we set the maximum budget to $100,000$ queries, and $\epsilon = 0.05$. We uniformly chose the target class among the incorrect ones. We evaluated our attacks in comparison with the bandits methods [Ilyas et al., 2018b] and the parsimonious attack [Moon et al., 2019] on InceptionV3 classifier. We also plotted the cumulative success rate in terms of required budget in Figure A.3. CMA-ES beats the state of the art on all criteria. DiagonalCMA-ES obtains acceptable results but is less powerful that CMA-ES in this specific case. The classical CMA optimizer is more precise, even if the run time is much longer. Cauchy $(1+1)$-ES and discretized optimization reach good results, but when the task is more complicated they do not reach as good results as the state of the art in black box targeted attacks.

A.4.5 Untargeted attacks against an adversarially trained network

In this section, we experiment our attacks against a defended network by adversarial training [Goodfellow et al., 2015]. Since adversarial training is computationally expensive, we

Table A.2: Comparison of our method with the parsimonious and bandits attacks in the targeted setting on ImageNet on InceptionV3 pretrained network for $\epsilon = 0.05$ and $100,000$ as budget limit.

Method	**# of tiles**	**Average queries**	**Median queries**	**Success rate**
Parsimonious	-	7184	5116	100%
Bandits	50	25341	18053	92.5%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	9789	6049	83.2%
$\text{DFO}_c - \text{DiagonalCMA}$	50	6768	**3797**	94.0%
$\text{DFO}_c - \text{CMA}$	50	**6662**	4692	**100%**
$\text{DFO}_d - \text{DiagonalCMA}$	50	8957	4619	64.2%

restricted ourselves to the CIFAR10 dataset [Krizhevsky et al., 2009] for this experiment. Image size is $32 \times 32 \times 3$. We adversarially trained a WideResNet28x10 [Zagoruyko and Komodakis, 2016] with PGD ℓ_∞ attacks [Kurakin et al., 2016, Madry et al., 2018] of norm $8/256$ and 10 steps of size $2/256$. In this setting, we randomly selected $1,000$ images, and limited the budget to $20,000$ queries. We ran PGD ℓ_∞ attacks [Kurakin et al., 2016, Madry et al., 2018] of norm $8/256$ and 20 steps of size $1/256$ against our network, and achieved a success rate up to 36%, which is the the state of the art in the white box setting. We also compared our method to the Parsimonious and bandit attacks. Results are reported in Appendix A.6. On this task, the parsimonious attack method is slightly better than our best approach.

A.5 Conclusion

In this paper, we proposed a new framework for crafting black box adversarial attacks based on derivative free optimization. Because of the high dimensionality and the characteristics of the problem (see Section A.3.3), not all optimization strategies give satisfying results. However, combined with the tiling trick, evolutionary strategies such as CMA, DiagonalCMA and Cauchy (1+1)-ES beats the current state of the art in both targeted and untargeted settings. In particular, $\text{DFO}_c - \text{CMA}$ improves the state of the art in terms of success rate in almost all settings. We also validated the robustness of our attack against an adversarially trained network. Future work will be devoted to better understanding the intriguing property of the effect that a neural network is not robust to a one shot randomly tiled attack.

A.A Algorithms

A.A.1 The (1+1)-ES algorithm

A.A.2 CMA-ES algorithm

Algorithm 6: The $(1+1)$ Evolution Strategy.

Require: Function $f : \mathbb{R}^d \to \mathbb{R}$ to minimize
  $m \leftarrow 0$, $C \leftarrow \boldsymbol{I}_d$, $\sigma \leftarrow 1$
  for $t = 1...n$ **do**
    (Generate candidates)
    Generate $m' \sim m + \sigma X$ where X is sampled from a Cauchy or Gaussian distribution.
    if $f(m') \leq f(m)$ **then**
      $m \leftarrow m'$, $\sigma \leftarrow 2\sigma$
    else
      $\sigma \leftarrow 2^{-\frac{1}{4}}\sigma$
    end if
  end for

Algorithm 7: CMA-ES algorithm. The T subscript denotes transposition.

Require: Function $f : \mathbb{R}^d \to \mathbb{R}$ to minimize, parameters b, c, $w_1 > \ldots, w_\mu > 0$, p_c and others as in e.g. [Hansen and Ostermeier, 2003].
  $m \leftarrow 0$, $C \leftarrow \boldsymbol{I}_d$, $\sigma \leftarrow 1$
  for $t = 1...n$ **do**
    Generate $x_1, ..., x_\lambda \sim m + \sigma\mathcal{N}(0, C)$.
    Define x'_i the i^{th} best of the x_i.
    Update the cumulation for C: $p_c \leftarrow$ cumulation of p_c, overall direction of progress.
    Update the covariance matrix:

$$C \leftarrow (1-c)\underbrace{C}_{inertia} + \frac{c}{b}\underbrace{(p_c \times p_c^T)}_{\text{overall direction}} + c(1-\frac{1}{b})\sum_{i=1}^{\mu} w_i \underbrace{\frac{x'_i - m}{\sigma} \times \frac{(x'_i - m)^T}{\sigma}}_{\text{“covariance” of the } \frac{1}{\sigma}x'_i}$$

    Update mean:

$$m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda}$$

    Update σ by cumulative step-size adaptation [Chotard et al., 2012].
  end for

A.B Additional plots for the tiling trick

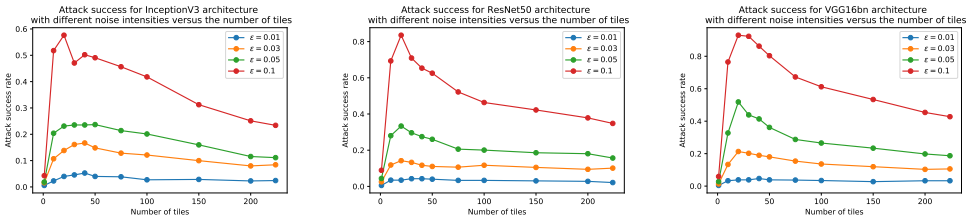


Figure A.4: Random attack success rate against InceptionV3 (left), ResNet50 (center), VGG16bn (right) for different noise intensities. We just randomly draw one tiled attack and check if it is successful.

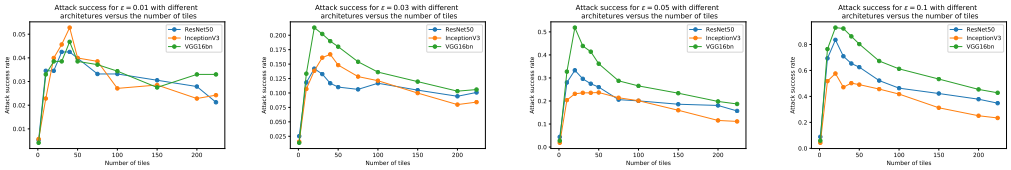


Figure A.5: Random attack success rate for different noise intensities $\epsilon \in \{0.01, 0.03, 0.05, 0.1\}$ (from right to left) against different architectures. We just randomly draw one tiled attack and check if it is successful.

A.C Results with "Carlini&Wagner" loss

In this section, we follow the same experimental setup as in Section A.4.3, but we built our attacks with the "Carlini&Wagner" loss instead of the cross entropy. We remark the results are comparable and similar.

Table A.3: Comparison of our method with"Carlini&Wagner" loss versus the parsimonious and bandits attacks in the untargeted setting on InceptionV3 pretrained network for $\epsilon = 0.05$ and $10,000$ as budget limit.

Method	**# of tiles**	**Average queries**	**Median queries**	**Success rate**
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	30	353	57	97.2%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	**347**	63	98.8%
$\text{DFO}_c - \text{DiagonalCMA}$	30	483	167	98.8%
$\text{DFO}_c - \text{DiagonalCMA}$	50	528	181	99.2%
$\text{DFO}_c - \text{CMA}$	30	475	225	99.2%
$\text{DFO}_c - \text{CMA}$	50	491	246	**99.4%**
$\text{DFO}_d - \text{DiagonalCMA}$	30	482	**27**	98.0%
$\text{DFO}_d - \text{DiagonalCMA}$	50	510	37	98.0%

A.D Untargeted attacks with smaller noise intensities

We evaluated our method on smaller noise intensities ($\epsilon \in \{0.01, 0.03, 0.05\}$) in the untargeted setting on ImageNet dataset. In this framework, we also picked up randomly $10,000$ images and limited our budget to $10,000$ queries. We compared to the bandits method [Ilyas et al., 2018b] and to the parsimonious attack [Moon et al., 2019] on InceptionV3 network. We limited our experiments to a number of tiles of 50. We report our results in Table A.4. We remark our attacks reach state of the art for $\epsilon = 0.03$ and $\epsilon = 0.05$ both in terms of success rate and queries budget. For $\epsilon = 0.01$, we reach results comparable to the state of the art.

Table A.4: Results of our method compared to the parsimonious and bandit attacks in the untargeted setting on InceptionV3 pretrained network for different values of noise intensities $\epsilon \in \{0.01, 0.03, 0.05\}$ and a maximum of $10,000$ queries.

ϵ	**Method**	**# of tiles**	**Avg. queries**	**Med. queries**	**Success rate**
0.05	Parsimonious	-	722	237	98.5%
	Bandits	50	995	249	95.1%
	$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	510	63	97.3%
	$\text{DFO}_c - \text{DiagonalCMA}$	50	623	191	98.7%
	$\text{DFO}_c - \text{CMA}$	50	630	259	**99.2%**
	$\text{DFO}_d - \text{DiagonalCMA}$	50	**485**	**38**	97.4%
0.03	Parsimonious	-	1104	392	95.7%
	Bandits	50	1376	466	92.7%
	$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	846	**203**	93,2%
	$\text{DFO}_c - \text{DiagonalCMA}$	50	971	429	96,5%
	$\text{DFO}_c - \text{CMA}$	50	911	404	**96.7%**
	$\text{DFO}_d - \text{DiagonalCMA}$	50	**799**	293	94,1%
0.01	Parsimonious	-	2104	1174	80.3%
	Bandits	50	2018	992	72.9%
	$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	1668	**751**	72,1%
	$\text{DFO}_c - \text{DiagonalCMA}$	50	1958	1175	79.2%
	$\text{DFO}_c - \text{CMA}$	50	1921	1107	**80.4%**
	$\text{DFO}_d - \text{DiagonalCMA}$	50	**1188**	849	71,3%

A.E Untargeted attacks against other architectures

We also evaluated our method on different neural networks architectures. For each network we randomly selected $10,000$ images that were correctly classified. We limit our budget to $10,000$ queries and set the number of tiles to 50. We achieve a success attack rate up to 100% on every classifier with a budget as low as 8 median queries for the VGG16bn for instance (see Table A.5). One should notice that the performances are lower on InceptionV3 as it is also reported for the bandit methods in [Ilyas et al., 2018b]. This possibly due to the fact that the tiling trick is less relevant on the Inception network than on the other networks (see Fig. A.2).

Table A.5: Comparison of our method on the ImageNet dataset with InceptionV3 (I), ResNet50 (R) and VGG16bn (V) for $\epsilon = 0.05$ and $10,000$ as budget limit.

Method	**Tile size**	**Avg queries**			**Med. queries**			**Succ. Rate**		
		I	R	V	I	R	V	I	R	V
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	30	466	**163**	86	60	**19**	8	95.2%	99.6%	**100%**
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	510	218	**67**	63	32	4	97.3%	99.6%	99.7%
$\text{DFO}_c - \text{DiagonalCMA}$	30	533	263	174	189	95	55	97.2%	99.0%	99.9%
$\text{DFO}_c - \text{DiagonalCMA}$	50	623	373	227	191	121	71	98.7%	99.9%	**100%**
$\text{DFO}_c - \text{CMA}$	30	588	256	176	232	138	72	98.9%	99.9%	99.9%
$\text{DFO}_c - \text{CMA}$	50	630	270	219	259	143	107	**99.2%**	**100%**	99.9%
$\text{DFO}_d - \text{DiagonalCMA}$	50	485	617	345	38	62	6	97.4%	99.2%	99.6%
$\text{DFO}_d - \text{DiagonalCMA}$	30	**424**	417	211	**20**	20	**2**	97.7%	98.8%	99.5%

A.F Table for attacks against adversarially tranined network

Table A.6: Adversarial attacks against an adversarially trained WideResnet28x10 network on CIFAR10 dataset for $\epsilon = 0.03125$ and $20,000$ as budget limit.

Method	**# of tiles**	**Average queries**	**Median queries**	**Success rate**
PGD (not black-box)	-	20	20	36%
Parsimonious	-	1130	450	**42%**
Bandits	10	1429	530	29.1%
Bandits	20	1802	798	33.8%
Bandits	32	1993	812	34.8%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	10	429	**60**	29.5%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	20	902	93	30.5%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	32	1865	764	31.7%
$\text{DFO}_c - \text{DiagonalCMA}$	10	395	85	30.5%
$\text{DFO}_c - \text{DiagonalCMA}$	20	624	151	31.3%
$\text{DFO}_c - \text{DiagonalCMA}$	32	1379	860	34.7%
$\text{DFO}_c - \text{CMA}$	10	**363**	156	30.4%
$\text{DFO}_c - \text{CMA}$	20	1676	740	40.2%
$\text{DFO}_c - \text{CMA}$	32	2311	1191	40.2%

A.G Failing methods

In this section, we compare our attacks to other optimization strategies. We run our experiments in the same setup as in Section A.4.3. Results are reported in Table A.7. DE and Normal (1+1)-ES performs poorly, probably because these optimization strategies converge slower when the optima are at "infinity". We reformulate this sentence accordingly in the updated version of the paper. Finally, as the initialization of Powell is linear with the dimension and with less variance, it performs poorer than simple random search. Newuoa, SQP and Cobyla algorithms have also been tried on a smaller number images (we did not report the results), but their initialization is also linear in the dimension, so they reach very poor results too.

Table A.7: Comparison with other DFO optimization strategies in the untargeted setting on ImageNet dataset InceptionV3 pretrained network for $\epsilon = 0.05$ and $10,000$ as budget limit.

Method	**# of tiles**	**Average queries**	**Median queries**	**Success rate**
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	30	466	60	95.2%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	510	63	97.3%
$\text{DFO}_c - \text{DiagonalCMA}$	30	533	189	97.2%
$\text{DFO}_c - \text{DiagonalCMA}$	50	623	191	98.7%
$\text{DFO}_c - \text{CMA}$	30	589	232	98.9%
$\text{DFO}_c - \text{CMA}$	50	630	259	99.2%
$\text{DFO}_c - \text{DE}$	30	756	159	78.8%
$\text{DFO}_c - \text{DE}$	50	699	149	76.0%
$\text{DFO}_c - \text{Normal}(1+1)\text{-ES}$	30	581	45	87.6%
$\text{DFO}_c - \text{Normal}(1+1)\text{-ES}$	50	661	66	92.8%
$\text{DFO}_c - \text{RandomSearch}$	30	568	6	37.9%
$\text{DFO}_c - \text{RandomSearch}$	50	527	5	38.2%
$\text{DFO}_c - \text{Powell}$	30	4889	5332	14.4%
$\text{DFO}_c - \text{Powell}$	50	4578	4076	7.3%

Appendix B

Advocating for Multiple Defense Strategies against Adversarial Examples

It has been empirically observed that defense mechanisms designed to protect neural networks against ℓ_∞ adversarial examples offer poor performance against ℓ_2 adversarial examples and vice versa. In this paper we conduct a geometrical analysis that validates this observation. Then, we provide a number of empirical insights to illustrate the effect of this phenomenon in practice. Then, we review some of the existing defense mechanism that attempts to defend against multiple attacks by mixing defense strategies. Thanks to our numerical experiments, we discuss the relevance of this method and state open questions for the adversarial examples community.

B.1 Introduction

Deep neural networks achieve state-of-the-art performances in a variety of domains such as natural language processing Radford et al. [2018], image recognition He et al. [2016] and speech recognition Hinton et al. [2012]. However, it has been shown that such neural networks are vulnerable to *adversarial examples*, *i.e.*, imperceptible variations of the natural examples, crafted to deliberately mislead the models Biggio et al. [2013], Globerson et al. [2006], Szegedy et al. [2014]. Since their discovery, a variety of algorithms have been developed to generate adversarial examples (a.k.a. attacks), for example FGSM [Goodfellow et al., 2015], PGD [Madry et al., 2018] and C&W [Carlini and Wagner, 2017], to mention the most popular ones.

Because it is difficult to characterize the space of visually imperceptible variations of a natural image, existing adversarial attacks use surrogates that can differ from one attack to another. For example, Goodfellow et al. [2015] use the ℓ_∞ norm to measure the distance between the original image and the adversarial image whereas Carlini and Wagner [2017] use the ℓ_2 norm. When the input dimension is low, the choice of the norm is of little importance because the ℓ_∞ and ℓ_2 balls overlap by a large margin, and the adversarial examples lie in the same space. An important insight in this paper is to observe that the overlap between the two balls diminishes exponentially quickly as the dimensionality of the input space increases. For typical image datasets with large dimensionality, the two balls are mostly disjoint. As a consequence, the

ℓ_∞ and the ℓ_2 adversarial examples lie in different areas of the space, and it explains why ℓ_∞ defense mechanisms perform poorly against ℓ_2 attacks and vice versa.

Building on this insight, we advocate for designing models that incorporate defense mechanisms against both ℓ_∞ and ℓ_2 attacks and review several ways of mixing existing defense mechanisms. In particular, we evaluate the performance of *Mixed Adversarial Training* (MAT) Goodfellow et al. [2015] which consists of augmenting training batches using *both* ℓ_∞ and ℓ_2 adversarial examples, and *Randomized Adversarial Training* (RAT) Salman et al. [2019], a solution to benefit from the advantages of both ℓ_∞ adversarial training, and ℓ_2 randomized defense.

Outline. The rest is organized as follows. In Section B.2, we recall the principle of existing attacks and defense mechanisms. In Section B.3, we conduct a theoretical analysis to show why the ℓ_∞ defense mechanisms cannot be robust against ℓ_2 attacks and vice versa. We then corroborate this analysis with empirical results using real adversarial attacks and defense mechanisms. In Section B.4, we discuss various strategies to mix defense mechanisms, conduct comparative experiments, and discuss the performance of each strategy.

B.2 Preliminaries on Adversarial Attacks and Defenses

Let us first consider a standard classification task with an input space $\mathcal{X} = [0,1]^d$ of dimension d, an output space $\mathcal{Y} = [K]$ and a data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. We assume the model f_θ has been trained to minimize the expectation over $\mathcal{D}$ of a loss function $\mathcal{L}$ as follows:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}(f_\theta(x), y)\right]. \tag{B.1}$$

B.2.1 Adversarial attacks

Given an input-output pair $(x, y) \sim \mathcal{D}$, an *adversarial attack* is a procedure that produces a small perturbation $\tau \in \mathcal{X}$ such that $f_\theta(x + \tau) \neq y$. To find the best perturbation τ, existing attacks can adopt one of the two following strategies: (i) maximizing the loss $\mathcal{L}(f_\theta(x + \tau), y)$ under some constraint on $\|\tau\|_p$[1] (a.k.a. loss maximization); or (ii) minimizing $\|\tau\|_p$ under some constraint on the loss $\mathcal{L}(f_\theta(x + \tau), y)$ (a.k.a. perturbation minimization).

(i) Loss maximization. In this scenario, the procedure maximizes the loss objective function, under the constraint that the ℓ_p norm of the perturbation remains bounded by some value ϵ, as follows:

$$\underset{\|\tau\|_p \leq \epsilon}{\operatorname{argmax}} \mathcal{L}(f_\theta(x + \tau), y). \tag{B.2}$$

The typical value of ϵ depends on the norm $\|\cdot\|_p$ considered in the problem setting. In order to compare ℓ_∞ and ℓ_2 attacks of similar strength, we choose values of ϵ_∞ and ϵ_2 (for ℓ_∞ and ℓ_2 norms respectively) which result in ℓ_∞ and ℓ_2 balls of equivalent volumes. For the particular case of CIFAR-10, this would lead us to choose $\epsilon_\infty = 0.03$ and $\epsilon_2 = 0.8$ which correspond to the maximum values chosen empirically to avoid the generation of visually detectable perturbations. The current state-of-the-art method to solve Problem (B.2) is based on a projected gradient

[1] with $p \in \{0, \cdots, \infty\}$.

descent (PGD) Madry et al. [2018] of radius ϵ. Given a budget ϵ, it recursively computes

$$x^{t+1} = \prod_{B_p(x,\epsilon)} \left(x^t + \alpha \underset{\delta \text{ s.t. } ||\delta||_p \leq 1}{\operatorname{argmax}} \left(\Delta^t | \delta \right) \right) \tag{B.3}$$

where $B_p(x, \epsilon) = \{x + \tau \text{ s.t. } \|\tau\|_p \leq \epsilon\}$, $\Delta^t = \nabla_x \mathcal{L}\left(f_\theta\left(x^t\right), y\right)$, α is a gradient step size, and $\prod_S$ is the projection operator on S. Both PGD attacks with $p = 2$, and $p = \infty$ are currently used in the literature as state-of-the-art attacks for the loss maximization problem.

(ii) Perturbation minimization. This type of procedure search for the perturbation that has the minimal ℓ_p norm, under the constraint that $\mathcal{L}(f_\theta(x+\tau), y)$ is bigger than a given bound c:

$$\underset{\mathcal{L}(f_\theta(x+\tau),y) \geq c}{\operatorname{argmin}} \|\tau\|_p. \tag{B.4}$$

The value of c is typically chosen depending on the loss function $\mathcal{L}$[2]. Problem (B.4) has been tackled in Carlini and Wagner [2017], leading to the following method, denoted C&W attack in the rest of this appendix. It aims at solving the following Lagrangian relaxation of Problem (B.4):

$$\underset{\tau}{\operatorname{argmin}} \|\tau\|_p + \lambda \times g(x + \tau) \tag{B.5}$$

where $g(x + \tau) < 0$ if and only if $\mathcal{L}(f_\theta(x + \tau), y) \geq c$. The authors use a change of variable $\tau = \tanh(w) - x$ to ensure that $-1 \leq x + \tau \leq 1$, a binary search to optimize the constant c, and Adam or SGD to compute an approximated solution. The C&W attack is well defined both for $p = 2$, and $p = \infty$, but there is a clear empirical gap of efficiency in favor of the ℓ_2 attack.

In this appendix, we focus on the *Loss Maximization* setting using the PGD attack. However we conduct some of our experiments using *Perturbation Minimization* algorithms such as C&W to capture more detailed information about the location of adversarial examples in the vector space[3].

B.2.2 Defense mechanisms

Adversarial Training (AT). Adversarial Training was introduced in Goodfellow et al. [2015] and later improved in Madry et al. [2018] as a first defense mechanism to train robust neural networks. It consists in augmenting training batches with adversarial examples generated during the training procedure. The standard training procedure from Equation (B.1) is thus replaced by the following min max problem, where the classifier tries to minimize the expected loss under maximum perturbation of its input:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\tau\|_p \leq \epsilon} \mathcal{L}\left(f_\theta(x+\tau), y\right) \right]. \tag{B.6}$$

In the case where $p = \infty$, this technique offers good robustness against ℓ_∞ attacks Athalye et al. [2018a]. AT can also be used with ℓ_2 attacks but as we will discuss in Section B.3, AT with one norm offers poor protection against the other. The main weakness of Adversarial Training is its lack of formal guarantees. Despite some recent work providing great insights Sinha et al. [2017], Zhang et al. [2019a], there is no worst case lower bound yet on the accuracy under attack of this method.

[2]For example, if $\mathcal{L}$ is the 0/1 loss, any $c > 0$ is acceptable.

[3]As it has a more flexible geometry than the *Loss Maximization* attacks.

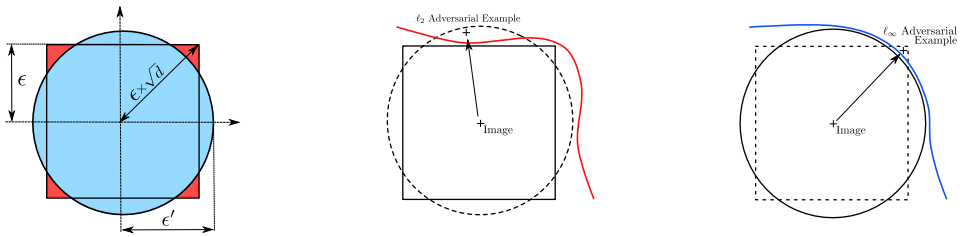
Noise injection mechanisms (NI). Another important technique to defend against adversarial examples is to use Noise Injection. In contrast with Adversarial Training, Noise Injection mechanisms are usually deployed after training. In a nutshell, it works as follows. At inference time, given a unlabeled sample x, the network outputs

$$\tilde{f}_\theta(x) := f_\theta(x + \eta) \quad \text{(instead of } f_\theta(x)\text{)} \tag{B.7}$$

where η is a random variable on $\mathbb{R}^d$. Even though, Noise Injection is often less efficient than Adversarial Training in practice (see *e.g.*, Table B.3), it benefits from strong theoretical background. In particular, recent work Lecuyer et al. [2018], followed by Cohen et al. [2019], Pinot et al. [2019] demonstrated that noise injection from a Gaussian distribution can give provable defense against ℓ_2 adversarial attacks. In this work, besides the classical Gaussian noises already investigated in previous works, we evaluate the efficiency of Uniform distributions to defend against ℓ_2 adversarial examples.

B.3 No Free Lunch for Adversarial Defenses

In this Section, we show both theoretically and empirically that defenses mechanisms intending to defend against ℓ_∞ attacks cannot provide suitable defense against ℓ_2 attacks. Our reasoning is perfectly general; hence we can similarly demonstrate the reciprocal statement, but we focus on this side for simplicity.



(a) (b) (c)

Figure B.1: Left: 2D representation of the ℓ_∞ and ℓ_2 balls of respective radius ϵ and ϵ'. Middle: a classifier trained with ℓ_∞ adversarial perturbations (materialized by the red line) remains vulnerable to ℓ_2 attacks. Right: a classifier trained with ℓ_2 adversarial perturbations (materialized by the blue line) remains vulnerable to ℓ_∞ attacks.

B.3.1 Theoretical analysis

Let us consider a classifier f_∞ that is provably robust against adversarial examples with maximum ℓ_∞ norm of value ϵ_∞. It guarantees that for any input-output pair $(x, y) \sim \mathcal{D}$ and for any perturbation τ such that $\|\tau\|_\infty \leq \epsilon_\infty$, f_∞ is not misled by the perturbation, *i.e.*, $f_\infty(x + \tau) = f_\infty(x)$. We now focus our study on the performance of this classifier against adversarial examples bounded with a ℓ_2 norm of value ϵ_2. Using Figure B.1(a), we observe that any ℓ_2 adversarial example that is also in the ℓ_∞ ball, will not fool f_∞. Conversely, if it is outside the ball, we have no guarantee.

To characterize the probability that such an ℓ_2 perturbation fools an ℓ_∞ defense mechanism in the general case (*i.e.*, any dimension d), we measure the ratio between the volume of the intersection of the ℓ_∞ ball of radius ϵ_∞ and the ℓ_2 ball of radius ϵ_2. As Theorem 15 shows, this

ratio depends on the dimensionality d of the input vector x, and rapidly converges to zero when d increases. Therefore a defense mechanism that protects against all ℓ_∞ bounded adversarial examples is unlikely to be efficient against ℓ_2 attacks.

Theorem 15 (Probability of the intersection goes to 0)**.** *Let* $B_{2,d}(\epsilon) := \left\{\tau \in \mathbb{R}^d \ s.t \ \|\tau\|_2 \leq \epsilon\right\}$ *and* $B_{\infty,d}(\epsilon') := \left\{\tau \in \mathbb{R}^d \ s.t \ \|\tau\|_\infty \leq \epsilon'\right\}$*. If for all d, we select* ϵ *and* ϵ*' such that* $\mathrm{Vol}\left(B_{2,d}(\epsilon)\right) = \mathrm{Vol}\left(B_{\infty,d}(\epsilon')\right)$*, then*

$$\frac{\mathrm{Vol}\left(B_{2,d}(\epsilon) \bigcap B_{\infty,d}(\epsilon')\right)}{\mathrm{Vol}\left(B_{\infty,d}(\epsilon')\right)} \to 0 \ \textit{when} \ d \to \infty.$$

Proof. Without loss of generality, let us fix $\epsilon = 1$. One can show that for all d,

$$\mathrm{Vol}\left(B_{2,d}\left(\frac{2}{\sqrt{\pi}}\Gamma\left(\frac{d}{2}+1\right)^{1/d}\right)\right) = \mathrm{Vol}\left(B_{\infty,d}\left(1\right)\right) \tag{B.8}$$

where Γ is the gamma function. Let us denote

$$r_2(d) = \frac{2}{\sqrt{\pi}}\Gamma\left(\frac{d}{2}+1\right)^{1/d}. \tag{B.9}$$

Then, thanks to Stirling's formula

$$r_2(d) \sim \sqrt{\frac{2}{\pi e}} d^{1/2}. \tag{B.10}$$

Finally, if we denote $\mathcal{U}_S$, the uniform distribution on set S, by using Hoeffding inequality between Equation B.14 and B.15, we get:

$$\frac{\mathrm{Vol}(B_{2,d}(r_2(d)) \bigcap B_{\infty,d}(1))}{\mathrm{Vol}(B_{\infty,d}(1))} \tag{B.11}$$

$$= \mathbb{P}_{x\sim\mathcal{U}_{B_{\infty,d}(1)}}\left[x \in B_{2,d}(r_2(d))\right] \tag{B.12}$$

$$= \mathbb{P}_{x\sim\mathcal{U}_{B_{\infty,d}(1)}}\left[\textstyle\sum_{i=1}^{d} |x_i|^2 \leq r_2^2(d)\right] \tag{B.13}$$

$$\leq \exp -d^{-1}\left(r_2^2(d) - d\mathbb{E}|x_1|^2\right)^2 \tag{B.14}$$

$$\leq \exp -\left(\frac{2}{\pi e} - \frac{1}{3}\right)^2 d + o(d). \tag{B.15}$$

Then the ratio between the volume of the intersection of the ball and the volume of the ball converges towards 0 when d goes to ∞. □

Theorem 15 states that, when d is large enough, ℓ_2 bounded perturbations have a null probability of being also in the ℓ_∞ ball of the same volume. As a consequence, for any value of d that is large enough, a defense mechanism that offers full protection against ℓ_∞ adversarial examples is not guaranteed to offer any protection against ℓ_2 attacks[4].

Note that this result defeats the 2-dimensional intuition: if we consider a 2 dimensional problem setting, the ℓ_∞ and the ℓ_2 balls have an important overlap (as illustrated in Figure B.1(a)) and the probability of sampling at the intersection of the two balls is bounded by approximately 98%. However, as we increase the dimensionality d, this probability quickly becomes negligible,

[4]Th. 15 can easily be extended to any two balls with different norms. For clarity, we restrict to the case of ℓ_∞ and ℓ_2 norms.

Table B.1: Bounds of Theorem 15 on the volume of the intersection of ℓ_2 and ℓ_∞ balls at equal volume for typical image classification datasets. When $d = 2$, the bound is $10^{-0.009} \approx 0.98$.

Dataset	**Dim. (d)**	**Vol. of the intersection**
–	2	$10^{-0.009}$ (≈ 0.98)
MNIST	784	10^{-144}
CIFAR	3072	10^{-578}
ImageNet	150528	10^{-28946}

even for very simple image datasets such as MNIST. An instantiation of the bound for classical image datasets is presented in Table B.1. The probability of sampling at the intersection of the ℓ_∞ and ℓ_2 balls is close to zero for any realistic image setting. In large dimensions, the volume of the corner of the ℓ_∞ ball is much bigger than it appears in Figure B.1(a).

B.3.2 No Free Lunch in Practice

Our theoretical analysis shows that if adversarial examples were uniformly distributed in a high-dimensional space, then any mechanism that perfectly defends against ℓ_∞ adversarial examples has a null probability of protecting against ℓ_2-bounded adversarial attacks. Although existing defense mechanisms do not necessarily assume such a distribution of adversarial examples, we demonstrate that whatever distribution they use, it offers no favorable bias with respect to the result of Theorem 15. As we discussed in Section B.2, there are two distinct attack settings: loss maximization (PGD) and perturbation minimization (C&W). Our analysis is mainly focusing on loss maximization attacks. However, these attacks have a very strict geometry[5]. This is why, to present a deeper analysis of the behavior of adversarial attacks and defenses, we also present a set of experiments that use perturbation minimization attacks.

Table B.2: Average norms of PGD-ℓ_2 and PGD-ℓ_∞ adversarial examples with and without ℓ_∞ adversarial training on CIFAR-10 ($d = 3072$).

	Attack PGD-ℓ_2		Attack PGD-ℓ_∞	
	Unprotected	AT-ℓ_∞	Unprotected	AT-ℓ_2
Average ℓ_2 norm	0.830	0.830	1.400	1.640
Average ℓ_∞ norm	0.075	0.200	0.031	0.031

Adversarial training vs. loss maximization attacks To demonstrate that ℓ_∞ adversarial training is not robust against PGD-ℓ_2 attacks we measure the evolution of ℓ_2 norm of adversarial examples generated with PGD-ℓ_∞ between an unprotected model and a model trained with AT-ℓ_∞, *i.e.*, AT where adversarial examples are generated with PGD-ℓ_∞[6]. Results are presented in Table B.2. [7]

[5]Due to the projection operator, all PGD attacks saturate the constraint, which makes them all lies in a very small part of the ball.

[6]To do so, we use the same experimental setting as in Section B.4 with ϵ_∞ and ϵ_2 such that the volumes of the two balls are equal.

[7]All experiments in this section are conducted on CIFAR-10, and the experimental setting is fully detailed in Section B.4.1.

The analysis is unambiguous: the average ℓ_∞ norm of a bounded ℓ_2 perturbation more than double between an unprotected model and a model trained with AT PGD-ℓ_∞. This phenomenon perfectly reflects the illustration of Figure B.1 (c). The attack will generate an adversarial example on the corner of the ℓ_∞ ball thus increasing the ℓ_∞ norm while maintaining the same ℓ_2 norm. We can observe the same phenomenon with AT-ℓ_2 against PGD-ℓ_∞ attack (see Figure B.1 (b) and Table B.2). PGD-ℓ_∞ attack increases the ℓ_2 norm while maintaining the same ℓ_∞ perturbation thus generating the perturbation in the upper area.

As a consequence, we cannot expect adversarial training ℓ_∞ to offer any guaranteed protection against ℓ_2 adversarial examples .

Adversarial training vs. perturbation minimization attacks. To better capture the behavior of ℓ_2 adversarial examples, we now study the performances of an ℓ_2 perturbation minimization attack (C&W) with and without AT-ℓ_∞. It allows us to understand in which area C&W discovers adversarial examples and the impact of AT-ℓ_∞. In high dimensions, the red corners (see Figure B.1 (a)) are very far away from the ℓ_2 ball. Therefore, we hypothesize that a large proportion of the ℓ_2 adversarial examples will remain unprotected. To validate this assumption, we measure the proportion of adversarial examples inside of the ℓ_2 ball before and after ℓ_∞ adversarial training. The results are presented in Figure B.2 (left: without adversarial training, right: with adversarial training).

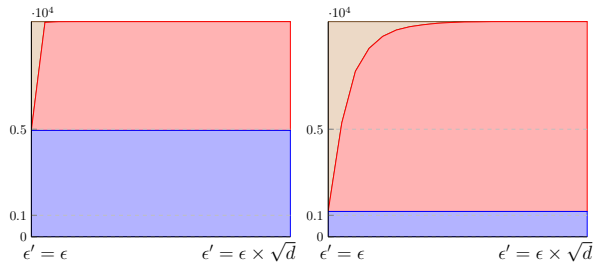


Figure B.2: Comparison of the number of adversarial examples found by C&W, inside the ℓ_∞ ball (lower, blue area), outside the ℓ_∞ ball but inside the ℓ_2 ball (middle, red area) and outside the ℓ_2 ball (upper gray area). ϵ is set to 0.3 and ϵ' varies along the x-axis. Left: without adversarial training, right: with adversarial training. Most adversarial examples have shifted from the ℓ_∞ ball to the cap of the ℓ_2 ball, but remain at the same ℓ_2 distance from the original example.

On both charts, the blue area represents the proportion of adversarial examples that are inside the ℓ_∞ ball. The red area represents the adversarial examples that are outside the ℓ_∞ ball but still inside the ℓ_2 ball (valid ℓ_2 adversarial examples). Finally, the brown-beige area represents the adversarial examples that are beyond the ℓ_2 bound. The radius ϵ' of the ℓ_2 ball varies along the x-axis from ϵ' to $\epsilon'\sqrt{d}$. On the left chart (without adversarial training) most ℓ_2 adversarial examples generated by C&W are inside both balls. On the right chart most of the adversarial examples have been shifted out the ℓ_∞ ball. This is the expected consequence of ℓ_∞ adversarial training. However, these adversarial examples remain in the ℓ_2 ball, *i.e.*, they are in the cap of the ℓ_2 ball. These examples are equally good from the ℓ_2 perspective. This means that even after adversarial training, it is still easy to find good ℓ_2 adversarial examples, making the ℓ_2 robustness of AT-ℓ_∞ almost null.

B.4 Reviewing Defenses Against Multiple Attacks

Table B.3: This table shows a comprehensive list of results consisting of the accuracy of several defense mechanisms against ℓ_2 and ℓ_∞ attacks. This table main objective is to compare the overall performance of 'single' norm defense mechanisms (AT and NI presented in the Section B.2.2) against mixed norms defense mechanisms (MAT & RAT mixed defenses presented in Section B.4).

	Baseline	**AT**		**MAT**		**NI**		**RAT-ℓ_∞**		**RAT-ℓ_2**	
	–	ℓ_∞	ℓ_2	Max	Rand	$\mathcal{N}$	$\mathcal{U}$	$\mathcal{N}$	$\mathcal{U}$	$\mathcal{N}$	$\mathcal{U}$
Natural	0.94	0.85	0.85	0.80	0.80	0.79	0.87	0.74	0.80	0.79	0.87
PGD-ℓ_∞	0.00	0.43	0.37	0.37	0.40	0.23	0.22	0.35	0.40	0.23	0.22
PGD-ℓ_2	0.00	0.37	0.52	0.50	0.55	0.34	0.36	0.43	0.39	0.34	0.37

Adversarial attacks have been an active topic in the machine learning community since their discovery Biggio et al. [2013], Globerson et al. [2006], Szegedy et al. [2014]. Many attacks have been developed. Most of them solve a loss maximization problem with either ℓ_∞ Goodfellow et al. [2015], Kurakin et al. [2016], Madry et al. [2018], ℓ_2 Carlini and Wagner [2017], Kurakin et al. [2016], Madry et al. [2018], ℓ_1 Tramèr and Boneh [2019] or ℓ_0 Papernot et al. [2016] surrogate norms. As we showed, these norms are really different in high dimension. Hence, defending against one norm-based attack is not sufficient to protect against another one. In order to solve this problem, we review several strategies to build defenses against multiple adversarial attacks. These strategies are based on the idea that both types of defense must be used simultaneously in order for the classifier to be protected against multiple attacks. The detailed description of the experimental setting is described in Section B.4.1.

B.4.1 Experimental Setting

To compare the robustness provided by the different defense mechanisms, we use strong adversarial attacks and a conservative setting: the attacker has a total knowledge of the parameters of the model (white-box setting) and we only consider untargeted attacks (a misclassification from one target to any other will be considered as adversarial). To evaluate defenses based on Noise Injection, we use *Expectation Over Transformation* (EOT), the rigorous experimental protocol proposed by Athalye et al. [2018b] and later used by Athalye et al. [2018a], Carlini et al. [2019] to identify flawed defense mechanisms.

To attack the models, we use state-of-the-art algorithms PGD. We run PGD with 20 iterations to generate adversarial examples and with 10 iterations when it is used for adversarial training. The maximum ℓ_∞ bound is fixed to 0.031 and the maximum ℓ_2 bound is fixed to 0.83. As discussed in Section B.2, we chose these values so that the ℓ_∞ and the ℓ_2 balls have similar volumes. Note that 0.83 is slightly above the values typically used in previous publications in the area, meaning the attacks are stronger, and thus more difficult to defend against.

All experiments are conducted on CIFAR-10 with the Wide-Resnet 28-10 architecture. We use the training procedure and the hyper-parameters described in the original paper by Zagoruyko and Komodakis [2016]. Training time varies from 1 day (AT) to 2 days (MAT) on 4 GPUs-V100 servers.

B.4.2 MAT – Mixed Adversarial Training

Earlier results have shown that AT-ℓ_p improves the robustness against corresponding ℓ_p-bounded adversarial examples, and the experiments we present in this section corroborate this observation (See Table B.3, column: AT). Building on this, it is natural to examine the efficiency of *Mixed Adversarial Training* (MAT) against mixed ℓ_∞ and ℓ_2 attacks. MAT is a variation of AT that uses both ℓ_∞-bounded adversarial examples and ℓ_2-bounded adversarial examples as training examples. As discussed in Tramèr and Boneh [2019], there are several possible strategies to mix the adversarial training examples. The first strategy (MAT-Rand) consists in randomly selecting one adversarial example among the two most damaging ℓ_∞ and ℓ_2, and to use it as a training example, as described in Equation (B.16):

MAT-Rand :

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathbb{E}_{p\sim\mathcal{U}(\{2,\infty\})} \max_{\|\tau\|_p \leq \epsilon} \mathcal{L}\left(f_\theta(x+\tau), y\right) \right]. \tag{B.16}$$

An alternative strategy is to systematically train the model with the most damaging adversarial example (ℓ_∞ or ℓ_2). As described in Equation (B.17):

MAT-Max :

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{p\in\{2,\infty\}} \max_{\|\tau\|_p \leq \epsilon} \mathcal{L}\left(f_\theta(x+\tau), y\right) \right]. \tag{B.17}$$

The accuracy of MAT-Rand and MAT-Max are reported in Table B.3 (Column: MAT). As expected, we observe that MAT-Rand and MAT-Max offer better robustness both against PGD-ℓ_2 and PGD-ℓ_∞ adversarial examples than the original AT does. More generally, we can see that AT is a good strategy against loss maximization attacks, and thus it is not surprising that MAT is a good strategy against mixed loss maximization attacks. However efficient in practice, MAT (for the same reasons as AT) lacks theoretical arguments. In order to get the best of both worlds, Salman et al. [2019] proposed to mix adversarial training with randomization.

B.4.3 RAT – Randomized Adversarial Training

We now examine the performance of Randomized Adversarial Training (RAT) first introduced in Salman et al. [2019]. This technique mixes Adversarial Training with Noise Injection. The corresponding loss function is defined as follows:

$$\min_\theta \mathbb{E}(x,y) \sim \mathcal{D} \left[\max_{\|\tau\|_p \leq \epsilon} \mathcal{L}\left(\tilde{f}_\theta(x+\tau), y)\right) \right]. \tag{B.18}$$

where $\tilde{f}_\theta$ is a randomized neural network with noise injection as described in Section B.2.2, and $\|\cdot\|_p$ define which kind of AT is used. For each setting, we consider two noise distributions, Gaussian and Uniform as we did with NI. We also consider two different Adversarial training AT-ℓ_∞ as well as AT-ℓ_2.

The results of RAT are reported in Table B.3 (Columns: RAT-ℓ_∞ and RAT-ℓ_2). We can observe that RAT-ℓ_∞ offers the best extra robustness with both noises, which is consistent with previous experiments, since AT is generally more effective against ℓ_∞ attacks whereas NI is more effective against ℓ_2-attacks. Overall, RAT-ℓ_∞ and a noise from uniform distribution offers the best performances but is still weaker than MAT-Rand. These results are also consistent with the literature, since adversarial training (and its variants) is the best defense against adversarial examples so far.

B.5 Conclusion & Perspective

In this paper, we tackled the problem of protecting neural networks against multiple attacks crafted from different norms. We demonstrated and gave a geometrical interpretation to explain why most defense mechanisms can only protect against one type of attack. Then we reviewed existing strategies that mix defense mechanisms in order to build models that are robust against multiple adversarial attacks. We conduct a rigorous and full comparison of *Randomized Adversarial Training* and *Mixed Adversarial Training* as defenses against multiple attacks.

We could argue that both techniques offer benefits and limitations. We have observed that MAT offers the best empirical robustness against multiples adversarial attacks but this technique is computationally expensive which hinders its use in large-scale applications. Randomized techniques have the important advantage of providing theoretical guarantees of robustness and being computationally cheaper. However, the certificate provided by such defenses is still too small for strong attacks. Furthermore, certain Randomized defenses also suffer from the curse of dimensionality as recently shown by Kumar et al. [2020a].

Although, randomized defenses based on noise injection seem limited in terms of accuracy under attack and scalability, they could be improved either by Learning the best distribution to use or by leveraging different types of randomization such as discrete randomization first proposed in Pinot et al. [2020]. We believe that these certified defenses are the best solution to ensure the robustness of classifiers deployed into real-world applications.

Appendix C

Adversarial Attacks on Linear Contextual Bandits

Contextual bandit algorithms are applied in a wide range of domains, from advertising to recommender systems, from clinical trials to education. In many of these domains, malicious agents may have incentives to force a bandit algorithm into a desired behavior. For instance, an unscrupulous ad publisher may try to increase their own revenue at the expense of the advertisers; a seller may want to increase the exposure of their products, or thwart a competitor's advertising campaign. In this paper, we study several attack scenarios and show that a malicious agent can force a linear contextual bandit algorithm to pull any desired arm $T - o(T)$ times over a horizon of T steps, while applying adversarial modifications to either rewards or contexts with a cumulative cost that only grow logarithmically as $O(\log T)$. We also investigate the case when a malicious agent is interested in affecting the behavior of the bandit algorithm in a single context (e.g., a specific user). We first provide sufficient conditions for the feasibility of the attack and an efficient algorithm to perform an attack. We empirically validate the proposed approaches in synthetic and real-world datasets.

C.1 Introduction

Recommender systems are at the heart of the business model of many industries like e-commerce or video streaming Davidson et al. [2010], Gomez-Uribe and Hunt [2015]. The two most common approaches for this task are based either on matrix factorization Park et al. [2017] or bandit algorithms Li et al. [2010], which both rely on a unaltered feedback loop between the recommender system and the user. In recent years, a fair amount of work has been dedicated to understanding how targeted perturbations in the feedback loop can fool a recommender system into recommending low quality items.

Following the line of research on adversarial attacks in supervised learning Biggio et al. [2012], Goodfellow et al. [2015], Jagielski et al. [2018], Li et al. [2016], Liu et al. [2017], attacks on recommender systems have been focused on filtering-based algorithms Christakopoulou and Banerjee [2019], Mehta and Nejdl [2008] and offline contextual bandits Ma et al. [2018]. The question of adversarial attacks for online bandit algorithms has only been studied quite recently Guan et al. [2020], Immorlica et al. [2018], Jun et al. [2018], Liu and Shroff [2019], and solely in the multi-armed stochastic setting. Although the idea of online adversarial bandit algorithms

is not new (see Exp3 algorithm in Auer et al. [2002]), the focus is different from what we are considering in this article. Indeed, algorithms like Exp3 or Exp4 Lattimore and Szepesvári [2018] are designed to find optimal actions in hindsight in order to adapt to any rewards stream.

The opposition between adversarial and stochastic bandit settings has sparked interests in studying a middle ground. In Bubeck and Slivkins [2012], the learning algorithm has no knowledge of the type of feedback it receives (either stochastic or adversarial). In Gupta et al. [2019], Kapoor et al. [2019], Li et al. [2019b], Lykouris et al. [2018, 2019], the rewards are assumed to be corrupted by adversarial rewards. The authors focus on building algorithms able to find the optimal actions even in the presence of some non-random perturbations. This setting is different from what is studied in this article because those perturbations are bounded and agnostic to arms pulled by the learning algorithm, i.e., the adversary corrupt the rewards before the algorithm chooses an arm.

In the broader Deep Reinforcement Learning (DRL) literature, the focus is placed on modifying the observations of different states to fool a DRL system at inference time Hussenot et al. [2019], Sun et al. [2020] or the rewards Ma et al. [2019].

Contribution. In this work, we first follow the research direction opened by Jun et al. [2018] where the attacker has the objective of fooling a learning algorithm into taking a specific action as much as possible. For example in a news recommendation problem, as described in Li et al. [2010], a bandit algorithm chooses between K articles to recommend to a user, based on some information about them, called context. We assume that an attacker sits between the user and the website, they can choose the reward (i.e., click or not) for the recommended article observed by the recommending algorithm. Their goal is to fool the bandit algorithm into recommending some articles to most users. The contributions of our work can be summarized as follows:

- We extend the work of Jun et al. [2018], Liu and Shroff [2019] to the contextual linear bandit setting showing how to perturb rewards for both stochastic and adversarial algorithms, forcing **any** bandit algorithms to pull a specific set of arms, $o(T)$ times for logarithmic cost for the attacker.
- We analyze, for the first time, the setting in which the attacker can only modify the context x associated with the current user (the reward is not altered). The goal of the attacker is to fool the bandit algorithm into pulling arms of a target set for most users (i.e., contexts) while minimizing the total norm of their attacks. We show that the widely known LinUCB algorithm Abbasi-Yadkori et al. [2011], Lattimore and Szepesvári [2018] is vulnerable to this new type of attack.
- We present a harder setting for the attacker, where the latter can only modify the context associated to a specific user. This situation may occur when a malicious agent has infected some computers with a Remote Access Trojan (RAT). The attacker can then modify the history of navigation of a specific user and, as a consequence, the information seen by the online recommender system.We show how the attacker can attack the two very common bandit algorithms LinUCB and Linear Thompson Sampling (LinTS) Abeille et al. [2017], Agrawal and Goyal [2013] and, in certain cases, force them to pull a set of arms most of the time when a specific context (i.e., user) is presented to the algorithm (i.e., visits a website).

C.2 Preliminaries

We consider the standard contextual linear bandit setting with $K \in \mathbb{N}$ arms. At each time t, the agent observes a context $x_t \in \mathbb{R}^d$, selects an action $a_t \in [\![1, K]\!]$ and observes a reward: $r_{t,a_t} = \langle \theta_{a_t}, x_t \rangle + \eta^t_{a_t}$ where for each arm a, $\theta_a \in \mathbb{R}^d$ is a feature vector and $\eta^t_{a_t}$ is a conditionally independent zero-mean, σ^2-subgaussian noise. The contexts are assumed to be sampled *stochastically* except in App. C.D.

Assumption 2. *There exist $L > 0$ and $\mathcal{D} \subset \mathbb{R}^d$, such that for all t, $x_t \in \mathcal{D}$ and, $\forall x \in \mathcal{D}, \forall a \in [\![1, K]\!]$, $\|x\|_2 \leq L$ and $\langle \theta_a, x \rangle \in (0, 1]$. In addition, we assume that there exists $S > 0$ such that $\|\theta_a\|_2 \leq S$ for all arms a.*

The agent minimizes the cumulative regret after T steps $R_T = \sum_{t=1}^{T} \langle \theta_{a^\star_t}, x_t \rangle - \langle \theta_{a_t}, x_t \rangle$, where $a^\star_t := \text{argmax}_a \langle \theta_a, x_t \rangle$. A bandit learning algorithm $\mathfrak{A}$ is said to be *no-regret* when it satisfies $R_T = o(T)$, i.e., the average expected reward received by $\mathfrak{A}$ converges to the optimal one. Classical bandit algorithms (e.g., LinUCB and LinTS) compute an estimate of the unknown parameters θ_a using past observations. Formally, for each arm $a \in [K]$ we define S^t_a as the set of times up to $t-1$ (included) where the agent played arm a. Then, the estimated parameters are obtained through regularized least-squares regression as $\widehat{\theta}^t_a = (X_{t,a} X^\top_{t,a} + \lambda I)^{-1} X_{t,a} Y_{t,a}$, where $\lambda > 0$, $X_{t,a} = (x_i)_{i \in S^t_a} \in \mathbb{R}^{d \times |S^t_a|}$ and $Y_{t,a} = (r_{i,a_i})_{i \in S^t_a} \in \mathbb{R}^{|S^t_a|}$. Denote by $V_{t,a} = \lambda I + X_{t,a} X^\top_{t,a}$ the design matrix of the regularized least-square problem and by $\|x\|_V = \sqrt{x^\top V x}$ the weighted norm w.r.t. any positive matrix $V \in \mathbb{R}^{d \times d}$. We define the confidence set:

$$\mathcal{C}_{t,a} = \left\{ \theta \in \mathbb{R}^d \,:\, \left\| \theta - \widehat{\theta}_{t,a} \right\|_{V_{t,a}} \leq \beta_{t,a} \right\} \tag{C.1}$$

where $\beta_{t,a} = \sigma \sqrt{d \log\left((1 + L^2(1 + |S^t_a|)/\lambda)/\delta \right)} + S\sqrt{\lambda}$, which guarantees that $\theta_a \in \mathcal{C}_{t,a}$, for all $t > 0$, w.p. $1 - \delta$. This uncertainty is used to balance the exploration-exploitation trade-off either through optimism (e.g., LinUCB) or through randomization (e.g., LinTS).

C.3 Online Adversarial Attacks on Rewards

The ultimate goal of a malicious agent is to force a bandit algorithm to perform a desired behavior. An attacker may simply want to induce the bandit algorithm to perform poorly—ruining the users' experience—or to force the algorithm to suggest a specific arm. The latter case is particularly interesting in advertising where a seller may want to increase the exposure of its product at the expense of the competitors. Note that the users' experience is also compromised by the latter attack since the suggestions they will receive will not be tailored to their needs. Similarly to Jun et al. [2018], Liu and Shroff [2019], we focus on the latter objective, i.e., to fool the bandit algorithm into pulling arms in $A^\dagger$, a set of target arms, for $T - o(T)$ time steps (*independently of the user*).

A way to obtain this behavior is to dynamically modify the reward in order to make the bandit algorithm believe that $a^\dagger$ is optimal, for some $a^\dagger \in A^\dagger$. Clearly, the attacker has to pay a price in order to modify the perceived bandit problem and fool the algorithm. If there is no restriction on when and how the attacker can alter the reward, the attacker can easily fool the algorithm. However, this setting is not interesting since the attacker may pay a cost higher than the loss suffered by the attacked algorithm. An attack strategy is considered successful when the total cost of the attack is sublinear in T.

In this section, we show that under Assumption 2, there exists an attack algorithm that is successful against any bandit algorithm, stochastic or adverserial.

Setting. We assume that the attacker has the same knowledge as the bandit algorithm $\mathfrak{A}$ about the problem (i.e., knows σ and L). The attacker is assumed to be able to observe the context x_t, the arm a_t pulled by $\mathfrak{A}$, and can modify the reward received by $\mathfrak{A}$. When the attacker modifies the reward r_{t,a_t} into $\widetilde{r}_{t,a_t}$ the *instantaneous cost* of the attack is defined as $c_t := \left|r_{t,a_t} - \widetilde{r}_{t,a_t}\right|$. The goal of the attacker is to fool algorithm $\mathfrak{A}$ such that the arms in $A^\dagger$ are pulled $T - o(T)$ times and $\sum_{t=1}^T c_t = o(T)$. We also assume that the action for the arms in the target set is strictly positive for every context $x \in \mathcal{D}$. That is to say that $\Delta := \min_{x\in\mathcal{D}} \left\{ \langle x, \theta_{a^\dagger_\star(x)} \rangle - \max_{a\in A^\dagger, a\neq a^\dagger_\star(x)} \langle x, \theta_a \rangle \right\} > 0$ where $a^\dagger_\star(x) = \arg\max_{a\in A^\dagger} \langle x, \theta_a \rangle$ for every $x \in \mathcal{D}$.

Attack idea. We leverage the idea presented in Liu and Shroff [2019] and Jun et al. [2018] where the attacker lowers the reward of arms $a \notin A^\dagger$ so that algorithm $\mathfrak{A}$ learns that an arm of the target set is optimal for every context. Since $\mathfrak{A}$ is assumed to be no-regret, the attacker only needs to modify the rewards $o(T)$ times to achieve this goal. Lowering the rewards has the effect of shifting the vectors $(\theta_a)_{a\notin A^\dagger}$ to new vectors $(\theta'_a)_{a\notin A^\dagger}$ such that for all arms $a \notin A^\dagger$ and all contexts $x \in \mathcal{D}$, there exists an arm $a^\dagger \in A^\dagger$ such that $\langle \theta'_a, x\rangle \leq \langle \theta_{a^\dagger}, x\rangle$. Since rewards are assumed to be bounded (see Asm. 2), this objective can be achieved by simply forcing the reward of non-target arms $a \notin A^\dagger$ to the minimum value. Contextual ACE (see Fig. C.2) implements a soft version of this idea by leveraging the knowledge of the reward distribution. At each round t, Contextual ACE modifies the reward perceived by $\mathfrak{A}$ as follows:

$$\widetilde{r}^1_{t,a_t} = \eta'_t \mathbb{1}_{\{a_t\notin A^\dagger\}} + r_{t,a_t}\mathbb{1}_{\{a_t\in A^\dagger\}} \tag{C.2}$$

where η'_t is a σ-subgaussian random variable generated by the attacker independently of all other random variables. Contextual ACE transforms the original problem into a *stationary* bandit problem in which there is a targeted arm that is optimal for all contexts and all non targeted arms have expected reward of 0. The following propostion shows that the cumulative cost of the attack is sublinear.

Proposition 17. *For any $\delta \in (0, 1/K]$, when using Contextual ACE algorithm (Fig. C.1) with perturbed rewards $\widetilde{r}^1$, with probability at least $1 - K\delta$, algorithm $\mathfrak{A}$ pulls an arm in $A^\dagger$ for $T - o(T)$ time steps and the total cost of attacks is $o(T)$.*

The proof of this proposition is provided in App. C.A.1. While Prop. 17 holds for any no-regret algorithm $\mathfrak{A}$, we can provide a more precise bound on the total cost by inspecting the algorithm. For example, we can show (see App. C.E), that, with probability at least $1 - K\delta$, the number of times LinUCB Abbasi-Yadkori et al. [2011] pulls arms not in $A^\dagger$ is at most $\sum_{j\notin A^\dagger} N_j(T) \leq \frac{64K\sigma^2\lambda S^2}{\Delta^2}\left(d\log\left(\frac{\lambda + \frac{TL^2}{d}}{\delta^2}\right)\right)^2$. This directly translates into a bound on the total cost.

Comparison with ACE Liu and Shroff [2019]. In the stochastic setting, the ACE algorithm Liu and Shroff [2019] leverages a bound on the expected reward of each arm in order to modify the reward. However, the perturbed reward process seen by algorithm $\mathfrak{A}$ is non-stationary and in general there is no guarantee that an algorithm minimizing the regret in a stationary bandit problem keeps the same performance when the bandit problem is not stationary anymore. Nonetheless, transposing the idea of the ACE algorithm to our setting would give an attack of the following form, where at time t, Alg. $\mathfrak{A}$ pulls arm a_t and receives rewards $\widetilde{r}^2_{t,a_t}$:

$$\widetilde{r}^2_{t,a_t} = (r_{t,a_t} + \max(-1, \min(0, C_{t,a_t})))\mathbb{1}_{\{a_t\notin A^\dagger\}} + r_{t,a_t}\mathbb{1}_{\{a_t\in A^\dagger\}}$$

with $C_{t,a_t} = (1-\gamma)\min_{a^\dagger\in A^\dagger}\min_{\theta\in\mathcal{C}_{t,a^\dagger}} \langle\theta, x_t\rangle - \max_{\theta\in\mathcal{C}_{t,a_t}}\langle\theta, x_t\rangle$. Note that $\mathcal{C}_{t,a}$ is defined as in Eq. C.1 using the *non-perturbed* rewards, i.e., $Y_{t,a} = (r_{i,a_i})_{i\in S^t_a}$.

Bounded Rewards. The bounded reward assumption is necessary in our analysis to prove a formal bound on the total cost of the attacks for *any* no-regret bandit algorithm, otherwise we need more information about the attacked algorithm. In practice, the second attack on the rewards, $\widetilde{r}^2$, can be used in the case of unbounded rewards for any algorithms. The difficulty for unbounded reward is that the attacker has to adapt to the environment reward but in order to do so the reward process observed by the bandit algorithm becomes non-stationary under the attack. Thus, there is no guarantee that an algorithm like LinUCB will pull a target arm as the proof relies on the environment observed by the bandit algorithm being stationary. We observe empirically that the total cost of attack is sublinear when using $\widetilde{r}^2$.

Jun et al. [2018] does not assume that rewards are bounded but focus on attacking algorithms in the stochastic multi-armed setting. That is to say they study attacks only designed for ε-greedy and UCB while we provide an efficient attack for any algorithms in the linear contextual case. We can extend their work, and thus remove the bounded reward assumption, in the linear contextual case by using the following attack, designed only for LinUCB:

$$\widetilde{r}^3_{t,a_t} = \left(r_{t,a_t} + \min_{a^\dagger \in A^\dagger} \min_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle \theta, x_t \rangle - \max_{\theta \in \mathcal{C}_{t,a_t}} \langle \theta, x_t \rangle \right) \mathbb{1}_{\{a_t \notin A^\dagger\}} + r_{t,a_t} \mathbb{1}_{\{a_t \in A^\dagger\}} \tag{C.3}$$

with $C_{t,a}$ defined as in Eq. (C.1). Although, the attack $\widetilde{r}^3$ is not stationary, it is possible to prove that the total cost of attack is $\mathcal{O}(\log(T))$ because we know that the attacked bandit algorithm is LinUCB.

Constrained Attack. When the attacker has a constraint on the instantaneous cost of the attack, using the perturbed reward $\widetilde{r}^1$ may not be possible as the cost of the attack at time t is not decreasing over time. Using the perturbed reward $\widetilde{r}^2$ offers a more flexible type of attack with more control on the instantaneous cost thanks to the parameter γ. But it still suffers from a minimal cost of attack from lowering rewards of arms not in $A^\dagger$.

Defense mechanism. The attack based on reward $\widetilde{r}_1$ is hardly detectable without prior knownledge about the problem. In fact, the reward process associated to $\widetilde{r}_1$ is stationary and compatible with the assumption about the true reward (e.g., subgaussian). While having very low rewards is reasonable in advertising, it can make the attack easily detectable in some other problems. On the other hand, the fact that $\widetilde{r}_2$ is a non-stationary process makes this attack easier to detect. When some data are already available on each arm, the learner can monitor the difference between the average rewards per action computed on new and old data.

C.4 Online Adversarial Attacks on Contexts

In this section, we consider the attacker to be able to alter the context x_t perceived by the algorithm rather than the reward. The attacker is now restricted to change the type of users presented to the learning algorithm $\mathfrak{A}$, hence changing its perception of the environment. We show that under the assumption that the attacker knows a lower-bound to the reward of the target set, it is possible to fool LinUCB.

Setting. As in Sec. C.3, we consider the attacker to have the same knowledge about the problem as $\mathfrak{A}$. The main difference with the previous setting is that the attacker attacks before the algorithm. We adopt a *white-box* Goodfellow et al. [2015] setting attacking LinUCB. The goal of the attacker is unchanged: they aim at forcing the algorithm to pull arms in $A^\dagger$ for $T - o(T)$ time steps while paying a sublinear total cost. We denote by $\widetilde{x}_t$ the context after the attack and by $c_t = \|x_t - \widetilde{x}_t\|_2$ the instantaneous cost.

Difference between attacks on contexts and rewards. Perturbing contexts is fundamentally different from perturbing the rewards. The attacker only modifies the context that is shown to the bandit algorithm. The true context, which is used to compute the reward, remains unchanged. In other words, the attacker cannot modify the reward observed by the bandit algorithm. Instead, the attack algorithm described in this section fools the bandit algorithm by making the rewards appear small relative to the contexts and requires more assumptions on the bandit algorithm than in Sec. C.3.

Attack Idea. The idea of the attack in this setting is similar to the attack of Sec. C.3. The attacker builds a bandit problem where arm an $a^\dagger \in A^\dagger$ is optimal for all contexts by lowering the perceived value of all other arms not in $A^\dagger$. The attacker cannot modify the reward but, thanks to the linear reward assumption, they can scale the contexts to decrease the predicted rewards in the original context.

At time t, the attacker receives the context x_t and computes the attack. Thanks to the white-box setting, it computes the arm a_t that algorithm $\mathfrak{A}$ would pull if presented with context x_t. If $a_t \notin A^\dagger$ then the attacker changes the context to $\widetilde{x}_t = \alpha_{a_t} x_t$ with $\alpha_{a_t} > \max_{x \in \mathcal{D}} \min_{a^\dagger \in A^\dagger} \langle \theta_{a_t}, x\rangle / \langle \theta_{a^\dagger}, x\rangle$.This factor is chosen such that for a ridge regression computed on the dataset $(\alpha x_i, \langle \theta, x_i\rangle)_i$ outputs a parameter close to θ/α therefore the attacker needs to choose α such that for every context $x \in \mathcal{D}$, $\langle x, \theta/\alpha\rangle \leq \max_{a^\dagger \in A^\dagger} \langle x, \theta_{a^\dagger}, x\rangle$. In other words, the attacker performs a dilation of the incoming context every time algorithm $\mathfrak{A}$ does not pull an arm in $A^\dagger$. The fact that the decision rule used by LinUCB is invariant by dilation guarantees that the attacker will not inadvertently lower the perceived rewards for arms in $A^\dagger$. Because the rewards are assumed to be linear, presenting a large context αx and receiving the reward associated with the normal context x will skew the estimated rewards of LinUCB. The attack protocol is summarized in Fig. C.2.

In order to compute the parameter α used in the attack, we make the following assumption concerning the performance of the arms in the target set:

Assumption 3. *For all $x \in \mathcal{D}$, there exists $a^\dagger \in A^\dagger$, such that $0 < \nu \leq \langle x, \theta_{a^\dagger}\rangle$ and ν is known to the attacker.*

Knowing ν. For advertising and recommendation systems, knowing ν is not problematic. Indeed in those cases, the reward is the probability of impression of the ad ($r \in [0, 1]$). The attacker has the freedom to choose one of multiple target arms with strictly positive click probability in every context. This freedom is an important aspect for the attacker since it allows the attacker to cherry pick the target ad(s). In particular, the attacker can estimate ν based on data from previous campaigns (only for the target ad). For instance, a company could have run many ad campaigns for one of their products and try to get the defender's system to advertise it.

An issue is that the norm of the attacked context can be greater that the upper bound L of Assumption 2. To prevent this issue, we choose a context-dependent multiplicative constant $\alpha(x) = \min\{2/\nu, L/\|x\|_2\}$ which amounts to clip the norm of the attacked context to L. In Sec. C.6, we show that this attack is effective for different size of target arms sets. We also show that in the case of contexts such that $\|x\|_2 \leq \nu L/2$ that the cost of attacks is logarithmic in the horizon T.

Proposition 18. *Using the attack described in Fig. C.2 and assuming that $\|x\|_2 \leq \nu L/2$ for all contexts $x \in \mathcal{D}$, for any $\delta \in (0, 1/K]$, with probability at least $1 - K\delta$, the number of times* LinUCB *does not pull an arm in $A^\dagger$ before time T is at most $\sum_{j \notin A^\dagger} N_j(T) \leq 32K^2 \left(\frac{\lambda}{\alpha^2} + \sigma^2 d \log\left(\frac{\lambda d + TL^2\alpha^2}{d\lambda\delta}\right)\right)^3$ with $N_j(T)$ the number of times arm j has been pulled during*

For time $t = 1, 2, ..., T$ **do**

1. Alg. $\mathfrak{A}$ chooses arm a_t based on context x_t
2. Environment generates reward: $r_{t,a_t} = \langle \theta_{a_t}, x_t \rangle + \eta_t$ with $\eta^t_{a_t}$ conditionally σ^2-subgaussian
3. Attacker observes reward r_{t,a_t} and feeds the perturbed reward $\widetilde{r}^1_{t,a_t}$ (or $\widetilde{r}^2_{t,a_t}$) to $\mathfrak{A}$

Figure C.1: Contextual ACE algorithm

Input: attack parameter: α
For time $t = 1, 2, ..., T$ **do**

1. Attacker observes the context x_t, computes potential arm a'_t and sets $\widetilde{x}_t = x_t + (\alpha(x_t) - 1)x_t \mathbb{1}_{\{a'_t \notin A^\dagger\}}$
2. Alg. $\mathfrak{A}$ chooses arm a_t based on context $\widetilde{x}_t$
3. Environment generates reward: $r_{t,a_t} = \langle \theta_{a_t}, x_t \rangle + \eta_t$ with η_t conditionally σ^2-subgaussian
4. Alg. $\mathfrak{A}$ observes reward r_{t,a_t}

Figure C.2: ConicAttack algorithm.

the first T steps, The total cost for the attacker is bounded by: $\sum_{t=1}^{T} c_t \leq \frac{64K^2}{\nu}\left(\frac{\lambda}{\alpha^2} + \sigma^2 d \log\left(\frac{\lambda d + TL^2\alpha^2}{d\lambda\delta}\right)\right)^3$ *with* $\alpha = 2/\nu$.

The proof of Proposition 18 (see App. C.A.2) assumes that the attacker can attack at any time step, and that they can know in advance which arm will be pulled by Alg. $\mathfrak{A}$ in a given context. Thus it is not applicable to random exploration algorithms like LinTS Agrawal and Goyal [2013] and ε-greedy. We also observed empirically that thowe two randomized algorithms are more robust to attacks (see Sec. C.6) than LinUCB.

Norm Clipping. Clipping the norm of the attacked contexts is not beneficial for the attacker. Indeed, this means that an attacked context was violating the assumption (used by the bandit algorithm) that contexts are bounded by L. The attack could then be easily detectable and may succeed only because it is breaking an underlying assumption used by the bandit algorithm. Prop. 18 provides a theoretical grounding for the proposed attack when contexts are bounded by $\nu L/2$ and not only L. Although, we can not prove a bound on the cumulative cost of attacks in general, we show in Sec. C.6 that attacks are still successful for multiple datasets where contexts are not bounded by $\nu L/2$.

C.5 Offline attacks on a Single Context

Previous sections focused on the man-in-the-middle (MITM) attack either on reward or context. The MITM attack allows the attacker to arbitrarily change the information observed by the recommender system at each round. This attack may be hardly feasible in practice, since the exchange channels are generally protected by authentication and cryptographic systems. In this section, we consider the scenario where the attacker has control over a single user u. As an example, consider the case where the device of the user is infected by a malware (e.g., Trojan horse), giving full control of the system to the malicious agent. The attacker can thus modify the context of the specific user (e.g., by altering the cookies) that is perceived by the recommender system. We believe that changes to the context (e.g., cookies) are more subtle and less easily detectable than changes to the reward (e.g., click). Moreover, if the reward is a purchase, it cannot be altered easily by taking control of the user's device. Clearly, the impact of the attacker on the overall performance of the recommender system depends on the frequency of the specific user, that is out of the attacker's control. It may be thus difficult to obtain guarantees on the cumulative regret of algorithm $\mathfrak{A}$. For this reason, we mainly focus on the study of the feasibility of the attack.

The attacker targets a specific user (i.e., the infected user) associated to a context $x^\dagger$. Similarly to Sec. C.4, the objective of the attacker is to find the minimal change to the context presented

to the recommender system $\mathfrak{A}$ such that $\mathfrak{A}$ selects an arm in $A^\dagger$. $\mathfrak{A}$ observes a modified context $\widetilde{x}$ instead of $x^\dagger$. After selecting an arm a_t, $\mathfrak{A}$ observes the true noisy reward $r_{t,a_t} = \langle \theta_{a_t}, x^\dagger \rangle + \eta^t_{a_t}$. We still study a white-box setting: the attacker can access all the parameters of $\mathfrak{A}$.

In this section, we show under which condition it is possible for an attacker to fool both an optimistic and posterior sampling algorithm.

C.5.1 Optimistic Algorithm: LinUCB

We consider the LinUCB algorithm which chooses the arm to pull by maximizing an upper-confidence bound on the expected reward. For each arm a and context x, the UCB value is given by $\max_{\theta \in \mathcal{C}_{t,a}} \langle x, \theta \rangle = \langle x, \hat{\theta}^t_a \rangle + \beta_{t,a} \|x\|_{\widetilde{V}^{-1}_{t,a}}$ (see Sec. **??**). The objective of the attacker is to force LinUCB to pull an arm in $A^\dagger$ once presented with context $x^\dagger$. This means to find a perturbation of context $x^\dagger$ that makes any arm in $A^\dagger$ the most optimistic arm. Clearly, we would like to keep the perturbation as small as possible to reduce the cost for the attacker and the probability of being detected. Formally, the attacker needs to solve the following *non-convex* optimization problem:

$$\min_{y \in \mathbb{R}^d} \quad \|y\|_2 \qquad \text{s.t} \qquad \max_{a \notin A^\dagger} \max_{\theta \in \widetilde{\mathcal{C}}_{t,a}} \langle x^\dagger + y, \theta \rangle + \xi \leq \max_{a^\dagger \in A^\dagger} \max_{\theta \in \widetilde{\mathcal{C}}_{t,a^\dagger}} \langle x^\dagger + y, \theta \rangle \tag{C.4}$$

where $\xi > 0$ is a parameter of the attacker and $\widetilde{\mathcal{C}}_{t,a} := \left\{ \theta \mid \|\theta - \hat{\theta}^t_a\|_{\widetilde{V}_{t,a}} \leq \beta_{t,a} \right\}$ is the confidence set constructed by LinUCB. We use the notation $\widetilde{\mathcal{C}}, \widetilde{V}$ to stress the fact that LinUCB observes only the modified context. In contrast to Sec. C.3 and C.4, the attacker may not be able to force the algorithm to pull any of the target arms in $A^\dagger$. In other words, Problem C.4 may not be feasible. However, we are able to characterize the feasibility of (C.4).

Theorem 16. *Problem* (C.4) *is feasible at time* t iff.

$$\exists \theta \in \cup_{a^\dagger \in A^\dagger} \widetilde{\mathcal{C}}_{t,a^\dagger}, \ \theta \notin \mathit{Conv}\Big(\cup_{a \notin A^\dagger} \widetilde{\mathcal{C}}_{t,a} \Big) \tag{C.5}$$

The condition given by Theorem 16 says that this attack can be done when there exists a vector x for which an arm in $A^\dagger$ is assumed to be optimal according to LinUCB. The condition mainly stems from the fact that optimizing a linear product on a convex compact set will reach its maximum on the edge of this set. In our case this set is the convex hull of the confidence ellipsoids of LinUCB. Although it is possible to use an optimization algorithm for this class of non-convex problems—e.g., DC programming Tuy [1995]—they are still slow compared to convex algorithms. Therefore, we present a simple convex relaxation of the previous problem for a single target arm $a^\dagger \in A^\dagger$ that still enjoys some empirical performance compared to Problem (C.4). The final attack can then be computed as the minimum of the attacks obtained for each $a^\dagger \in A^\dagger$. The relaxed problem is the following for each $a^\dagger \in A^\dagger$:

$$\min_{y \in \mathbb{R}^d} \quad \|y\|_2 \qquad \text{s.t} \qquad \max_{a \neq a^\dagger, a \notin A^\dagger} \max_{\theta \in \mathcal{C}_{t,a}} \langle x^\dagger + y, \theta - \hat{\theta}^t_{a^\dagger} \rangle \leq -\xi \tag{C.6}$$

Since the RHS of the constraint in Problem (C.4) can be written as $\max_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle \theta, x^\dagger + y \rangle$ for any y, the relaxation here consists in using $\langle \theta, x^\dagger + y \rangle$ as a lower-bound to this maximum for any $\theta \in \mathcal{C}_{t,a^\dagger}$.

For the relaxed Problem (C.6), the same type of reasoning as for Problem (C.4) gives that Problem (C.6) is feasible if and only if $\hat{\theta}_{a^\dagger}(t) \notin \text{Conv}\left(\bigcup_{a \neq a^\dagger, a \notin A^\dagger} \mathcal{C}_{t,a} \right)$.

If Condition (C.5) is not met, no arm $a^\dagger \in A^\dagger$ can be pulled by LinUCB. Indeed, the proof of Theorem 16 shows that the upper-confidence of every arm in $A^\dagger$ is always dominated by another arm for any context. In other words, if any arm in $A^\dagger$ is optimal for some contexts then the condition is satisfied a linear number of times for LinUCB (for formal proof of this fact see App. C.A.4).

C.5.2 Random Exploration Algorithm: LinTS

The previous subsection focused on LinUCB, however we can obtain similar guarantees for algorithms with random exploration such as LinTS. In this case, it is not possible to guarantee that a specific arm will be pulled for a given context because of the randomness in the arm selection process. The objective is to guarantee that an arm from $A^\dagger$ is pulled with probability at least $1-\delta$. Similarly to the previous subsection, the problem of the attacker can be written as:

$$\min_{y\in\mathbb{R}^d} \quad \|y\| \qquad \text{s.t} \qquad \mathbb{P}\left(\exists a^\dagger \in A^\dagger,\ \forall a \notin A^\dagger,\ \langle x^\dagger + y, \widetilde{\theta}_a - \widetilde{\theta}_{a^\dagger}\rangle \leq -\xi\right) \geq 1-\delta \tag{C.7}$$

where the $\widetilde{\theta}_a$ for different arms a are independently drawn from a normal distribution with mean $\hat{\theta}_a(t)$ and covariance matrix $\upsilon^2 \bar{V}_a^{-1}(t)$ with $\upsilon = \sigma\sqrt{9d\ln(T/\delta)}$. Solving this problem is not easy and in general not possible, even for a single arm. For a given x and arm a, the random variable $\langle x, \widetilde{\theta}_a\rangle$ is normally distributed with mean $\mu_a(x) := \langle \hat{\theta}_a(t), x\rangle$ and variance $\sigma_a^2(x) := \nu^2 ||x||^2_{\bar{V}_a^{-1}(t)}$. We can then write $\langle x, \widetilde{\theta}_a\rangle = \mu_a(x) + \sigma_a(x) Z_a$ with $(Z_a)_a \sim \mathcal{N}(0, I_K)$. For the sake of clarity, we drop the variable x when writing $\mu_a(x)$ and $\sigma_a(x)$.

Let's imagine (just for this paragraph) that $A^\dagger = \{a^\dagger\}$, then the constraint in Problem (C.7) becomes $\left[1 - \mathbb{E}_{Z_{a^\dagger}}\left(\Pi_{a\notin A^\dagger}\Phi\left(\frac{\sigma_{a^\dagger} Z_{a^\dagger} + \mu_{a^\dagger} - \mu_a}{\sigma_a}\right)\right)\right] \leq \delta$ where Φ is the cumulative distribution function of a normally distributed Gaussian random variable. Unfortunately, computing exactly this expectation is an open problem.

In the more general case where $|A^\dagger| \geq 1$, rewriting the constraints of Problem (C.7) is not possible. Following the idea of Liu and Shroff [2019], for every single target arm $a^\dagger \in A^\dagger$, a possible relaxation of the constraint in Problem (C.7) is, to ensure that there exists an arm $a^\dagger \in A^\dagger$ such that for every arm $a \notin A^\dagger$, $1 - \Phi\left((\mu_{a^\dagger} - \mu_a - \xi)/(\sqrt{\sigma_a^2 + \sigma_{a^\dagger}^2})\right) \leq \frac{\delta}{K - |A^\dagger|}$, where $|A^\dagger|$ is the cardinal of $A^\dagger$. Thus the relaxed version of the attack on LinTS for a single arm $a^\dagger$ is:

$$\min_{y\in\mathbb{R}^d} \|y\| \quad \text{s.t.} \quad \forall a \notin A^\dagger, \langle x^\dagger + y, \hat{\theta}_{a^\dagger} - \hat{\theta}_a\rangle - \xi \geq \nu\Phi^{-1}\left(1 - \tfrac{\delta}{K-|A^\dagger|}\right)\left\|x^\dagger + y\right\|_{\bar{V}_a^{-1} + \bar{V}_{a^\dagger}^{-1}} \tag{C.8}$$

Problem (C.8) is similar to Problem (C.6) as the constraint is also a Second Order Cone Program but with different parameters (see App. C.C). As in section C.5.1, we compute the final attack as the minimum of the attacks computed for each arm in $A^\dagger$.

C.6 Experiments

In this section, we conduct experiments on the attacks on contextual bandit problems with simulated data and two real-word datasets: MovieLens25M Harper and Konstan [2015] and Jester Goldberg et al. [2001]. The synthetic dataset and the data preprocessing step are presented in App. C.B.1.

C.6.1 Attacks on Rewards

We study the impact of the reward attack for 4 contextual algorithms: LinUCB, LinTS, ε-greedy and Exp4. As parameters, we use $L = 1$ for the maximal norm of the contexts, $\delta = 0.01$, $\upsilon = \sigma\sqrt{d\ln(t/\delta))/2}$, $\varepsilon_t = 1/\sqrt{t}$ at each time step t and $\lambda = 0.1$. We choose only a *unique target arm* $a^\dagger$. For Exp4, we use $N = 10$ experts with $N - 2$ experts returning a random arm at each time, one expert choosing arm $a^\dagger$ every time and one expert returning the optimal arm for every context. With this set of experts the regret of bandits with expert advice is the same as in the contextual case. To test the performance of each algorithm, we generate 40 random contextual bandit problems and run each algorithm for $T = 10^6$ steps on each. We report the average cost and regret for each of the 40 problems. Figure C.3 (Top) shows the attacked algorithms using the attacked reward $\widetilde{r}^1$ (reported as "stationary CACE") and the rewards $\widetilde{r}^2$ (reported as CACE).

These experiments show that, even though the reward process is non-stationary, usual stochastic algorithms like LinUCB can still adapt to it and pull the optimal arm for this reward process (which is arm $a^\dagger$). The true regret of the attacked algorithms is linear as $a^\dagger$ is not optimal for all contexts. In the synthetic case, for the algorithms attacked with the rewards $\widetilde{r}^2$, over 1M iterations and $\gamma = 0.22$, the target arm is drawn more than 99.4% of the time on average for every algorithm and more than 97.8% of the time for the stationary attack $\widetilde{r}^1$ (see Table C.2 in App. C.B.2). The dataset-based environments (see Figure C.3 (Left)) exhibit the same behavior: the target arm is pulled more than 94.0% of the time on average for all our attacks on Jester and MovieLens and more than 77.0% of the time in the worst case (for LinTS attacked with the stationary rewards) (see Table C.2).

C.6.2 Attacks on Contexts

We now illustrate the effectiveness of the attack in Alg. C.2. We study the behavior of attacked LinUCB, LinTS, ε-greedy with different size of target arms set ($|A^\dagger|/K \in \{0.3, 0.6, 0.9\}$ with K the total number of arms). We test the performance of LinUCB with the same parameters as in the previous experiments. Yet since the variance is much smaller in this case, we generate a random problem and run 20 simulations for each algorithm. The target arms are chosen randomly and we use the exact lower-bound on the reward of those arms to compute ν.

Table C.1: Percentage of iterations for which the algorithm pulled an arm in the target set $A^\dagger$ (with a target set size of $0.3K$ arms) (**Left**) Online attacks using ContextualConic (CC) algorithm. Percentages are averaged over 20 runs of 1M iterations. (**Right**) Offline attacks with exact (Full) and Relaxed optimization problem. Percentages are averaged over 40 runs of 1M iterations.

	Synthetic	Jester	Movilens
LinUCB	28.91%	26.59%	31.13%
CC LinUCB	98.55%	98.36%	99.61%
ε-greedy	25.7%	25.85%	31.78%
CC ε-greedy	89.71%	99.85%	99.92%
LinTS	27.2%	26.10%	33.24%
CC LinTS	30.93%	97.26%	98.82%

	Synthetic	Jester	MovieLens
LinUCB	0.07%	0.01%	0.39%
LinUCB Relaxed	13.76%	97.81%	4.09%
LinUCB Full	88.30%	99.98%	99.99%
ε-greedy	0.01%	0.00%	0.03%
ε-greedy Full	99.98%	99.95%	99.97%
LinTS	0.02%	0.01%	0.05%
LinTS Relaxed	18.21%	80.48%	5.56%

Table C.1 (Left) shows the percentage of times an arm in $A^\dagger$, for $|A^\dagger| = 0.3K$, has been selected by the attacked algorithm. We see that, as expected, CC LinUCB reaches a ratio of almost 1, meaning the target arms are indeed pulled a linear number of times. A more surprising result (at least not covered by the theory) is that ε-greedy exhibits the same behavior. Similarly to LinTS, ε-greedy exhibits some randomness in the action selection process. It can cause an

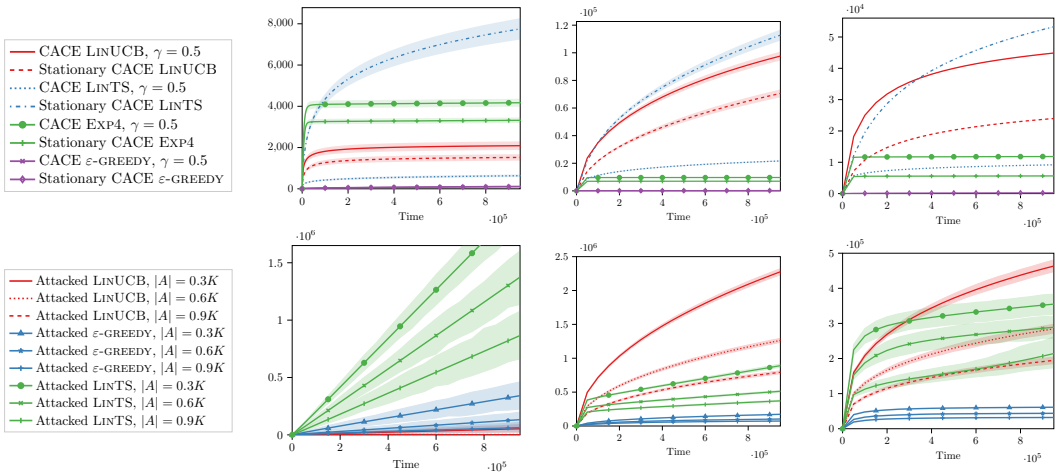


Figure C.3: Total cost of attacks on rewards for the synthetic (Left, $\gamma = 0.22$), Jester (Center, $\gamma = 0.5$) and MovieLens (Right, $\gamma = 0.5$) environments. Bottom, total cost of ContextualConic attacks on the synthetic (Left), Jester (Center) and MovieLens (Right) environments.

arm $a^\dagger \in A^\dagger$ to be chosen when the context is attacked and interfere with the principle of the attack. We suspect that is what happens for LinTS. Fig. C.3 (Bottom) shows the total cost of the attacks for the attacked algorithms . Despite the fact that the estimate of $\theta_{a^\dagger}$ can be polluted by attacked samples, it seems that LinTS can still pick up $a^\dagger$ as being optimal for some dataset like MovieLens and Jester but not on the simulated dataset.

C.6.3 Offline attacks on a Single Context

We now move to the setting described in Sec. C.5 and test the same algorithms as in Sec. C.6.2. We run 40 simulations for each algorithm and each attack type. The target context $x^\dagger$ is chosen randomly and the target arm as the arm minimizing the expected reward for $x^\dagger$. The attacker is only able to modify the incoming context for the target context (which corresponds to the context of one user) and the incoming contexts are sampled uniformly from the set of all possible contexts (of size 100). Table C.1 (Right) shows the percentage of success for each attack. We observe that the non-relaxed attacks on ε-greedy and LinUCB work well across all datasets. However, the relaxed attack for LinUCB and LinTS are not as successful, on the synthetic dataset and MovieLens25M. The Jester dataset seems to be particularly suited to this type of attacks because the true feature vectors are well separated from the convex hull formed by the feature vectors of the other arms: only 5% of Jester's feature vectors are within the convex hull of the others versus 8% for MovieLens and 20% for the synthetic dataset. As expected, the cost of the attacks is linear on all the datasets (see Figure C.6 in App. C.B.4). The cost is also lower for the non-relaxed than for the relaxed version of the attack on LinUCB. Unsurprisingly, the cost of the attacks on LinTS is the highest due to the need to guarantee that $a^\dagger$ will be chosen with high probability (95% in our experiments).

C.7 Conclusion

We presented several settings for online attacks on contextual bandits. We showed that an attacker can force any contextual bandit algorithm to almost always pull an arbitrary target

arm $a^\dagger$ with only sublinear modifications of the rewards. When the attacker can only modify the contexts, we prove that LinUCB can still be attacked and made to almost always pull an arm in $A^\dagger$ by adding sublinear perturbations to the contexts. When the attacker can only attack a single context, we derive a feasibility condition for the attacks and we introduce a method to compute some attacks of small instantaneous cost for LinUCB, ε-greedy and LinTS. To the best of our knowledge, this paper is the first to describe effective attacks on the contexts of contextual bandit algorithms. Our numerical experiments, conducted on both synthetic and real-world data, validate our results and show that the attacks on all contexts are actually effective on several algorithms and with more permissible settings.

C.A Proofs

In this appendix, we present the proofs of different theoretical results presented in the paper.

C.A.1 Proof of Proposition 17

Proposition. *For any $\delta \in (0, 1/K]$, when using Contextual ACE algorithm (Alg. C.1) with perturbed rewards $\tilde{r}^1$, with probability at least $1 - K\delta$, algorithm $\mathfrak{A}$ pulls an arm in $A^\dagger$ for $T - o(T)$ time steps and the total cost of attacks is $o(T)$.*

Proof. Let us consider the contextual bandit problem $\mathcal{A}_1$, with K arms with contexts $x \in \mathcal{D}$ such that every arm in $a^\dagger \in A^\dagger$ has mean reward $\langle \theta_{a^\dagger}, x\rangle$ and all other arms has mean 0. Then the regret of algorithm $\mathfrak{A}$ for this bandit problem is upper-bounded with probability at least $1-\delta$ by a function $f_{\mathfrak{A}}(T)$ such that $f_{\mathfrak{A}}(T) = o(T)$. In addition, the reward process fed to Alg. $\mathfrak{A}$ by the attacker is a stationary reward process with σ^2-subgaussian noise. Therefore, the number of times algorithm $\mathfrak{A}$ pulls an arm not in $A^\dagger$ is upper-bounded by $f_{\mathfrak{A}}(T)/\min_{x\in\mathcal{D}} \Delta(x)$ where for every context $x \in \mathcal{D}$, let $a^\dagger_\star(x) := \arg\max_{a\in A^\dagger} \langle x, \theta_a\rangle$ and $\Delta(x) = \langle x, \theta_{a^\dagger_\star(x)}\rangle - \max_{a\in A^\dagger, a\neq a^\dagger_\star(x)} \langle x, \theta_a\rangle$. In addition, the total cost of the attack is upper-bounded by $\max_{a\in[\![1,K]\!]} \max_{x\in\mathcal{D}} |\langle x, \theta_a\rangle|(T - N_{A^\dagger}(T))$ where $N_{A^\dagger}(T)$ is the number of times an arm in $A^\dagger$ has been pulled up to time T. Thanks to the previous argument, $T - N_{A^\dagger}(T) \leq f_{\mathfrak{A}}(T)/\min_{x\in\mathcal{D}} \Delta(x)$. □

C.A.2 Proof of Proposition 18

Proposition. *Using the attack described in Alg. C.2, for any $\delta \in (0, 1/K]$, with probability at least $1 - K\delta$, the number of times* LinUCB *does not pull an arm in $A^\dagger$ is at most:*

$$\sum_{j\notin A^\dagger} N_j(T) \leq 32K^2 \left(\frac{\lambda}{\alpha^2} + \sigma^2 d \log\left(\frac{\lambda d + TL^2\alpha^2}{d\lambda\delta}\right)\right)^3$$

with $N_j(T)$ the number of times arm j has been pulled after T steps, $||\theta_a|| \leq S$ for all arms a, λ the regularization parameter of LinUCB *and for all $x \in \mathcal{D}$, $||x||_2 \leq L$. The total cost for the attacker is bounded by:*

$$\sum_{t=1}^{T} c_t \leq \frac{64K^2}{\nu} \left(\frac{\lambda}{\alpha^2} + \sigma^2 d \log\left(\frac{\lambda d + TL^2\alpha^2}{d\lambda\delta}\right)\right)^3$$

Proof. Let a_t be the arm pulled by LinUCB at time t. For each arms a, let $\tilde{\theta}_a(t)$ be the result of the linear regression with the attacked context and $\hat{\theta}_a(t, \lambda/\alpha^2)$ the one with the unattacked context and a regularization of $\frac{\lambda}{\alpha^2}$. At any time step t, we can write, for all $a \notin A^\dagger$:

$$\begin{aligned}\tilde{\theta}_a(t) &= \left(\lambda I_d + \sum_{l=0,a_l=a}^{t} \alpha^2 x_l x_l^\intercal\right)^{-1} \sum_{k=0,a_k=a}^{t} r_k \alpha x_k \\ &= \frac{1}{\alpha}\left(\frac{\lambda}{\alpha^2} I_d + \sum_{k=0,a_k=a}^{t} x_k x_k^\intercal\right)^{-1} \sum_{k=0,a_k=a}^{t} r_k x_k \\ &= \frac{\hat{\theta}_a(t, \lambda/\alpha^2)}{\alpha}\end{aligned}$$

We also note that, since the contexts are not modified for arms in $a^\dagger \in A^\dagger$: $\tilde{\theta}_{a^\dagger}(t) = \hat{\theta}_{a^\dagger}(t, \lambda)$. In addition, for any context x and arm $a \notin A^\dagger$, the exploration term used by LinUCB becomes:

$$||x||_{\tilde{V}_{a,t}^{-1}} = \frac{1}{\alpha}||x||_{\hat{V}_{a,t}^{-1}} \tag{C.9}$$

where $\tilde{V}_{a,t} = \lambda I_d + \sum_{l=0,a_l=a}^{t} \alpha^2 x_l x_l^\intercal$ and $\hat{V}_{a,t}^{-1} = \lambda/\alpha^2 I_d + \sum_{k=0,a_k=a}^{t} x_k x_k^\intercal$. For a time t, if presented with context x_t LinUCB pulls arm $a_t \notin A^\dagger$, we have:

$$\alpha\left(\left\langle \hat{\theta}_{a^\dagger}(t), x_t \right\rangle + \beta_{a^\dagger}(t)||x_t||_{V_{a^\dagger,t}^{-1}}\right) \leq \left\langle \hat{\theta}_{a_t}(t, \lambda/\alpha^2), x_t \right\rangle + \beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}}$$

As $\alpha = \frac{2}{\nu} \geq \min_{a^\dagger \in A^\dagger} \frac{2}{\langle \theta_{a^\dagger}, x_t \rangle}$, we deduce that on the event that the confidence sets (Theorem 2 in Abbasi-Yadkori et al. [2011]) hold for arm $a^\star$:

$$2 \leq \left\langle \hat{\theta}_{a_t}(t, \lambda/\alpha^2), x_t \right\rangle + \beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}} \leq \langle \theta_{a_t}, x_t \rangle + 2\beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}}$$

Thus, $1 \leq 2 - \langle \theta_{a_t}, x_t \rangle \leq 2\beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}}$. Therefore,

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{1}_{\{a_t \notin A^\dagger\}} &\leq \sum_{t=1}^{T} \min(2\beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}}, 1)\mathbb{1}_{\{a_t \notin A^\dagger\}} \\ &\leq \sum_{j \notin A^\dagger} 2\beta_j(T)\sqrt{\sum_{t=1}^{T} \mathbb{1}_{\{a_t=j\}} \sum_{t=1,a_t=j}^{T} \min(1, ||x_t||^2_{\hat{V}_{j,t}^{-1}})}\end{aligned}$$

But using Lemma 11 from Abbasi-Yadkori et al. [2011] and the bound on the $\beta_j(T)$ for all arms j, we have with Jensen inequality:

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{1}_{\{a_t \notin A^\dagger\}} \leq & 4\sqrt{K \sum_{t=1}^{T} \mathbb{1}_{\{a_t \notin A^\dagger\}} d \log\left(1 + \frac{\alpha^2 T L^2}{\lambda d}\right)} \\ & \times \left(\sqrt{\lambda/\alpha^2} S + \sigma\sqrt{2\log(1/\delta) + d\log(1 + \frac{\alpha^2 T L^2}{\lambda d})}\right)\end{aligned}$$

□

C.A.3 Proof of Theorem 16

Theorem. *For any $\xi > 0$, Problem* (C.4) *is feasible if and only if:*

$$\exists \theta \in \bigcup_{a^\dagger \in A^\dagger} \mathcal{C}_{t,a^\dagger}, \qquad \theta \notin Conv\left(\bigcup_{a \notin A^\dagger} \mathcal{C}_{t,a}\right) \tag{C.10}$$

where for every arm a, $\mathcal{C}_{t,a} := \left\{\theta \mid ||\theta - \hat{\theta}_a(t)||_{\tilde{V}_{a,t}} \leq \beta_a(t)\right\}$ with $\hat{\theta}_a(t)$ the least squares estimate for arm a built by LinUCB *and*

$$\tilde{V}_{a,t} = \lambda I_d + \sum_{l=1, x_l \neq x^\dagger}^{t} \mathbb{1}_{\{a_l = a\}} x_l x_l^\intercal + \sum_{l=1, x_l = x^\dagger}^{t} \mathbb{1}_{\{a_l = a\}} \tilde{x}_l \tilde{x}_l^\intercal$$

the design matrix of LinUCB *at time t for all arms a (where $\tilde{x}_l$ is the modified context)*

Proof. The proof of Theorem 16 is decomposed in two parts.

First, let us assume that Equation (C.10) is satisfied. Then, let us define $a^\dagger \in A^\dagger$ such that $\theta \in \mathcal{C}_{t,a^\dagger} \setminus \text{Conv}\left(\bigcup_{a \notin A^\dagger} \mathcal{C}_{t,a}\right)$, then by the theorem of separation of convex sets applied to $\mathcal{C}_{t,a^\dagger}$ and $\{\theta\}$. There exists a vector v and $c_1 < c_2$ such that for all $y \in \text{Conv}\left(\bigcup_{a \neq a^\dagger} \mathcal{C}_{t,a}\right)$:

$$\langle y, v \rangle \leq c_1 < c_2 \leq \langle \theta, v \rangle .$$

Hence, for $\xi > 0$ we have that for $\tilde{v} = \frac{\xi}{c_2 - c_1} v$ that:

$$\langle y, \tilde{v} \rangle + \xi \leq \langle \theta, \tilde{v} \rangle$$

So the problem is feasible.

Secondly, let us assume that an attack is feasible. Then there exists a vector y such that:

$$\max_{a^\dagger \in A^\dagger} \max_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle y, \theta \rangle > c_1 := \max_{a \notin A^\dagger} \max_{\theta \in \mathcal{C}_{t,a}} \langle y, \theta \rangle \tag{C.11}$$

Let us reason by contradiction. We assume that $\bigcup_{a \in A^\dagger} \mathcal{C}_{t,a^\dagger} \subset \text{Conv}\left(\bigcup_{a \notin A^\dagger} \mathcal{C}_{t,a}\right)$ and consider

$$\theta^* \in \bigcup_{a \in A^\dagger} \mathcal{C}_{t,a^\dagger} \text{ such that } \langle y, \theta^* \rangle = \max_{a^\dagger \in A^\dagger} \max_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle y, \theta \rangle$$

As we assumed $\bigcup_{a \in A^\dagger} \mathcal{C}_{t,a^\dagger} \subset \text{Conv}\left(\bigcup_{a \notin A^\dagger} \mathcal{C}_{t,a}\right)$, there exists $n \in \mathbb{N}^\star$, $\lambda_1, \cdots, \lambda_n \geq 0$ and $\theta_1, \cdots, \theta_n \in \bigcup_{a \notin A^\dagger} \mathcal{C}_{t,a}$ such that

$$\theta^* = \sum_{i=1}^{n} \lambda_i \theta_i \text{ and } \sum_{i=1}^{n} \lambda_i = 1$$

Thus

$$\langle y, \theta^* \rangle = \sum_i \lambda_i \langle y, \theta_i \rangle \leq c_1 \sum_{i=1}^{n} \lambda_i = c_1 \tag{C.12}$$

We assumed that the problem is feasible, so $c_1 < \langle y, \theta^* \rangle$ according to Eq. C.11. It contradicts Eq. C.12. □

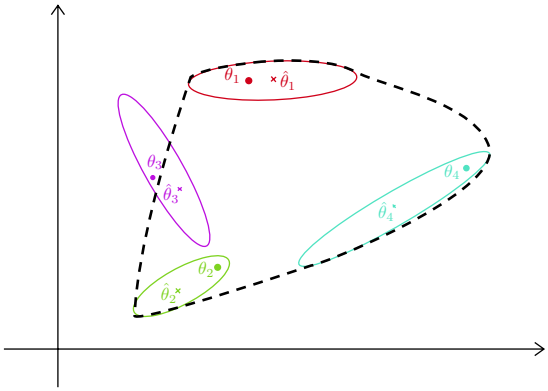


Figure C.4: Illustrative example of condition (C.5). The target arm is arm 3 or 5 and the dashed black line is the convex hull of the other confidence sets. The ellipsoids are the confidence sets $\mathcal{C}_{t,a}$ for each arm a. If we consider only arms $\{1, 2, 4, 5\}$, and we use 5 as the target arm, the condition (C.5) is satisfied as there is a θ outside the convex hull of the other confidence sets. On the other hand, if we consider arms $\{1, 2, 3, 4\}$ and we use 3 as the target arm, the condition is not satisfied anymore.

C.A.4 Condition of Sec. C.5

Let us assume that there is an arm in $a^\dagger \in A^\dagger$ which is optimal for some contexts. More formally, there exists a subspace $V \subset \mathcal{D}$ such that:

$$\forall x \in V, \exists a^\dagger_\star(x) \in A^\dagger, \forall a \in [\![1, K]\!] \setminus \{a^\dagger_\star(x)\} \qquad \langle x, \theta_{a^\dagger_\star(x)} \rangle > \langle x, \theta_a \rangle \,.$$

We also assume that the distribution of the contexts is such that, for all t, $\mu := \mathbb{P}(x_t \in V) > 0$. Then, the regret is lower-bounded in expectation by:

$$\mathbb{E}(R_T) = \mathbb{E}\left(\sum_{t=1}^{T} \mathbb{1}_{\{x_t \in V\}}\big(\left\langle x_t, \theta_{a^\dagger_\star(x_t)} - \theta_{a_t} \right\rangle\big)\right) \geq \mu m(T) \min_{x \in V} \max_{a \neq a^\dagger_\star(x)} \langle \theta_{a^\dagger_\star(x)} - \theta_a, x \rangle$$

where $m(T)$ is the expected number of times $t \leq T$ such that condition (C.5) is not met. LinUCB guarantees that $\mathbb{E}(R_T) \leq \mathcal{O}(\sqrt{T})$ for every T. Hence, $m(T) \leq \mathcal{O}\left(\frac{\sqrt{T}}{\mu \min_{x \in V} \max_{a \neq a^\dagger_\star(x)} \langle \theta_{a^\dagger_\star(x)} - \theta_a, x \rangle}\right)$. This means that, in an unattacked problem, condition (C.5) is met $T - \mathcal{O}(\sqrt{T})$ times. On the other hand, when the algorithm is attacked the regret of LinUCB is not sub-linear as the confidence bound for the target arm is not valid anymore. Hence we cannot provide the same type of guarantees for the attacked problem.

C.B Experiments

C.B.1 Datasets and preprocessing

We present here the datasets used in the article and how we preprocess them for numerical experiments conducted in Section C.6.

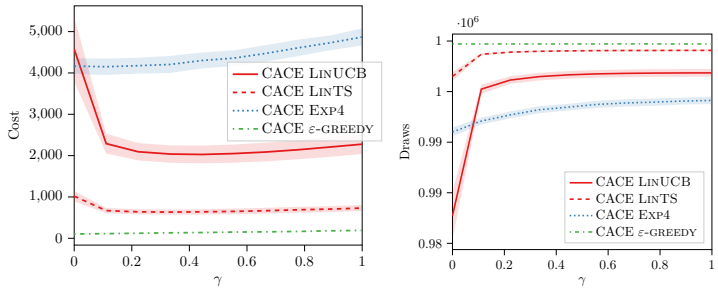
We consider two types of experiments, one on synthetic data with a contextual MAB problems with $K = 10$ arms such that for every arm a, θ_a is drawn from a folded normal distribution in

dimension $d = 30$. We also use a finite number of contexts (10), each of them is drawn from a folded normal distribution projected on the unit circle multiplied by a uniform radius variable (i.i.d. across all contexts). Finally, we scale the expected rewards in $(0, 1]$ and the noise is drawn from a centered Gaussian distribution $\mathcal{N}(0, 0.01)$.

The second type of experiments is conducted in the real-world datasets Jester Goldberg et al. [2001] and MovieLens25M Harper and Konstan [2015]. Jester consists of joke ratings on a continuous scale from -10 to 10 for 100 jokes from a total of 73421 users. We use the features extracted via a low-rank matrix factorization ($d = 35$) to represent the actions (i.e., the jokes). We consider a complete subset of 40 jokes and 19181 users . Each user rates all the 40 jokes. At each time, a user is randomly selected from the 19181 users and mean rewards are normalized in $[0, 1]$. The reward noise is $\mathcal{N}(0, 0.01)$. The second dataset we use is MovieLens25M. It contains 25000095 ratings created by 162541 users on 62423 movies. We perform a low-rank matrix factorization to compute users features and movies features. We keep only movies with at least 1000 ratings, which leave us with 162539 users and 3794 movies. At each time step, we present a random user, and the reward is the scalar product between the user feature and the recommend movie feature. All rewards are scaled to lie in $[0, 1]$ and a Gaussian noise $\mathcal{N}(0, 0.01)$ is added to the rewards.

C.B.2 Attacks on Rewards

In this appendix, we present empirical evolution of the total cost and the number of draws for a unique target arm as a function of the attack parameter γ for the Contextual ACE attack with perturbed rewards $\tilde{r}^2$ on generated data.



(a) Total cost (b) Number of draws

Figure C.5: Total cost of attacks and number of draws of the target arm at $T = 10^6$ as a function of γ on synthetic data

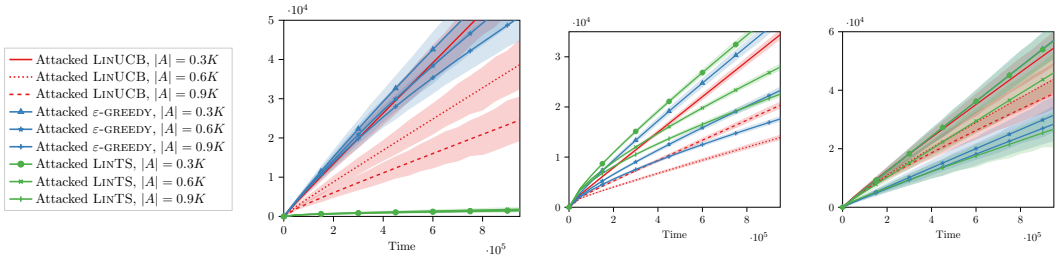
Fig. C.5 (left) shows that the total cost of attacks seems to be quite invariant w.r.t. γ except when $\gamma \to 0$ because the difference between the target arm and the other becomes negligible. This is also depicted by the total number of draws (Fig. C.5, Right) as the number of draws plummets when $\gamma \to 0$.

C.B.3 Attacks on all Contexts

Fig. **??** shows the regret for all the attacks. This figure shows that even though the total cost of attacks is linear for algorithms like LinTS in the synthetic dataset, the regret is linear. More generally, we observe that the regret is linear for all attacked algorithms on all datasets.

Table C.2: Number of draws of the target arm $a^\dagger$ at $T = 10^6$, for the synthetic data, $\gamma = 0.22$ for the Contextual ACE algorithm and for the Jester and MovieLens datasets $\gamma = 0.5$.

	Synthetic	Jester	Movilens
LinUCB	86,731.6	23,548.16	25,017.31
CACE LinUCB	996,238.6	921,083.69	944,721.28
Stationary CACE LinUCB	995,578.88	862,095.67	931,531.6
ε-greedy	111,380.44	21,911.54	3,165.81
CACE ε-greedy	999,812.92	999,755.72	999,776.82
Stationary CACE ε-greedy	999,806.32	999,615.98	999,316.76
LinTS	91,664.8	23,398.3	30,189.84
CACE LinTS	998,997.04	976,708.9	990,250.67
Stationary CACE LinTS	977,850.96	784,715.62	845,512.98
Exp4	93,860.4	29,147.01	17,985.78
CACE Exp4	992,793.36	989,214.36	936,230.4
Stationary CACE Exp4	993,673.24	988,463.56	934,304.23



C.B.4 Attack on a single context

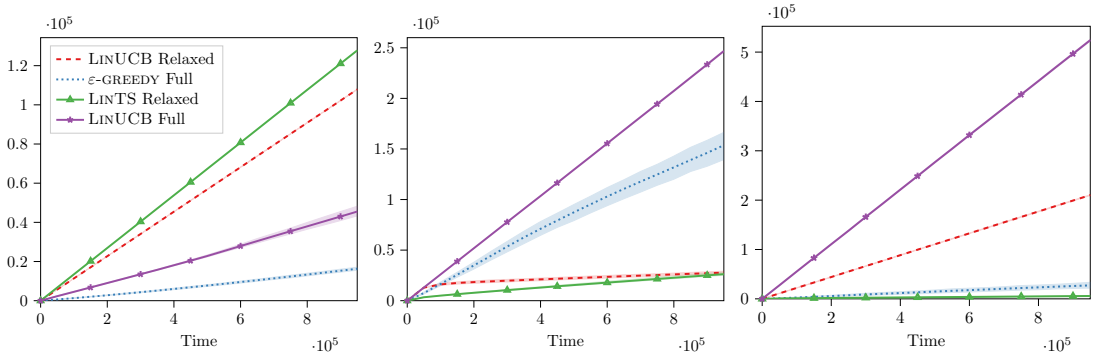
The attacks are computed by solving the optimization problems C.4 and C.6 (Sec. C.5). We choose the libraries according to their efficiency for each problem we need to solve. For Problem (C.6) and Problem (C.8) we use cvxpy Agrawal et al. [2018] and the ECOS solver. For Problem (C.4) we use the SLSQP method from the Scipy optimize library Virtanen et al. [2019] to solve the full LinUCB problem (Equation C.4) and quadprog to solve the quadratic problem to attack ε-greedy.

C.C Problem (C.8) as a Second Order Cone (SOC) Program

Problem (C.6) and Problem (C.8) are both SOC programs. We can see the similarities between both problems as follows. Let us define for every arm $a \notin A^\dagger$, the ellipsoid:

$$\mathcal{C}'_{t,a} := \left\{ y \in \mathbb{R}^d \mid ||y - \hat{\theta}_a(t)||_{A_a^{-1}(t)} \leq \upsilon \Phi^{-1} \left(1 - \frac{\delta}{K - |A^\dagger|}\right) \right\}$$

with $A_a(t) = \tilde{V}_a^{-1}(t) + \tilde{V}_{a^\dagger}^{-1}(t)$ with $\tilde{V}_a(t)$ and $\tilde{V}_{a^\dagger}(t)$ the design matrix built by LinTS and $\hat{\theta}_a(t)$ the least squares estimate of θ_a at time t. Therefore for an arm a, the constraint in Problem



(a) Synthetic data (b) Jester Dataset (c) MovieLens Dataset

Figure C.6: Total cost of the attacks for the attacks one one context on our synthetic dataset, Jester and MovieLens. As expected, the total cost is linear.

(C.8) can be written for any $y \in \mathbb{R}^d$ and some arm $a^\dagger \in A^\dagger$ as:

$$\left\langle x^\star + y, \hat{\theta}_{a^\dagger}(t) \right\rangle - \xi \geq \max_{z \in \mathcal{C}'_{t,a}} \langle z, x^\star + y \rangle$$

Indeed for any $x \in \mathbb{R}^d$,

$$\begin{aligned}
\max_{y \in \mathcal{C}'_{t,a}} \langle y, x \rangle &= \left\langle x, \hat{\theta}_a(t) \right\rangle + \upsilon \Phi^{-1}\left(1 - \frac{\delta}{K - |A^\dagger|}\right) \times \max_{||A_a^{-1/2}(t)u||_2 \leq 1} \langle u, x \rangle \\
&= \left\langle x, \hat{\theta}_a(t) \right\rangle + \upsilon \Phi^{-1}\left(1 - \frac{\delta}{K - |A^\dagger|}\right) \max_{||z||_2 \leq 1} \left\langle z, A_a^{1/2}(t)x \right\rangle \\
&= \left\langle x, \hat{\theta}_a(t) \right\rangle + \upsilon \Phi^{-1}\left(1 - \frac{\delta}{K - |A^\dagger|}\right) \|A_a^{1/2}(t)x\|_2
\end{aligned}$$

Thus, the constraint is feasible if and only if:

$$\hat{\theta}_{a^\dagger}(t) \notin \text{Conv}\left(\bigcup_{a \notin A^\dagger} \mathcal{C}'_{t,a}\right)$$

C.D Attacks on Adversarial Bandits

In the previous sections, we studied algorithms with sublinear regret R_T, i.e., mainly bandit algorithms designed for stochastic stationary environments. Adversarial algorithms like Exp4 do not provably enjoy a sublinear **stochastic** regret R_T (as defined in the introduction) [1]. In addition, because this type of algorithms are, by design, robust to non-stationary environments, one could expect them to induce a linear cost on the attacker. In this section, we show that this is not the case for most contextual adversarial algorithms. Contextual

[1] Exp4 enjoys a sublinear hindsight regret though. Showing a sublinear upper-bound for the stochastic regret of Exp4 is still an open problem (see Section 29.1 in Lattimore and Szepesvári [2018])

adversarial algorithms are studied through the reduction to the bandit with expert advice problem. This is a bandit problem with K arms where at every step, N experts suggest a probability distribution over the arms. The goal of the algorithm is to learn which expert gets the best expected reward in hindsight after T steps. The regret in this type of problem is defined as $R_T^{\exp} = \mathbb{E}\left(\max_{m\in[\![1,N]\!]}\sum_{t=1}^T\sum_{j=1}^K E_{m,j}^{(t)} r_{t,j} - r_{t,a_t}\right)$ where $E_{m,j}^{(t)}$ is the probability of selecting arm j for expert m. In the case of contextual adversarial bandits, the experts first observe the context x_t before recommending an expert m. Assuming the current setting with linear rewards, we can show that if an algorithm $\mathfrak{A}$, like Exp4, enjoys a sublinear regret $R_T^{\exp}$, then, using the Contextual ACE attack with either $\tilde{r}^1$ or $\tilde{r}^2$, the attacker can fool the algorithm into pulling arm $a^\dagger$ a linear number of times under some mild assumptions. However, attacking contexts for this type of algorithm is difficult because, even though the rewards are linear, the experts are not assumed to use a specific model for selecting an action.

Proposition 19. *Suppose an adversarial algorithm $\mathfrak{A}$ satisfies a regret $R_T^{\exp}$ of order $o(T)$ for any bandit problem and that there exists an expert $m^\star$ such that $T - \sum_{t=1}^T \mathbb{E}\left(E_{m^\star,a_{t,\star}^\dagger}^{(t)}\right) = o(T)$ with $a_{t,\star}^\dagger$ the optimal arim in $A^\dagger$ at time t. Then attacking alg. $\mathfrak{A}$ with Contextual ACE leads to pulling arm $a^\dagger$, $T - o(T)$ of times in expectation with a total cost of $o(T)$ for the attacker.*

Proof. Similarly to the proof of Proposition 17, let's define the bandit with expert advice problem, $\mathcal{A}_i$, such that at each time t the reward vector is $(\tilde{r}_{t,a}^i)_a$ (with $i \in \{1,2\}$). The regret of this algorithm is: $\tilde{R}_T^{i,\exp} = \mathbb{E}\left(\max_{m\in[\![1,N]\!]}\sum_{t=1}^T E_m^{(t)}\tilde{r}_t^i - \tilde{r}_{t,a_t}^i\right) \in o(T)$. The regret of the learner is: $\mathbb{E}\left(\max_{m\in[\![1,N]\!]}\sum_{t=1}^T E_m^{(t)} r_t - r_{t,a_t}\right)$ where a_t are the actions taken by algorithm $\mathcal{A}_i$ to minimize $\tilde{R}_T^{i,\exp}$. Then we have:

$$\tilde{R}_T^{i,\exp} \geq \mathbb{E}\left(\sum_{t=1}^T\sum_{j=1}^K (E_{m^\star,j}^{(t)} - \mathbb{1}_{\{j=a_{t,\star}^\dagger\}})\tilde{r}_{t,j}^i + \sum_{t=1}^T \tilde{r}_{t,a_{t,\star}^\dagger}^i - \tilde{r}_{t,a_t}^i\right)$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left(\sum_{t=1}^T \tilde{r}_{t,a_{t,\star}^\dagger}^i - \tilde{r}_{t,a_t}^i\right) &\leq \tilde{R}_T^{i,\exp} + \mathbb{E}\left(\sum_{t=1}^T\sum_{j=1}^K (\mathbb{1}_{\{j=a_{t,\star}^\dagger\}} - E_{m^\star,j}^{(t)})\tilde{r}_{t,j}^i\right) \\
&\leq \tilde{R}_T^{i,\exp} + \mathbb{E}\left(\sum_{t=1}^T (1 - E_{m^\star,a_{t,\star}^\dagger}^{(t)})\tilde{r}_{t,j}^i\right) \\
&\leq \tilde{R}_T^{i,\exp} + \mathbb{E}\left(\sum_{t=1}^T (1 - E_{m^\star,a_{t,\star}^\dagger}^{(t)})\right)
\end{aligned}$$

For strategy $i = 1$, we have:

$$\mathbb{E}\left(\sum_{t=1}^T \tilde{r}_{t,a_{t,\star}^\dagger}^1 - \tilde{r}_{t,a_t}^1\right) = \sum_{t=1}^T \mathbb{E}\left(r_{t,a_{t,\star}^\dagger} - \mathbb{1}_{\{a_t\in A^\dagger\}}\right) \geq \left(T - \mathbb{E}\left(\sum_{t=1}^T \mathbb{1}_{\{a_t = a_{t,\star}^\dagger\}}\right)\right)\Delta$$

where $\Delta := \min_{x\in\mathcal{D}}\left\{\langle\theta_{a^\dagger(x)}, x\rangle - \max_{a\in A^\dagger, a\neq a^\dagger(x)}\langle\theta_{a'}, x\rangle\right\}$ with $a^\dagger(x) := \arg\max_{a\in A^\dagger}\langle\theta_a, x\rangle$. Then, as $\tilde{R}_T^{1,\exp} \in o(T)$ and $\mathbb{E}\left(\sum_{t=1}^T (1 - E_{m^\star,a_{t,\star}^\dagger}^{(t)})\right) \in o(T)$, we deduce that $\mathbb{E}(\sum_t \mathbb{1}_{\{a_t = a_{t,\star}^\dagger\}}) = T - o(T)$.

For strategy $i = 2$, and $\delta > 0$, let us denote by E_δ the event that all confidence intervals hold with probability $1-\delta$. But on the event E_δ, for a time t where $a_t \neq a^\dagger_{t,\star}$ and such that $-1 \leq C_{t,a_t} \leq 0$:

$$\begin{aligned}\tilde{r}^2_{t,a_t} = r_{t,a_t} + C_{t,a_t} &= (1-\gamma) \min_{a^\dagger \in A^\dagger} \min_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle \theta, x_t \rangle + \eta_{a_t,t} + \langle \theta_a, x_t \rangle - \max_{\theta \in \mathcal{C}_{t,a_t}} \langle \theta, x_t \rangle \\ &\leq (1-\gamma)\langle \theta_{a^\dagger_{t,\star}}, x_t \rangle + \eta_{a_t,t}\end{aligned}$$

when $C_{t,a_t} > 0$ (still on the event E_δ):

$$\tilde{r}^2_{t,a_t} = r_{t,a_t} \leq (1-\gamma)\langle \theta_{a^\dagger_{t,\star}}, x_t \rangle + \eta_{a_t,t}$$

because $C_{t,a_t} > 0$ means that $(1-\gamma)\langle \theta_{a^\dagger_{t,\star}}, x_t \rangle \geq (1-\gamma) \min_{a^\dagger \in A^\dagger} \min_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle \theta, x_t \rangle \geq \max_{\theta \in \mathcal{C}_{t,a_t}} \langle \theta, x_t \rangle \geq \langle \theta_a, x_t \rangle$. But finally, when $C_{t,a_t} \leq -1$, $\tilde{r}^2_{t,a_t} = r_{t,a_t} - 1 \leq \eta_{a_t,t} \leq (1-\gamma)\langle \theta_{a^\dagger_{t,\star}}, x_t \rangle + \eta_{a_t,t}$. But on the complementary event E^c_δ, $\tilde{r}^2_{t,a_t} \leq r_{t,a_t}$. Thus, given that the expected reward is assumed to be bounded in $(0,1]$ (Assumption 2):

$$\begin{aligned}\mathbb{E}\left(\sum_{t=1}^T \tilde{r}^2_{t,a^\dagger_{t,\star}} - \tilde{r}^2_{t,a_t}\right) &= \mathbb{E}\left(\sum_{t=1}^T (r_{t,a^\dagger} - \tilde{r}^2_{t,a_t}) \mathbb{1}_{\{a_t \neq a^\dagger_{t,\star}\}}\right) \\ &\geq \mathbb{E}\left(\sum_{t=1}^T \min\{\gamma \min_{x \in \mathcal{D}} \langle x, \theta_{a^\dagger_{t,\star}} \rangle, \Delta\} \mathbb{1}_{\{a_t \neq a^\dagger_{t,\star}\}} \mathbb{1}_{\{E_\delta\}}\right) - T\delta\end{aligned}$$

Finally, putting everything together we have:

$$\mathbb{E}\left(\sum_{t=1}^T \gamma \min_{x \in \mathcal{D}} \langle x, \theta_{a^\dagger_{t,\star}} \rangle \mathbb{1}_{\{a_t \neq a^\dagger_{t,\star}\}}\right) \leq \tilde{R}_T^{2,\exp} + \mathbb{E}\left(\sum_{t=1}^T (1 - E^{(t)}_{m^\star, a^\dagger_{t,\star}})\right) + \delta T \left(\min\{\gamma \min_{a^\dagger \in A^\dagger} \min_{x \in \mathcal{D}} \langle x, \theta_{a^\dagger} \rangle, \Delta\} + 1\right)$$

Hence, because $\tilde{R}_T^{1,\exp} = o(T)$ and $\mathbb{E}\left(\sum_{t=1}^T (1 - E^{(t)}_{m^\star, a^\dagger})\right) = o(T)$ we have that for $\delta \leq 1/T$, the expected number of pulls of the optimal arm in $A^\dagger$ is of order $o(T)$. In addition, the cost for the attacker is bounded by:

$$\mathbb{E}\left(\sum_{t=1}^T c_t\right) = \mathbb{E}\left(\sum_{t=1}^T \mathbb{1}_{\{a_t \neq a^\dagger_{t,\star}\}} \left| \max(-1, \min(C_{t,a_t}, 0)) \right|\right) \leq \mathbb{E}\left(\sum_{t=1}^T \mathbb{1}_{\{a_t \neq a^\dagger_{t,\star}\}}\right)$$

□

The proof is similar to the one of Prop. 17. The condition on the expert in Prop. 19 means that there exists an expert which believes an arm $a^\dagger \in A^\dagger$ is optimal most of the time. The adversarial algorithm will then learn that this expert is optimal. Algorithm Exp4 has a regret $R_T^{\exp}$ bounded by $\sqrt{2TK\log(N)}$, thus the total number of pulls of arms not in $A^\dagger$ is bounded by $\sqrt{2TK\log(M)}/\gamma$. This result also implies that for adversarial algorithms like Exp3 Auer et al. [2002], the same type of attacks could be used to fool $\mathfrak{A}$ into pulling arms in $A^\dagger$ because the MAB problem can be seen as a reduction of the contextual bandit problem with a unique context and one expert for each arm.

C.E Contextual Bandit Algorithms

In this appendix, we present the different bandit algorithms studied in this paper. All algorithms we consider except Exp4 uses disjoint models for building estimate of the arm feature vectors $(\theta_a)_{a \in [\![1,K]\!]}$. Each algorithm (except Exp4) builds least squares estimates of the arm features.

Algorithm 8: Contextual LinUCB

Input: regularization λ, number of arms K, number of rounds T, bound on context norms: L, bound on norms θ_a: D
Initialize for every arm a, $\bar{V}_a^{-1}(t) = \frac{1}{\lambda} I_d$, $\hat{\theta}_a(t) = 0$ and $b_a(t) = 0$
for $t = 1, ..., T$ **do**
  Observe context x_t
  Compute $\beta_a(t) = \sigma \sqrt{d \log\left(\frac{1+N_a(t)L^2/\lambda}{\delta}\right)}$ with $N_a(t)$ the number of pulls of arm a
  Pull arm $a_t = \text{argmax}_a \langle \hat{\theta}_a(t), x_t \rangle + \beta_a(t) ||x_t||_{\bar{V}_a^{-1}(t)}$
  Observe reward r_t and update parameters $\hat{\theta}_a(t)$ and $\bar{V}_a^{-1}(t)$ such that:

$$\bar{V}_{a_t}(t+1) = \bar{V}_{a_t}(t) + x_t x_t^\intercal, \quad b_{a_t}(t+1) = b_{a_t}(t) + r_t x_t, \quad \theta_{a_t}(t+1) = \bar{V}_{a_t}^{-1}(t+1) b_{a_t}(t+1)$$

end for

Algorithm 9: Linear Thompson Sampling with Gaussian prior

Input: regularization λ, number of arms K, number of rounds T, variance υ
Initialize for every arm a, $\bar{V}_a^{-1}(t) = \lambda I_d$ and $\hat{\theta}_a(t) = 0$, $b_a(t) = 0$
for $t = 1, ..., T$ **do**
  Observe context x_t
  Draw $\tilde{\theta}_a \sim \mathcal{N}(\hat{\theta}_a(t), \upsilon^2 \bar{V}_a^{-1}(t))$
  Pull arm $a_t = \text{argmax}_{a \in [\![1,K]\!]} \left\langle \tilde{\theta}_a, x_t \right\rangle$
  Observe reward r_t and update parameters $\hat{\theta}_a(t)$ and $\bar{V}_a^{-1}(t)$

$$\bar{V}_{a_t}(t+1) = \bar{V}_{a_t}(t) + x_t x_t^\intercal, \quad b_{a_t}(t+1) = b_{a_t}(t) + r_t x_t, \quad \theta_{a_t}(t+1) = \bar{V}_{a_t}^{-1}(t+1) b_{a_t}(t+1)$$

end for

C.F Semi-Online Attacks

Liu and Shroff [2019] studies what they call the offline setting for adversarial attacks on stochastic bandits. They consider a setting where a bandit algorithm is successively updated with mini-batches of fixed size B. The attacker can tamper with some of the incoming mini-batches. More precisely, they can modify the context, the reward and even the arm that was pulled for any entry of the attacked mini-batches. The main difference between this type of attacks and the online attacks we considered in the main paper is that we do not assume that we can attack from the start of the learning process: the bandit algorithm may have already converged by the time we attack.

We can still study the cumulative cost for the attacker to change the mini-batch in order to fool a bandit algorithm to pull a target arm $a^\dagger$ (here we take $A^\dagger = \{a^\dagger\}$). Contrarily to Liu and Shroff [2019], we call this setting semi-online. We first study the impact of an attacker on LinUCB where we show that, by modifying only $(K-1)d$ entries from the batch $\mathcal{B}$, the attacker can force LinUCB to pull arm $a^\dagger$, $M'B - o(M'B)$ times with M' the number of remaining batches updates. The cost of our attack is $\sqrt{MB}$ with M the total number of batches.

Algorithm 10: ε-GREEDY

Input: regularization λ, number of arms K, number of rounds T, exploration parameter $(\varepsilon)_t$
Initialize, for all arms a, $\bar{V}_a^{-1}(t) = \lambda I_d$ and $\hat{\theta}_a(t) = 0$, $\varepsilon_t = 1$, $b_a(t) = 0$
for $t = 1, ..., T$ **do**
  Observe context x_t
  With probability ε_t, pull $a_t \sim \mathcal{U}(\llbracket 1, K \rrbracket)$, or pull $a_t = \text{argmax}\langle \theta_a, x_t \rangle$
  Observe reward r_t and update parameters $\hat{\theta}_a(t)$ and $\bar{V}_a^{-1}(t)$

$$\bar{V}_{a_t}(t+1) = \bar{V}_{a_t}(t) + x_t x_t^\intercal, \quad b_{a_t}(t+1) = b_{a_t}(t) + r_t x_t,$$
$$\theta_{a_t}(t+1) = \bar{V}_{a_t}^{-1}(t+1) b_{a_t}(t+1)$$

end for

Algorithm 11: EXP4

Input: number of arms K, experts: $(E_m)_{m \in \llbracket 1, N \rrbracket}$, parameter η
Set $Q_1 = (1/N)_{j \in \llbracket 1, N \rrbracket}$
for $t = 1, ..., T$ **do**
  Observe context x_t and probability recommendation $(E_m^{(t)})_{m \in \llbracket 1, N \rrbracket}$
  Pull arm $a_t \sim P_t$ where $P_{t,j} = \sum_{k=1}^N Q_{t,k} E_{j,k}^{(t)}$
  Observe reward r_t and define for all arms i $\hat{r}_{t,i} = 1 - \mathbb{1}_{\{a_t = i\}}(1 - r_t)/P_{t,i}$
  Define $\tilde{X}_{t,k} = \sum_a E_{k,a}^{(t)} \hat{r}_{t,a}$
  Update $Q_{t+1,j} = \exp(\eta Q_{t,i}) / \sum_{j=1}^N \exp(\eta Q_{t,j})$ for all experts i
end for

Cost of an attack: If presented with a mini-batch $\mathcal{B}$, with elements (x_t, a_t, r_t) composed of the context x_t presented at time t, the action taken a_t and the reward received r_t, the attacker modifies element i, namely (x_t^i, a_t^i, r_t^i) into $(\tilde{x}_t^i, \tilde{a}_t^i, \tilde{r}_t^i)$. The cost of doing so is $c_t^i = ||x_t^i - \tilde{x}_t^i||_2 + \left|\tilde{r}_t^i - r_t^i\right| + \mathbb{1}_{\{a_t^i \neq \tilde{a}_t^i\}}$ and the total cost for mini-batch $\mathcal{B}$ is defined as $c_{\mathcal{B}} = \sum_{i \in \mathcal{B}} c_t^i$. Finally, we consider the cumulative cost of the attack over M different mini-batches $\mathcal{B}_1, \ldots, \mathcal{B}_M$, $\sum_{l=1}^M c_{\mathcal{B}_l}$. The interaction between the environment, the attacker and the learning algorithm is summarized in Alg. 12.

The attack presented here is based on the Ahlberg–Nilson–Varah bound Varah [1975], which gives a control on the sup norm of a matrix with dominant diagonal elements. More precisely, when presented with a mini-batch $\mathcal{B}$, the attacker needs to modify the contexts and the rewards. We assume that the attacker knows the number of mini-batch updates M and has access to a lower-bound on the reward of the target arm, ν as in Assumption 3.

The attacker changes $(K-1) \times d$ rows of the first mini-batch to rewards of 0 with a context $\delta_a e_i$ for each arm $a \neq a^\dagger$ with (e_i) the canonical basis of $\mathbb{R}^d$. Moreover, δ_a is chosen such that:

$$\delta_a > \max\left(\sqrt{\frac{2MBL^2d}{\nu} + dMB}, \sqrt{\frac{4\beta_{max}^2 L^2 d}{\nu^2} + dMB}\right) \tag{C.13}$$

with $\beta_{max} = \max_{t=0}^{MB} \beta_a(t)$ and M the number of mini-batch updates.

Proposition 20. *After the first attack, with probability* $1 - \delta$, LINUCB *always pulls arm* $a^\dagger$,

Proof. After having poisoned the first mini-batch $\mathcal{B}$, the latter can be partitioned into two

Algorithm 12: Semi-Online Attack Setting.

Input: Bandit alg. $\mathfrak{A}$, size of a mini-batch: B
Set $t = 0$
while True **do**
  $\mathfrak{A}$ observe context x_t
  $\mathfrak{A}$ pulls arm a_t and observes reward r_t
  Interaction (x_t, a_t, r_t) is saved in mini-batch $\mathcal{B}$
  if $\left|\mathcal{B}\right| = B$ **then**
    Attacker modifies mini-batch $\mathcal{B}$ into $\tilde{\mathcal{B}}$
    Update alg. $\mathfrak{A}$ with poisoned mini-batch $\tilde{\mathcal{B}}$
  end if
end while

subsets, $\mathcal{B}_c$ (with non-perturbed rows) and $\mathcal{B}_{nc}$ (with the poisoned rows). The design matrix of arm $a \neq a^\dagger$ for every time t after the poisoning is:

$$V_{t,a} = \lambda I_d + \sum_{l=1, a_l=a}^{t} x_l x_l^\intercal + \delta_a^2 \sum_{i=1}^{d} e_i e_i^\intercal \tag{C.14}$$

For every time t, non diagonal elements of $V_{t,a} = (v_{i,j})_{i,j}$ are bounded by:

$$\forall i, r_i := \sum_{j \neq i} v_{i,j} \leq \sum_{j \neq i} \sum_{l=1, a_l=a}^{t} \|x_l x_l^\intercal\|_\infty \leq dN_a(kB) \tag{C.15}$$

Whereas for all diagonal elements, $v_{i,i} \geq \delta_a^2 > r_i$. Thus $V_{t,a}$ is strictly diagonal dominant and by the Ahlberg–Nilson–Varah bound Varah [1975]:

$$\|V_{t,a}^{-1}\|_\infty \leq \frac{1}{\min_i \left(\|v_{i,i}\| - r_i\right)} \leq \frac{1}{\delta_a^2 - dMB} \tag{C.16}$$

Then, for every arm $a \neq a^\dagger$ and any context $x \in \mathcal{D}$ and any time t after the attack:

$$\begin{aligned}
\langle \hat{\theta}_{a,t}, x \rangle + \beta_a(t) ||x||_{V_{t,a}^{-1}} &\leq \sum_{l=1, a_l=a}^{t} r_t (V_{t,a}^{-1} x_t)^\intercal x + \beta_a(t) ||x||_1 \sqrt{\|V_{t,a}^{-1}\|_\infty} \\
&\leq \|V_{t,a}^{-1}\|_\infty dN_t(a) \sup_{y \in \mathcal{D}} ||y||_2^2 + \beta_{\max} \sqrt{d} \sup_{y \in \mathcal{D}} ||y||_2 \sqrt{\|V_{t,a}^{-1}\|_\infty} < \nu
\end{aligned}$$

We have shown that for any arm $a \neq a^\dagger$ and any time step t after the attack, the upper confidence bound computed by LinUCB is upper-bounded bu ν the arm $a^\dagger$. Then, with probability $1 - \delta$, the confidence set for arm $a^\dagger$ holds and, for all $x \in \mathcal{D}$, arm $a^\dagger$ is chosen by LinUCB. The total cost of this attack is $d \sum_{a \neq a^\dagger} \delta_a L = O(\sqrt{MB})$ □

Appendix D

ROPUST

ROPUST: Improving Robustness through Fine-tuning with Photonic Processors and Synthetic Gradients

Alessandro Cappelli [1] **Julien Launay** [1 2] **Laurent Meunier** [3 4] **Ruben Ohana** [1 2] **Iacopo Poli** [1]

Abstract

Robustness to adversarial attacks is typically obtained through expensive adversarial training with Projected Gradient Descent. We introduce ROPUST, a remarkably simple and efficient method to leverage robust pre-trained models and further increase their robustness, at no cost in natural accuracy. Our technique relies on the use of an Optical Processing Unit (OPU), a photonic co-processor, and a fine-tuning step performed with Direct Feedback Alignment, a synthetic gradient training scheme. We test our method on nine different models against four attacks in RobustBench, consistently improving over state-of-the-art performance. We also introduce phase retrieval attacks, specifically designed to target our own defense. We show that even with state-of-the-art phase retrieval techniques, ROPUST is effective.

1. Introduction

Adversarial examples (Goodfellow et al., 2015) threaten the safety and reliability of machine learning models deployed in the wild. Because of the sheer number of attack and defense scenarios, robustness can be difficult to evaluate (Bubeck et al., 2019). Standardized benchmarks, such as RobustBench (Croce et al., 2020) with AutoAttack (Croce & Hein, 2020b), have helped better evaluate progress in the field. The development of defense-specific attacks is also crucial (Tramèr & Boneh, 2019). To date, one of the most effective defense techniques remains adversarial training with Projected Gradient Descent (PGD) (Madry et al., 2018). Adversarial training can be resource-consuming, but robust networks pre-trained with PGD are now widely available, motivating their use as a foundation for simple and widely applicable defenses that further enhance their robustness.

To this end, we introduce **ROPUST**, a drop-in replacement for the classifier of already robust models. Our defense leverages a photonic co-processor (the Optical Processing Unit, OPU) for physical *parameter obfuscation* (Cappelli et al., 2021): because the *fixed* random parameters are optically implemented, they remain unknown at training and inference time. Additionally, a synthetic gradient method, Direct Feedback Alignment (DFA) (Nøkland, 2016), is used to fine-tune the ROPUST classifier.

We evaluate our method against AutoAttack on nine different models in RobustBench, and consistently improve robust accuracies over the state-of-the-art (Fig. 1). We also develop a *phase retrieval* attack targeting our parameter obfuscation, and show that ROPUST remains effective.

1.1. Related work

Attacks. Adversarial attacks have been framed in a variety of settings: white-box, where the attacker is assumed to have unlimited access to the model, including its parameters (e.g. FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018; Kurakin et al., 2016), Carlini & Wagner (Carlini & Wagner, 2017)); black-box, assuming only limited access to the network for the attacker, with methods attempting to estimate the gradients (Chen et al., 2017; Ilyas et al., 2018a;b), or derived from genetic algorithms (Andriushchenko et al., 2019; Meunier et al., 2019) and combinatorial optimization (Moon et al., 2019); transfer attacks, where an attack is crafted on a model that is accessible to the attacker, and then applied to the target network (Papernot et al., 2016). Automated schemes, such as AutoAttack (Croce & Hein, 2020b), have been proposed to autonomously select attacks and tune their hyperparameters.

Defenses. Adversarial training adds adversarial robustness as an explicit training objective (Goodfellow et al., 2015; Madry et al., 2018), by incorporating adversarial examples during the training. Theoretically grounded defenses have been proposed (Lecuyer et al., 2018; Cohen et al.; Alexandre Araujo & Negrevergne, 2020; Pinot et al., 2019; Wong et al., 2018; Wong & Kolter, 2018), but these fail to match the clean accuracy of state-of-the-art networks. Many empir-

[1]LightOn, France [2]LPENS, École Normale Supérieure, France [3]Facebook AI Research, France [4]Université Paris-Dauphine, France. Correspondence to: Alessandro Cappelli <alessandro@lighton.ai>.

Accepted by the ICML 2021 workshop on A Blessing in Disguise: The Prospects and Perils of Adversarial Machine Learning.

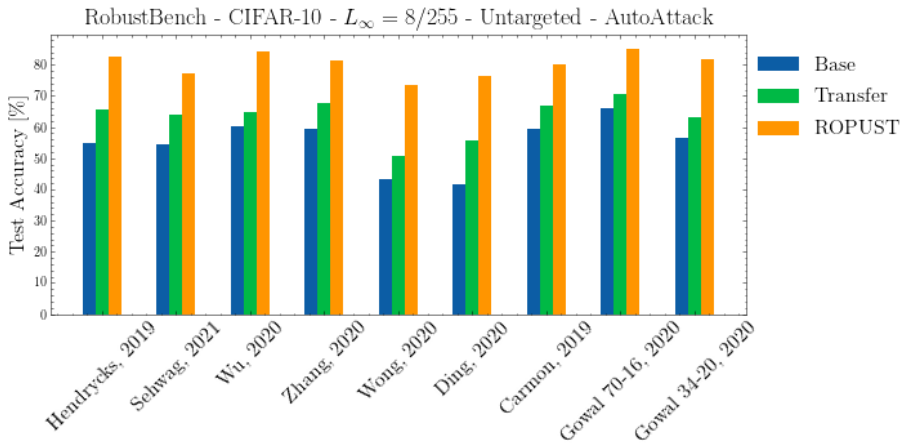


Figure 1. **ROPUST systematically improves the test accuracy of already robust models**. Transfer refers to the performance when attacks are generated on the base model and transferred to the ROPUST model. Models from the RobustBench model zoo: Hendrycks, 2019 (Hendrycks et al., 2019), Sehwag, 2021 (Sehwag et al., 2021), Wu, 2020 (Wu et al., 2020), Zhang, 2020 (Zhang et al., 2020), Wong, 2020 (Wong et al., 2020), Ding, 2020 (Ding et al., 2020), Carmon, 2019 (Carmon et al., 2019), Gowal, 2020 (Gowal et al., 2020).

ical defenses have been criticized for providing a false sense of security (Athalye et al., 2018; Tramèr & Boneh, 2019), by not evaluating on attacks adapted to the defense. Gradient obfuscation, through the use of a non-differentiable activation function, has been proposed as a way to protect against white-box attacks (Papernot et al., 2017). However, it can be easily bypassed by Backward Pass Differentiable Approximation (BPDA) (Athalye et al., 2018), where the defense is replaced by a differentiable relaxation. *Parameter obfuscation* has been proposed with dedicated photonic co-processor (Cappelli et al., 2021). However, by itself, this kind of defense falls short of adversarial training.

Fine-tuning and analog computing. Previous work introduced *adversarial fine-tuning* (Jeddi et al., 2020): fine-tuning a non-robust model with an adversarial objective. In this work instead we fine-tune a robust model without adversarial training. Additionally, it was shown that robustness improves transfer performance (Salman et al., 2020) and that robustness transfers across datasets (Shafahi et al., 2020). The advantage of non-ideal analog computations in terms of robustness has been investigated in the context of NVM crossbars (Roy et al., 2020).

1.2. Motivations and contributions

We propose to simplify and extend the applicability of photonic-based parameter obfuscation defenses. The use of dedicated hardware to perform the random projection physically guarantees *parameter obfuscation* (Cappelli et al., 2021). Our defense, ROPUST, can be dropped-in to supplement any robust pre-trained model and fine-tuning its classifier is fast. In contrast with existing parameter-obfuscation methods, it leverages pre-trained robust models, and achieves state-of-the-art performance. Drawing inspiration from the field of phase retrieval, we introduce a new kind of attack against defenses relying on parameter obfuscation, *phase retrieval attacks*. We show that ROPUST remains robust even against state-of-the-art retrieval methods.

2. Methods

2.1. Automated adversarial attacks

We evaluate our model against the four attacks implemented in RobustBench: APGD-CE and APGD-T (Croce & Hein, 2020b), Square attack (Andriushchenko et al., 2019), and Fast Adaptive Boundary (FAB) attack (Croce & Hein, 2020a). We describe these attacks more in detail in the supplementary. In RobustBench, using AutoAttack, given a batch of samples, these are first attacked with APGD-CE. Then, the samples that were successfully attacked are discarded, and the remaining ones are attacked with APGD-T. This procedure continues with Square and FAB attack.

2.2. Our defense

Optical Processing Units. Optical Processing Units (OPU)[1] are photonic co-processors dedicated to efficient large-scale random projections (Ohana et al., 2020). Assuming an input vector $\mathbf{x}$, the OPU computes the following operation using light scattering through a diffusive medium:

$$\mathbf{y} = |\mathbf{U}\mathbf{x}|^2 \tag{1}$$

[1]Accessible at `https://cloud.lighton.ai`.

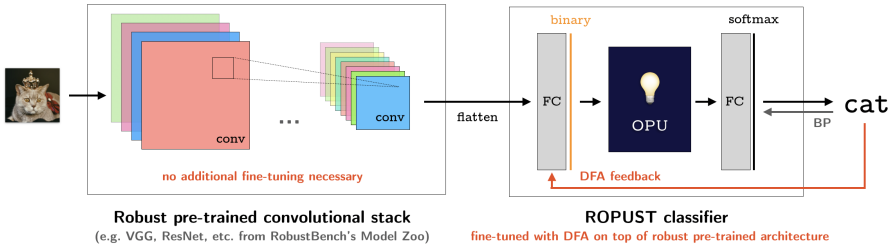


Figure 2. **ROPUST replaces the classifier of already robust models, enhancing their adversarial robustness.** Only the ROPUST classifier needs fine-tuning; the convolutional stack is frozen. Convolutional features first go through a fully-connected layer, before binarization for use in the Optical Processing Unit (OPU). The OPU performs a non-linear random projection, with *fixed unknown parameters*. A fully-connected layer is then used to obtain a prediction from the output of the OPU. Direct Feedback Alignment is used to train the layer underneath the OPU.

With $\mathbf{U}$ a *fixed* complex Gaussian random matrix of size up to $10^6 \times 10^6$, which entries are not readily known. In the following, we sometimes refer to $\mathbf{U}$ as the *transmission matrix* (TM). The input is binary and the output is in 8 bits.

The matrix $\mathbf{U}$ is physically implemented through the diffusive medium. As only the non-linear intensity $|\mathbf{Ux}|^2$ can be measured, an attacker has to perform *phase retrieval* to retrieve the coefficients of $\mathbf{U}$. We develop such an attack scenario in Section 4.

Direct Feedback Alignment. Because the fixed random parameters implemented by the OPU are unknown, it is impossible to backpropagate through it. We bypass this limitation by training layers upstream of the OPU using Direct Feedback Alignment (DFA) (Nøkland, 2016).

In a fully connected network, at layer i out of N, neglecting biases, with $\mathbf{W}_i$ its weight matrix, f_i its non-linearity, and $\mathbf{h}_i$ its activations, the forward pass can be written as $\mathbf{a}_i = \mathbf{W}_i\mathbf{h}_{i-1}, \mathbf{h}_i = f_i(\mathbf{a}_i)$. $\mathbf{h}_0 = X$ is the input data, and $\mathbf{h}_N = f(\mathbf{a}_N) = \hat{\mathbf{y}}$ are the predictions. A task-specific cost function $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$ is computed to quantify the quality of the predictions with respect to the targets $\mathbf{y}$. The weight updates are obtained through:

$$\delta\mathbf{W}_i = -\frac{\partial\mathcal{L}}{\partial\mathbf{W}_i} = -[(\mathbf{W}_{i+1}^T\delta\mathbf{a}_{i+1}) \odot f_i'(\mathbf{a}_i)]\mathbf{h}_{i-1}^T \quad (2)$$

where $\odot$ is the Hadamard product. With DFA, the gradient signal $\mathbf{W}_{i+1}^T\delta\mathbf{a}_{i+1}$ of the (i+1)-th layer is replaced with a random projection of the gradient of the loss at the top layer $\delta\mathbf{a}_y$–which is the error $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$ for the cross-entropy loss:

$$\delta\mathbf{W}_i = -[(\mathbf{B}_i\delta\mathbf{a}_y) \odot f_i'(\mathbf{a}_i)]\mathbf{h}_{i-1}^T, \delta\mathbf{a}_y = \frac{\partial\mathcal{L}}{\partial\mathbf{a}_y} \quad (3)$$

ROPUST We propose to replace their classifier with the ROPUST module to enhance the adversarial robustness of pretrained robust models (Fig. 2). We use robust models from the RobustBench model zoo, extracting and freezing their convolutional stack. The robust convolutional features go through a fully connected layer and a sign function, preparing them for the OPU. The OPU then performs a non-linear random projection, with fixed unknown parameters. The predictions are obtained through a final fully-connected layer. While the convolutional layers are frozen, we train the ROPUST module on natural data using DFA to bypass the non-differentiable photonic hardware.

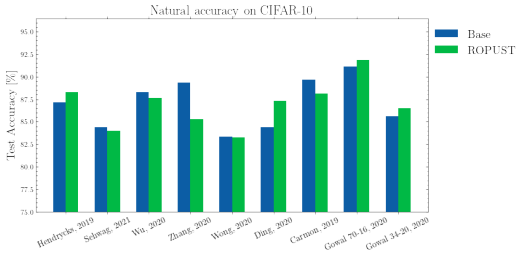


Figure 3. **Our ROPUST defense comes at no cost in natural accuracy.** In some cases, natural accuracy is even improved. The model from Zhang, 2020 (Zhang et al., 2020) is an isolated exception. The papers related to each model are cited in Fig. 1.

Attacking ROPUST. Previous work has shown that methods devoid of weight transport are not effective in generating compelling adversarial examples (Akrout, 2019). Therefore, we use backward pass differentiable approximation (BPDA) in place of DFA when attacking our defense: we relax non-differentiable layers to a differentiable version. For the binarization function, we simply use the derivative of $\tanh$ in the backward pass, while we approximate the transpose of the obfuscated parameters with a different fixed random matrix drawn at initialization of the module.

3. Evaluating ROPUST on RobustBench

All of the attacks are performed on CIFAR-10 (Krizhevsky, 2009), using a differentiable backward pass approximation

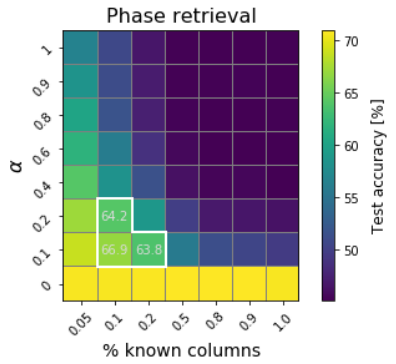


Figure 4. **Performance of an APGD-CE attack with a retrieved matrix in place of the, otherwise unknown, transpose of the transmission matrix.** A better knowledge of the transmission matrix correlates with the success of the attack, with a sharp phase transition. It may seem that even a coarse-grained knowledge of the TM can help the attacker. However, even state-of-the-art phase retrieval methods operate only in the white contoured region, where the robustness is still greater than the *Base* models. We highlighted the accuracies achieved under attack in this region.

(Athalye et al., 2018) as explained in Section 2.2. For our experiments, we use OPU input size 512 and output size 8000. We use the Adam optimizer (Kingma & Ba, 2014), with learning rate 0.001, for 10 epochs. The process typically takes 10 minutes on a single NVIDIA V100 GPU.

We show our results on nine different models in RobustBench in Fig. 1. The performance of the original pretrained models from the RobustBench leaderboard is reported as *Base*. *ROPUST* represents the same models equipped with our defense. Finally, *Transfer* indicates the performance of attacks created on the original model and transferred to fool the ROPUST defense. For all models considered, ROPUST improves the robustness significantly, even under transfer. For transfer, we also tested crafting the attacks on the *Base* model while using the loss of the ROPUST model for the learning rate schedule of APGD. We also tried to use the predictions of ROPUST, instead of the base model, to *remove* the samples that were successfully attacked from the next stage of the ensemble; however, these modifications did not improve transfer performance. We remark that the robustness increase typically comes at no cost in natural accuracy; we show the accuracy on natural data of the *Base* and the *ROPUST* models in Fig. 3. We ablate our defense against white-box attacks in the supplementary.

4. Phase retrieval attack

Our defense leverages parameter obfuscation to achieve robustness. Yet, however demanding, it is still technically possible to recover the parameters through phase retrieval schemes (Gupta et al., 2019; 2020). To provide a thorough and fair evaluation of our attack, we study in this section *phase retrieval* attacks. We first consider an idealized setting, and then confront this setting with a real-world phase retrieval algorithm from (Gupta et al., 2020).

Ideal retrieval model. We build an idealized phase retrieval attack, where the attacker knows a certain fraction of columns, up to a certain precision. We model the retrieved matrix $\mathbf{U}'$ as a linear interpolation of the real transmission matrix $\mathbf{U}$ and a completely different random matrix $\mathbf{R}$. In practice, this model is valid only for a certain fraction of columns, and the remaining ones are modeled as independent random vectors. We can model this with a Boolean mask matrix $\mathbf{M}$, so our retrieval model in the end is:

$$\mathbf{U}' = \alpha \mathbf{U} \odot \mathbf{M} + (1 - \alpha)\mathbf{R} \tag{4}$$

In this setting, we vary the knowledge of the attacker from the minimum to the maximum by varying α and the percentage of retrieved columns, and we show how the performance of our defense changes in Fig. 4. In this simplified model only a crude knowledge of the parameters seems sufficient, given the sharp phase transition. We now need to chart where state-of-the-art retrieval methods are on this graph to estimate their ability to break our defense.

Real-world retrieval performance. State-of-the-art phase retrieval methods seek to maximize output correlation, i.e. the correlation on $\mathbf{y}$ in Eq. 1, in place of the correlation with respect to the parameters of the transmission matrix, i.e. $\mathbf{U}$ in Eq. 1. We find this is a significant limitation for attackers. In Fig. 4, following numerical experiments, we highlight with a white contour the operating region of a state-of-the-art phase retrieval algorithm (Gupta et al., 2020), showing that it can manage to only partially reduce the robustness of ROPUST.

5. Conclusion

We introduced ROPUST, a drop-in module to enhance the adversarial robustness of pretrained already robust models. Our technique relies on parameter obfuscation guaranteed by a photonic co-processor, and a synthetic gradient method: it is simple, fast and widely applicable.

We thoroughly evaluated our defense on nine different models in the standardized RobustBench benchmark, reaching state-of-the-art performance. In light of these results, we encourage to extend RobustBench to include parameter obfuscation methods.

Finally, we developed a new kind of attacks, *phase retrieval attacks*, specifically suited to parameter obfuscation defense such as ours, and we tested their effectiveness. We found that the typical precision regime of even state-of-the-art phase retrieval methods is not enough to completely break ROPUST.

References

Akrout, M. On the adversarial robustness of neural networks without weight transport. *ArXiv*, abs/1908.03560, 2019.

Alexandre Araujo, Laurent Meunier, R. P. and Negrevergne, B. Advocating for multiple defense strategies against adversarial examples. *Workshop on Machine Learning for CyberSecurity (MLCS@ECML-PKDD)*, 2020.

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. *arXiv preprint arXiv:1912.00049*, 2019.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Dy, J. and Krause, A. (eds.), *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.

Bubeck, S., Lee, Y. T., Price, E., and Razenshteyn, I. Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pp. 831–840. PMLR, 2019.

Cappelli, A., Ohana, R., Launay, J., Meunier, L., Poli, I., and Krzakala, F. Adversarial robustness by design through analog computing and synthetic gradients. *ArXiv*, abs/2101.02115, 2021.

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.

Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., and Duchi, J. C. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26. ACM, 2017.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*.

Croce, F. and Hein, M. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020a.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020b.

Croce, F., Andriushchenko, M., Sehwag, V., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Ding, G. W., Sharma, Y., Lui, K. Y.-C., and Huang, R. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. *ArXiv*, abs/1812.02637, 2020.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

Gowal, S., Qin, C., Uesato, J., Mann, T. A., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *ArXiv*, abs/2010.03593, 2020.

Gupta, S., Gribonval, R., Daudet, L., and Dokmanić, I. Don't take it lightly: Phasing optical random projections with unknown operators. In *NeurIPS*, 2019.

Gupta, S., Gribonval, R., Daudet, L., and Dokmani'c, I. Fast optical system identification by numerical interferometry. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1474–1478, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. *ArXiv*, abs/1603.05027, 2016.

Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018a.

Ilyas, A., Engstrom, L., and Madry, A. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018b.

Jeddi, A., Shafiee, M., and Wong, A. A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning. *ArXiv*, abs/2012.13628, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

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.

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.

Meunier, L., Atif, J., and Teytaud, O. Yet another but more efficient black-box adversarial attack: tiling and evolution strategies. *arXiv preprint arXiv:1910.02244*, 2019.

Moon, S., An, G., and Song, H. O. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4636–4645, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/moon19a.html.

Nøkland, A. Direct feedback alignment provides learning in deep neural networks. In *NIPS*, 2016.

Ohana, R., Wacker, J., Dong, J., Marmin, S., Krzakala, F., Filippone, M., and Daudet, L. Kernel computations from large-scale random features obtained by optical processing units. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 9294–9298. IEEE, 2020.

Papernot, N., McDaniel, P., and Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Pinot, R., Meunier, L., Araujo, A., Kashima, H., Yger, F., Gouy-Pailler, C., and Atif, J. Theoretical evidence for adversarial robustness through randomization: the case of the exponential family. *arXiv preprint arXiv:1902.01148*, 2019.

Roy, D., Chakraborty, I., Ibrayev, T., and Roy, K. Robustness hidden in plain sight: Can analog computing defend against adversarial attacks? *arXiv preprint arXiv:2008.12016*, 2020.

Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., and Madry, A. Do adversarially robust imagenet models transfer better? *ArXiv*, abs/2007.08489, 2020.

Sehwag, V., Mahloujifar, S., Handina, T., Dai, S., Xiang, C., Chiang, M., and Mittal, P. Improving adversarial robustness using proxy distributions. *ArXiv*, abs/2104.09425, 2021.

Shafahi, A., Saadatpanah, P., Zhu, C., Ghiasi, A., Studer, C., Jacobs, D., and Goldstein, T. Adversarially robust transfer learning. *ArXiv*, abs/1905.08232, 2020.

Tramèr, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems*, pp. 5866–5876, 2019.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295, 2018.

Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, pp. 8400–8409, 2018.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. *ArXiv*, abs/2001.03994, 2020.

Wu, D., Xia, S., and Wang, Y. Adversarial weight perturbation helps robust generalization. *arXiv: Learning*, 2020.

Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M. Geometry-aware instance-reweighted adversarial training. *ArXiv*, abs/2010.01736, 2020.

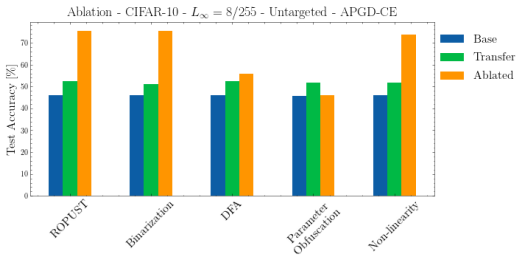


Figure 5. **Removing either parameter obfuscation or DFA from our defense causes a large drop in accuracy.** Robustness is given by the inability to efficiently generate attacks in a white-box settings when the parameters are obfuscated, and DFA is capable of generating partially robust features. Even though the non-linearity $|.|^2$ does not contribute to robustness, it is key to obfuscation, preventing trivial retrieval. Transfer performance does not change much when removing components of the defense. The performance of the `Base` is shown for comparison.

Appendix

Description of the attacks in AutoAttack

APGD-CE is a standard PGD where the step size is tuned using the loss trend information, squeezing the best performance out of a limited iterations budget. APGD-T, on top of the step size schedule, substitutes the cross-entropy loss with the Difference of Logits Ratio (DLR) loss, reducing the risk of vanishing gradients. Square attack is based on a random search. Random updates δ are sampled from an attack-norm dependent distribution at each iteration: if they improve the objective function they are kept, otherwise they are discarded. FAB attack aims at finding adversarial samples with minimal distortion with respect to the attack point. With respect to PGD, it does not need to be restarted and it achieves fast good quality results.

Ablation study: white-box setting

We perform an ablation study and find that the robustness of our defense against white-box attacks comes from both *parameter obfuscation* and DFA. We use the model from (Wong et al., 2020) available in the RobustBench model zoo. It consists in a PreAct ResNet-18 (He et al., 2016), pretrained with a *"revisited"* FGSM of increased effectiveness. We conduct the ablation study by removing a single component of our defense at a time in simulation: binarization, DFA, parameter obfuscation, and non-linearity $|.|^2$ of the random projection. To remove DFA, we also remove the binarization step and train the ROPUST module with backpropagation, since we have access to the transpose of the transmission matrix in the simulated setting of the ablation study. We show the results in Fig. 5: removing the non-linearity $|.|^2$ and the binarization does not have an effect, with the robustness given by *parameter obfuscation* and DFA.

Impact statement

Adversarial attacks have been identified as a significant threat to applications of machine learning in-the-wild. Developing simple and accessible ways to make neural networks more robust is key to mitigating some of the risks and making machine learning applications safer. More robust models would enable a wider range of business applications, especially in safety-critical sectors. We do not foresee negative societal impacts of our work, beyond the risk of our defense being broken by future developments in adversarial attacks. A limit of our work is that we prove increased robustness only empirically and not theoretically. However, theoretically grounded defense methods typically fall short of other techniques more used in practice. We rely on photonic hardware accessible by anyone, similarly to GPUs or TPUs on commercial cloud providers. We performed all of our experiments on single-GPU nodes with NVIDIA V100, and an OPU, on a cloud provider. We estimate a total of ~ 500 GPU hours was spent.