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 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., 2014a]**: 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. [2020]. 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

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

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 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., 2014a] 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.

[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.

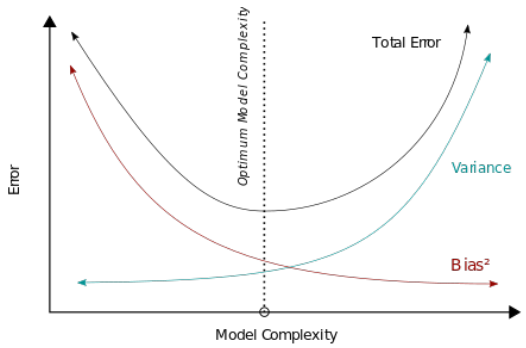


Figure 1.1: Bias-Variance tradeoff. A model with low complexity will have a low variance but an high bias. A model with high complexity will have a low bias but an high variance.

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 $\{(x_1, y_1), \dots, (x_N, y_N)\}$ 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 $\{(x'_1, y'_1), \dots, (x'_m, y'_m)\}$ 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, $(f_1(x), \dots, f_K(x))$ 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 $\Omega_{\mathcal{H}}(f)$ 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, deepen our understanding of machine learning/deep learning in high dimensional regimes.

1.3.2 Classification in Presence of Adversarial Attacks

Yet 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., 2014a]. 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 [**?**].

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 a 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 lack 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 point of views 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 3, 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.

This work is not submitted yet. We plan to submit it in the coming months

1.4.3 Building Certifiable Models

The last problem we deal with in this manuscript is the implementation of robust certifiable models. 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. This work is not submitted yet. We plan to submit it in the coming months

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 ??):** 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 ??):** 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 Standard Classification in a Nutshell

2.1.1 Notations

We recall that a classification aims at assigning a label to a given input. We formalize it as follows:

- 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 image $x \in \mathcal{X}$ has to be associated with a label y. We designate the set of labels $\mathcal{Y} := \{1, \dots, K\}$ as descriptors of an input. For instance the label of an image will be its description. $\mathcal{Y}$ will be 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 space $(\mathcal{X} \times \mathcal{Y}, d \oplus d')$ is also a proper Polish space.

In classification problem in machine learning, the data is assumed to be sampled from an unknown probability distribution $\mathbb{P}$. We will assume from now that the distributions we consider are Borel. 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 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.

2.1.2 Classification Task in Supervised Learning

In standard classification, it is usual to maximize the accuracy of the classifier, or equivalently, to minimize 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:*

$$\mathcal{R}_{0/1,\mathbb{P}}(h) := \mathbb{P}(h(x) \neq y) = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\mathbf{1}_{h(x)\neq y}\right] \tag{2.1}$$

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

$$\mathcal{R}^\star_{0/1,\mathbb{P}} := \inf_{h \in \mathcal{F}(\mathcal{X},\mathcal{Y})} \mathcal{R}_{0/1,\mathbb{P}}(h) \tag{2.2}$$

Note that this quantity is well defined when h is measurable. The optimal classifier is called the Bayes Optimal classifier and is defined as $h(x) = \operatorname{argmax}_k \mathbb{P}(y = k \mid x)$. We remark that the disintegration theorem ensures that h is indeed Borel measurable.

In practice, the access to the Bayes Optimal 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 constitute the training set. The knowledge of the Bayes classifier on training points is not sufficient to generalize on points out of the training set. Thus the learner needs to learn the classifier over smaller classes of functions. Secondly, the 0/1 loss is not convex neither continuous,

The two main questions in classification are the following:

Hence one need to reduce the search space of measurable functions to a much smaller one, denoted $\mathcal{H}$ in the sequel.

to have generalization properties for the classifier on out-of-sample data points because such functions would overfit the training set. Hence one need to reduce the search space of measurable functions to a much smaller one, denoted $\mathcal{H}$ in the sequel. More precisely, for binary classification (i.e $\mathcal{Y} := \{-1, +1\}$), we aim at learning a function $\mathbf{f} : \mathcal{X} \to \mathbb{R}$ such that $h(x) = \operatorname{sign}(f(x))$ (with a convention on $sign(0)$). In multilabel classification (i.e $|\mathcal{Y}| \geq 2$), we learn a function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^K$ and h is set to $h(x) = \operatorname{argmax}_k f_k(x)$. Minimizing directly the 0/1 loss risk is NP-hard CITE. Then one needs to minimize a well-chosen surrogate loss function L. A *loss function* $L : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ will be without loss of generality a non negative Borel measurable function. An example of such a loss is the cross entropy loss defined as:

$$L(\mathbf{f}(x), y) = -\sum_{i=1}^{K} \mathbf{1}_{y=i} \log f_i(x)$$

where $f_i(x)$ is the probability learnt by the model that input x belongs to the class i. The study of which loss is suited for classification has been a widely studied topic [Bartlett et al., 2006, Steinwart, 2007]. Hence the learning objective is then defined as:

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

Usual Datasets in Image Classification In this PhD thesis, although the results are general and applicable to every type of data, we only use image classification To evaluate the perfomance 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).
- **CIFAR10 and CIFAR100 [Krizhevsky et al., 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 $32 \times 32 \times 3$ (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%.
- **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%. Further than the standard dataset, ImageNet project is still in development: the project gathers 14197122 images and 21841 labels on August 31th, 2021.

2.1.3 Surrogate losses, consistency and calibration

In classification, optimizing the risk associated 0/1 loss is a difficult task, generally an NP-hard problem **?**.

Definition 2 (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 $f \in \mathcal{F}(\mathcal{X})$.*

Similarly to risk associated with the 0/1 loss,

Definition 3 (L-risk of a classifier)**.** *For a given measurable loss L, and a Borel probability distribution $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$ we define the risk of 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:

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

The risk of a classifier is then defined as the average loss over the distribution $\mathbb{P}$. The loss L is difficult to optimize in practice: the loss might, for instance, not convex not differentiable or continuous. It is then often preferred to optimize a surrogate loss function instead. In the literature [Bartlett et al., 2006, Steinwart, 2007, **?**], the notion of surrogate losses has been studied as a consistency problem: 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 as follows.

Definition 4 (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 $(f_n)_n \in \mathcal{F}(\mathcal{X})^{\mathbb{N}}$:*

$$\mathcal{R}_{L_2,\mathbb{P}}(f_n) \to \mathcal{R}^\star_{L_2,\mathbb{P}} \implies \mathcal{R}_{L_1,\mathbb{P}}(f_n) \to \mathcal{R}^\star_{L_1,\mathbb{P}} \tag{2.3}$$

We say that a loss L_2 is said to be consistent with respect to a loss L_1 if consistency holds for every distribution $\mathbb{P}$.

2.1.4 Empirical Risk Minimization and Generalization

As mentioned earlier, the learner has access to training points

2.2 Game Theory in a Nutshell

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

2.2.1 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.

2.2.2 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 guarantees. We are going to formalize mathematically the meaning of a Nash equilibrium. 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. In there is duality but the optima are 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.

Computing Equilibria. The computation of Nash or Stackelberg equilbria highly depends on the setting of the game. For instance, the players might have or not knowledge of the other strategy or utility functions. In finite case settings there have been a large TODO

For instance, under complete knowledge of utilities and strategies, vanilla optimization problems

2.2.3 Strong Duality Theorems

To prove strong duality results, many results have been derived especially

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 **??** 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)**.** *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 hypothesis, namely Sion's Theorem and Fan's Theorem.

Theorem 2 (Sion's Theorem)**.** *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, the infimum is attained.

Theorem 3 (Fan's Theorem)**.** *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 hypothesis 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-$*$ topology of measures and use Sion's or Fan's Theorem directly on these probability spaces.

2.2.4 Computation of Nash Equilibria

TODO

2.3 Introduction to Adversarial Classification

2.3.1 What is an adversarial example?

Informally, an adversarial example is a perturbation of an input that is imperceptible to human senses, 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 5. *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:*

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

Definition 6. *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 as:*

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

We call adversarial Bayes risk, the infimum of adversarial risk over the set of Borel measurable classifiers $\mathcal{F}(\mathcal{X}, \mathcal{Y})$:

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

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 $\mathcal{R}_\varepsilon(h)$ is well defined.*

2.3.2 Adversarial examples in the wild

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:

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

To solve the previous optimization problem, 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, 2020a], FAB [Croce and Hein, 2020a], 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 and Hein, 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 [**?**] and NES attack [Ilyas et al., 2018, 2019].

of the attacker has no knowledge on the classifier parameters and a limited access to the classifier, e.g. he can only access logits or predicted class for instance. CITE ZOO, nes. square, yetanother

TO COMPLETE

Usual Datasets 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 et al., 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.

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

- **Image Segmentation**

- **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 frame or a lot only a few frames.

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

- **Audio classification**
- **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.
- **Recommender Systems** A recent line of work Garcelon et al. [2020], **?** 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.

2.3.3 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 $d(x, x') \leq \varepsilon$, $h(x') = y$.

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., 2014b, 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 2 (Tsuzuku et al. [2018])**.** *Let* $\mathbf{f}$ *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 17, 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.

Hence, 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., 2019b, Singla and Feizi, 2021, Trockman et al., 2021]. The work of Li et al. [2019b], Trockman et al. [2021] and Singla and Feizi [2021] (denoted BCOP, Cayley and SOC respectively) are examples of approaches that aims at building orthogonal weights in layers. Indeed, their 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. The BCOP layer (Block Convolution Orthogonal Parameterization) uses an iterative algorithm proposed by Björck et al. [1971] to orthogonalize the linear transform performed by a convolution. The SOC layer (Skew Orthogonal Convolutions) uses the property that if A is a skew symmetric matrix then $Q = \exp A$ is an orthogonal matrix. To approximate the exponential, the authors proposed to use a finite number of terms in its Taylor series expansion. Finally, the 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 $Q = (I - A)^{-1}(I + A)$. Both methods are well adapted to convolutional layers and are able to reach high accuracy levels on CIFAR datasets. 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.

2.4 Evaluating Adversarial Robustness

2.4.1 Evaluation Protocol

Unless defense mechanisms are provable and have guarantees, evaluating Adversarial Robustness is a complicated task. Many defenses introduced in the literature are actually "false" defenses. Indeed, one needs to adapt each attack to the defense. We describe the following common issues. CITE more

Evaluation of Randomized Classifiers. 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 nature of the classifier. To do so, Athalye et al. [2018a] proposed to average either the outputs or the gradient of the classifier to build an efficient attack. This procedure was called Expectation Over Transformation (EOT).

Benchmarking defenses. To answer the need of adversarial examples 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 attacks (AutoPGD [Croce and Hein, 2020b], FAB [Croce and Hein, 2020a], SquareAttack [Andriushchenko et al., 2019]). The platform does not propose to evaluate the robustness of randomized classifiers.

2.4.2 Theoretical knowledge in Adversarial classification

Curse of dimensionality.

Chapter 3

Game Theory of Adversarial Examples

Contents

3.1 The Adversarial Attack Problem

3.1.1 A Motivating Example

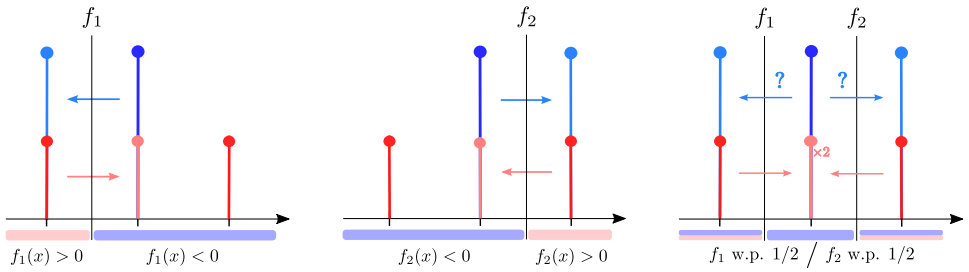


Figure 3.1: Motivating example: blue distribution represents label -1 and the red one, label $+1$. The height of columns represents their mass. The red and blue arrows represent the attack on the given classifier. On left: deterministic classifiers (f_1 on the left, f_2 in the middle) for whose, the 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 3.1. 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 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.

3.1.2 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.

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

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 [Bao et al., 2020, Cranko et al., 2019]; hence it is essential that the $0/1$ loss satisfies Assumption 1. In the binary classification setting (*i.e.* $\mathcal{Y} = \{-1, +1\}$) the $0/1$ loss writes $L_{0/1}(\theta, (x, y)) = \mathbf{1}_{yf_\theta(x) \leq 0}$. 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.

3.1.3 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:

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

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 $u \in \bar{\mathbb{R}}$:

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

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. □

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

$$\mathcal{V}_{rand}^\varepsilon := \inf_{\mu\in\mathcal{M}_+^1(\Theta)} \mathcal{R}^\varepsilon(\mu),\ \mathcal{V}_{det}^\varepsilon := \inf_{\theta\in\Theta} \mathcal{R}^\varepsilon(\theta) \tag{3.1}$$

Remark 1. *We can show (see Appendix 3.6.5) that the standard risk infima are equal : $\mathcal{V}_{rand}^0 = \mathcal{V}_{det}^0$. Hence, no randomization is needed for minimizing the standard risk. Denoting $\mathcal{V}$ this common value, we also have the following inequalities for any $\varepsilon > 0$, $\mathcal{V} \leq \mathcal{V}_{rand}^\varepsilon \leq \mathcal{V}_{det}^\varepsilon$.*

3.1.5 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.

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

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

Definition 7 (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*

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

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. Adversarial examples have been studied in the light of DRO by former works [Sinha et al., 2017, Tu et al., 2018], but an exact reformulation of the adversarial risk as a DRO problem has not been made yet. When $(\mathcal{Z}, d)$ is a Polish space and $c : \mathcal{Z}^2 \to \mathbb{R}^+ \cup \{+\infty\}$ is a lower semi-continuous function, for $\mathbb{P}, \mathbb{Q} \in \mathcal{M}_1^+(\mathcal{Z})$, the primal Optimal Transport problem is defined as

$$W_c(\mathbb{P}, \mathbb{Q}) := \inf_{\gamma \in \Gamma_{\mathbb{P},\mathbb{Q}}} \int_{\mathcal{Z}^2} c(z, z') d\gamma(z, z')$$

with $\Gamma_{\mathbb{P},\mathbb{Q}} := \big\{\gamma \in \mathcal{M}_1^+(\mathcal{Z}^2) \mid\ \Pi_{1\sharp}\gamma = \mathbb{P},\ \Pi_{2\sharp}\gamma = \mathbb{Q}\big\}$. When $\eta > 0$ and for $\mathbb{P} \in \mathcal{M}_1^+(\mathcal{Z})$, the associated Wasserstein uncertainty set is defined as:

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

A DRO problem is a linear optimization problem over Wasserstein uncertainty sets:

$$\sup_{\mathbb{Q} \in \mathcal{B}_c(\mathbb{P},\eta)} \int g(z) d\mathbb{Q}(z)$$

for some upper semi-continuous function g [Yue et al., 2020]. 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 3 shows, the Wasserstein ball associated with c_ε is equal to $\mathcal{A}_\varepsilon(\mathbb{P})$.

Proposition 3. *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 4. *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:*

$$\mathcal{R}^\varepsilon(\mu) = \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{(x',y') \sim \mathbb{Q}, \theta \sim \mu} \left[L(\theta, (x', y')) \right]. \tag{3.2}$$

The supremum is attained. Moreover $\mathbb{Q}^* \in \mathcal{A}_\varepsilon(\mathbb{P})$ is an optimum of Problem (3.2) 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: $\mathbb{E}_{\theta \sim \mu} \left[L(\theta, (x', y')) \right] \leq \sup_{(u,v),\ d(x,u) \leq \varepsilon, y = v} \mathbb{E}_{\theta \sim \mu} \left[L(\theta, (u, v)) \right]$ γ-almost surely. Then, we deduce that:

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

Then we deduce the expected result:

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

Let us show that the optimum is attained. $\mathbb{Q} \mapsto \mathbb{E}_{(x,y)\sim\mathbb{Q}}\left[\mathbb{E}_{\theta\sim\mu}\left[L(\theta,(x,y))\right]\right]$ is upper semi continuous by Lemma 3 for the weak topology of measures, and $\mathcal{A}_{\varepsilon}(\mathbb{P})$ is compact by Proposition 3, 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 4 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 4, we also deduce that the adversarial risk can be casted as a linear optimization problem over distributions.

Remark 2. *In a recent work, [Pydi and Jog, 2020] 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 4 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.*

3.2 Nash Equilibria in the Adversarial Game

3.2.1 Adversarial Attacks as a Zero-Sum Game

Thanks to Proposition 3.1, the adversarial risk minimization problem can be seen as a two-player zero-sum game that writes as follows,

$$\inf_{\mu\in\mathcal{M}_+^1(\Theta)} \sup_{\mathbb{Q}\in\mathcal{A}_{\varepsilon}(\mathbb{P})} \mathbb{E}_{(x,y)\sim\mathbb{Q},\theta\sim\mu}\left[L(\theta,(x,y))\right]. \tag{3.3}$$

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 (3.3) simply amounts to solving the adversarial risk minimization problem – Problem (3.1), 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

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

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

Remark 3. *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 $T : \mathcal{X}\times\mathcal{Y} \to \mathcal{X}$ be defined such that for all $(x,y) \in \mathcal{X}\times\mathcal{Y}$, $\mathbb{E}_{\theta\sim\mu}\left[L(T(x,y),y)\right] = \sup_{x',\ d(x,x')\leq\varepsilon} \mathbb{E}_{\theta\sim\mu}\left[L(x',y)\right]$. Thanks to [Bertsekas and Shreve, 2004, Proposition 7.50], (T,id) 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.*

3.2.2 Dual Formulation of the Game

Every zero sum game has a dual formulation that allows a deeper understanding of the framework. Here, from Proposition 4, 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,

$$\sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})}\inf_{\mu\in\mathcal{M}_+^1(\Theta)}\mathbb{E}_{(x,y)\sim\mathbb{Q},\theta\sim\mu}\left[L(\theta,(x,y))\right]. \tag{3.4}$$

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 3.1.1, 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

$$\mathcal{D}^\varepsilon \leq \mathcal{V}^\varepsilon_{rand} \leq \mathcal{V}^\varepsilon_{det}. \tag{3.5}$$

Inequalities in Equation (3.5) 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 3.1.1, 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,

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

Furthermore, we can demonstrate that the dual problems for deterministic and randomized classifiers have the same value, hence the inequalities in Equation (3.5). Indeed in the absence of attackers, the deterministic and randomized optimal risks are the same.

Proposition 5. *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. □

We can deduce an immediate corollary 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.*

3.2.3 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

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

Alternatively, we can say that $(\mu^*, \mathbb{Q}^*)$ is a Nash equilibrium if and only if μ^* solves the adversarial risk minimization problem – Problem (3.1), $\mathbb{Q}^*$ the dual problem – Problem (3.6), 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 4. *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:*

$$\begin{aligned} &\inf_{\mu \in \mathcal{M}_1^+(\Theta)} \max_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{\theta\sim\mu, (x,y)\sim\mathbb{Q}} \left[L(\theta, (x, y)) \right] \\ = &\max_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \inf_{\mu \in \mathcal{M}_1^+(\Theta)} \mathbb{E}_{\theta\sim\mu, (x,y)\sim\mathbb{Q}} \left[L(\theta, (x, y)) \right] \end{aligned} \tag{3.6}$$

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.

To prove this result, let us first recall the Fan's Theorem.

Theorem 5. *Let U be a compact convex Haussdorff space and V be convex space (not necessarily topological). Let $\psi : U \times V \to \mathbb{R}$ be a concave-convex function such that for all $v \in V$, $\psi(\cdot, v)$ is upper semi-continuous then:*

$$\inf_{v \in V} \max_{u \in U} \psi(u, v) = \max_{u \in U} \inf_{v \in V} \psi(u, v)$$

We are now set to prove Theorem 4.

Proof. $\mathcal{A}_\varepsilon(\mathbb{P})$, endowed with the weak topology of measures, is a Hausdorff compact convex space, thanks to Proposition 3. 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 4 for convex cases. As mentioned, this result has limited impact in the adversarial classification setting. It is still a direct corollary of Fan's theorem. This theorem can be stated as follows:

Theorem 6. *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 4 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 4. *Theorem 4 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).*

3.3 Finding the Optimal Classifiers

3.3.1 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:

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

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

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

More details on this decomposition are given in Appendix 3.6.5. 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:

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

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

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

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

Proposition 6. *For $\mu \in \mathcal{M}_1^+(\Theta)$, one has*

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

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

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

[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

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. □

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

$$\inf_{\mu\in\mathcal{M}_1^+(\Theta)} \sum_{i=1}^N \frac{\alpha_i}{N}\log\left(\frac{1}{m_i}\sum_{j=1}^{m_i}\exp\frac{\mathbb{E}_\mu\left[L(\theta,u_j^{(i)})\right]}{\alpha_i}\right) \tag{3.7}$$

denoted $\widehat{\mathcal{R}}^{\varepsilon,\boldsymbol{m}}_{adv,\boldsymbol{\alpha}}$ 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

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

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 7. *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 $1-\delta$*

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

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 $\mu \in \mathcal{M}_1^+(\Theta)$,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\mathcal{F}\circ\boldsymbol{z} := \Big\{(f(z_1),\dots,f(z_q)),\ f\in\mathcal{F}\Big\} \quad , \quad \mathrm{Rad}(\mathcal{F}\circ\boldsymbol{z}) := \frac{1}{q}\mathbb{E}_{\boldsymbol{\sigma}\sim\{\pm1\}}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{q}\sigma_i f(z_i)\right]$$

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

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

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

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

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

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 $1-\delta$

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

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

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

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

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

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 7, let $(\mu_n^\varepsilon)_{n\geq 0}$ and $(\mu_n)_{n\geq 0}$ two sequences such that

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

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

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

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

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

□

Now that we have shown that solving (3.7) allows to obtain an approximation of the true solution $\widehat{\mathcal{R}}^{\varepsilon,*}_{adv}$, we next aim at deriving an algorithm to compute it.

3.3.2 Proposed Algorithms

From now on, we focus on finite class of classifiers. Let $\Theta = \{\theta_1, \ldots, \theta_l\}$, 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 1: Oracle-based Algorithm

$\boldsymbol{\lambda}_0 = \frac{\mathbf{1}_L}{L}; T;\ \eta = \frac{2}{M\sqrt{LT}}$
for $t = 1, \ldots, T$ **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 $k \in [L]$,
  $|\mathbb{E}_{\tilde{\mathbb{Q}}}(l(\theta_k, (x,y))) - \mathbb{E}_{\mathbb{Q}^*}(l(\theta_k, (x,y)))| \leq \delta$
  $\boldsymbol{g}_t = \left(\mathbb{E}_{\tilde{\mathbb{Q}}}(l(\theta_1, (x,y)), \ldots, \mathbb{E}_{\tilde{\mathbb{Q}}}(l(\theta_L, (x,y))\right)^T$
  $\boldsymbol{\lambda}_t = \Pi_{\Delta_L}\left(\boldsymbol{\lambda}_{t-1} - \eta \boldsymbol{g}_t\right)$
end

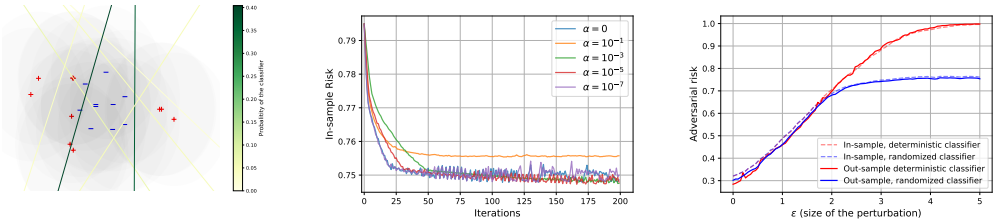


Figure 3.2: On left, 40 data samples with their set of possible attacks represented in shadow and the optimal randomized classifier, with a color gradient representing the probability of the classifier. In the middle, convergence of the oracle ($\alpha = 0$) and regularized algorithm for different values 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 1.

An Entropic Relaxation. Using the results from Section 3.3.1, 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: $f_{\theta_k}(x) = \langle \theta, x \rangle$ 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:

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

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 1. We get the following guarantee for this algorithm.

Proposition 9. *Let $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ satisfying Assumption 1. Then, Algorithm 1 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.

3.3.3 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., 2016, 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 2) 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 3.3.2. More details on this algorithm are available in Appendix 3.6.

Algorithm 2: 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

3.4 Experiments

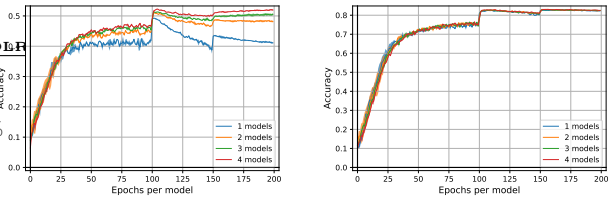
3.4.1 Synthetic Dataset

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 3.2 (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 3.2 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.

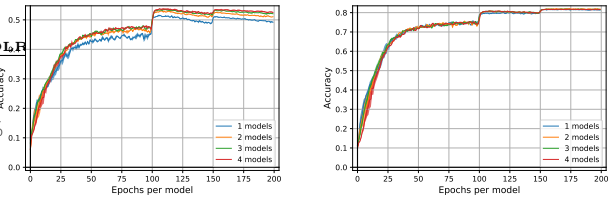
Finally, to demonstrate the improvement randomized techniques offer against deterministic defenses, we plot in Figure 3.2 (right) the minimum adversarial risk for both randomized and deterministic classifiers w.r.t. ε. The adversarial risk is strictly better for randomized classifier

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



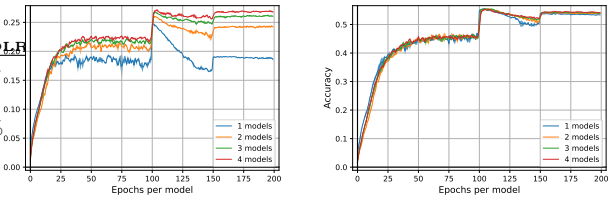
Adversarial Training, CIFAR-10 dataset results

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



TRADES, CIFAR-10 dataset results

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



Adversarial Training, CIFAR-100 dataset results

Figure 3.3: 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 $\varepsilon = 8/255$.

whenever the adversarial budget ε is bigger than 2. This illustration validates our analysis of Theorem 4, and motivates a in depth study of a more challenging framework, namely image classification with neural networks.

3.4.2 CIFAR Datasets

Experimental Setup. We now implement our heuristic algorithm (Alg. 2) on CIFAR-10 and CIFAR-100 datasets for both Adversarial Traning [Madry et al., 2018] and TRADES [Zhang et al., 2019] loss. To evaluate the performance of Algorithm 2, we trained from 1 to 4 ResNet18 [He et al., 2016a] 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 2 exactly corresponds to adversarial training [Madry

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

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. We give more details on implementation in Appendix 3.6.

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 3.3. 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.

3.5 Additional Lemmas and Results for Chapter 3

3.6 Additional Experimental Results

3.6.1 Experimental setting.

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 2 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 $T_\lambda = 25$.

3.6.2 Effect of the Regularization

In this subsection, we experimentally investigate the effect of the regularization. In Figure 3.4, we notice, that the regularization has the effect of stabilizing, reducing the variance and improving the level of the robust accuracy for adversarial training for mixtures (Algorithm 2). The standard accuracy curves are very similar in both cases.

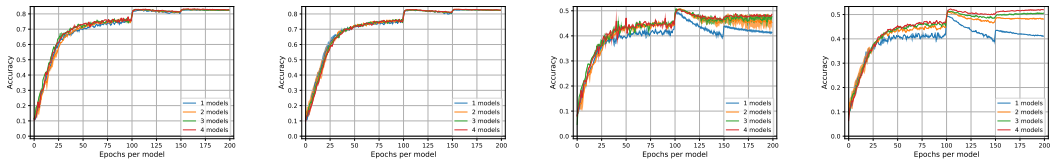


Figure 3.4: On left and middle-left: Standard accuracies over epochs with respectively no regularization and regularization set to $\alpha = 0.001$. On middle right and right: Robust accuracies for the same parameters against PGD attack with 20 iterations and $\varepsilon = 0.03$.

3.6.3 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 3.5. We remark this architecture can lead to more robust models, corroborating the results from Gowal et al. [2020].

Models	Acc.	APGD_{CE}	APGD_{DLR}
1	85.2%	49.9%	50.2%
2	**86.0%**	**51.5%**	**52.1%**

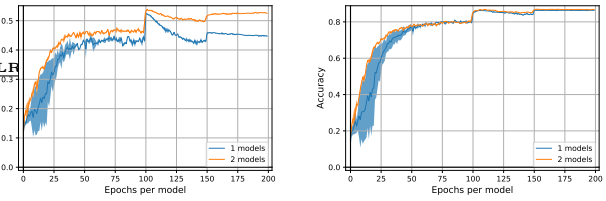


Figure 3.5: On left: Comparison of our algorithm with a standard adversarial training (one model) on WideResNet28x10. We reported the results for the model with the best robust accuracy obtained over two independent runs because adversarial training might be unstable. Standard and Robust accuracy (respectively 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 $\varepsilon = 8/255$.

3.6.4 Overfitting in Adversarial Robustness

We further investigate the overfitting of our heuristic algorithm. We plotted in Figure 3.6 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.

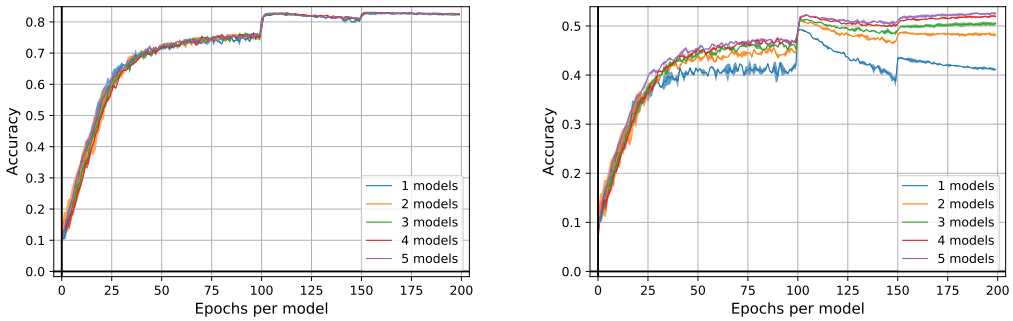


Figure 3.6: Standard and Robust accuracy (respectively on left and on right) on CIFAR-10 test images in function of the number of epochs per classifier with 1 to 5 ResNet18 models. The performed attack is PGD with 20 iterations and $\varepsilon = 8/255$. The best mixture for 5 models occurs at the end of training (epoch 198).

3.6.5 Additional Results

3.6.6 Decomposition of the Empirical Risk for Entropic Regularization

Proposition 10. *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:*

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

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

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

Proof. This proposition is a direct application of Proposition 4 for diracs $\delta_{(x_i,y_i)}$. □

3.6.7 Case of Separated Conditional Distribtions

Proposition 11. *Let $\mathcal{Y} = \{-1,+1\}$. Let $\mathbb{P} \in \mathcal{M}_+^1(\mathcal{X}\times\mathcal{Y})$. Let $\varepsilon > 0$. For $i \in \mathcal{Y}$, let us denote $\mathbb{P}_i$ the distribution of $\mathbb{P}$ conditionally to $y = i$. Let us assume that $d_{\mathcal{X}}(\mathrm{supp}(\mathbb{P}_{1+1}), \mathrm{supp}(\mathbb{P}_{-1})) > 2\varepsilon$. Let us consider the nearest neighbor deterministic classifier : $f(x) = d(x,\mathrm{supp}(\mathbb{P}_{+1})) - d(x,\mathrm{supp}(\mathbb{P}_{-1}))$ and the 0/1 loss $l(f,(x,y)) = \mathbf{1}_{yf(x)\leq 0}$. Then f satisfies both optimal standard and adversarial risks: $\mathcal{R}(f) = 0$ and $\mathcal{R}_\epsilon^\varepsilon(f) = 0$.*

Proof. Let Let denote $p_i = \mathbb{P}(y = i)$. Then we have

$$\mathcal{R}_\epsilon^\varepsilon(f) = p_{+1}\mathbb{E}_{\mathbb{P}_{+1}}\left[\sup_{x',\ d(x,x')\leq\varepsilon} \mathbf{1}_{f(x')\leq 0}\right] + p_{-1}\mathbb{E}_{\mathbb{P}_{-1}}\left[\sup_{x',\ d(x,x')\leq\varepsilon} \mathbf{1}_{f(x')\geq 0}\right]$$

For $x \in \text{supp}(\mathbb{P}_{+1})$, we have, for all x' such that $d(x, x') \neq 0$, $f(x') > 0$, then: $\mathbb{E}_{\mathbb{P}_{+1}}\left[\sup_{x',\ d(x,x')\leq\varepsilon} \mathbf{1}_{f(x')\leq 0}\right] = 0$. Similarly, we have $\mathbb{E}_{\mathbb{P}_{-1}}\left[\sup_{x',\ d(x,x')\leq\varepsilon} \mathbf{1}_{f(x')\geq 0}\right] = 0$. We then deduce the result. □

Chapter 4

Consistency Study of Adversarial loss

Contents

Disclaimer: This section is still an unachieved work

In this section we assume a binary classification task, i.e. an output space $\mathcal{Y} = \{-1, +1\}$. W extend the notion of loss functions to general cost functions. A cost 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 universally measurable for all $f \in \mathcal{F}(\mathcal{X})$. We will denote $\mathcal{R}_L$ the risk associated with a loss L.

4.1 Loss Consistency in Classification

4.1.1 Consistency in Standard Classification

In a standard classification setting, given (x, y), we recall the classification loss is defined as:

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

In the literature the notion of consistency with regards to the 0/1 loss is defined as follows.

Definition 8 (Classification Consistency)**.** *A cost function L is said to be consistent with respect to 0/1 loss for a a probability distribution $\mathbb{P} \in \mathcal{M}_1^+(\mathcal{X} \times \mathcal{Y})$ if and only if for all sequences $(f_n)_n$ of measurable functions:*

$$\mathcal{R}_L(f_n) \to \mathcal{R}_L^\star \implies \mathcal{R}_{0/1}(f_n) \to \mathcal{R}_{0/1}^\star \tag{4.1}$$

Standard results in classification. Previous literature have focused, in standard classification, on margin losses, and it is defined as follows.

Definition 9 (Margin loss)**.** *We say a loss L is a margin loss if there exists a measurable function $\phi : \mathbb{R} \to \mathbb{R}_+$, such that for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $f : \mathcal{X} \to \mathbb{R}$ a measurable function,*

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

Without loss of generality, we will denote $\mathcal{R}_\phi$ the risk associated with a margin loss. Standard classification consistency highly relies on the fact that given a loss, the minimization can be made pointwisely, independently from the considered distribution $\mathbb{P}$. The notion of consistency exactly matches with the notion of calibration, defined as follows:

Definition 10 (Classification Calibration)**.** *A margin loss ϕ is said to be classification-calibrated if for all $\eta \in [0, 1]$, $\eta \neq \frac{1}{2}$:*

$$H(\eta) > H^-(\eta)$$

where $H(\eta) := \inf_{\alpha \in \mathbb{R}} C_\eta(\alpha)$, $H^-(\eta) = \inf_{\alpha,\ \alpha(2\eta-1)\leq 0} C_\eta(\alpha)$ and $C_\eta(\alpha) := \eta\phi(\alpha)+(1-\eta)\phi(-\alpha)$.

Bartlett et al. [2006] and Steinwart [2007] show that consistency and calibration are equivalent notions in standard classification.

Theorem 7. *In standard calibration, a margin loss ϕ is consistent with regards to classification loss if and only if ϕ is classification calibrated.*

In particular, one can state the following results on convex margin losses.

Theorem 8. *Let ϕ be a convex margin differentiable in 0. ϕ is consistent with regards to classification loss if and only if $\phi'(0) < 0$.*

What is the good 0/1 loss? In this paragraph, we precise why this 0/1 loss is used in consistency/calibration study. Let $f : \mathcal{X} \to \mathbb{R}$ be a measurable function. In the literature, we generally find three ways for defining the 0/1 loss:

- $L_\leq(x, y, f) = \mathbf{1}_{yf(x)\leq 0}$, the loss that penalizes indecision.
- $L_<(x, y, f) = \mathbf{1}_{yf(x)<0}$, the loss that encourages indecision.
- $L_{0/1}(x, y, f) = \mathbf{1}_{y\text{sign}(f(x))\leq 0}$, with a sign convention ($sign(0) = 1$ for instance).

The original results from [Bartlett et al., 2006, Steinwart, 2007] tackles the problem of the consistency with respect to the latter one. We can even remark that usual consistency results stated in the two previous papers are false for the two first losses, as shown in the following simple counterexample.

Counterexample. *Let $\mathbb{P}$ defined as follows: $\mathbb{P}(Y = -1) = \mathbb{P}(Y = 1) = \frac{1}{2}$, and $\mathbb{P}(X = 0 \mid Y) = 1$. Let $\phi : \mathbb{R} \to \mathbb{R}$ be a margin loss. The ϕ-risk minimization problem writes $\inf_\alpha \frac{1}{2}(\phi(\alpha)+\phi(-\alpha))$. For a convex margin loss ϕ the optimum is attained in $\alpha = 0$.*

1. *$f_n : x \mapsto 0$ 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 can be calibrated for $L_\leq$.*
2. *$f_n : x \mapsto \frac{1}{n}$ is a minimizing sequence for the ϕ-risk. However $R_{L_<}(f_n) = 1/2$ for all n and $R^*_{L_<} = 0$. Then we deduce that no convex margin based loss can be calibrated for $L_<$.*

4.1.2 Consistency in adversarial setting

Consistency in standard classification is studied with respect to the cost $L_{0/1}(x, y, f) = \mathbf{1}_{y\text{sign}(f(x))\leq 0}$. In the adversarial setting, the natural loss to consider then is

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

with a sign convention (e.g. $sign(0) = 1$). Otherwise, when $\varepsilon = 0$, the consistency results would be wrong as stated in the previous counterexample. Previous works [Awasthi et al., 2021a,b, Bao et al., 2020] focused on the loss $\sup_{x' \in B_\varepsilon(x)} \mathbf{1}_{yf(x')\leq 0}$ that consequently lead to misleading results because not compatible with standard classification when $\varepsilon = 0$. We define the notion of consistency with regards to adversarial 0/1 loss as follows.

Definition 11 (Adversarial Consistency)**.** *A cost function L is said to be consistent with respect to 0/1 loss for a distribution $\mathbb{P} \in \mathcal{M}_1^+(\mathcal{X} \times \mathcal{Y})$ if and only if for all sequences $(f_n)_n$ of measurable functions:*

$$\mathcal{R}_L(f_n) \to \mathcal{R}_L^\star \implies \mathcal{R}_{0/1}^\varepsilon(f_n) \to \mathcal{R}_{0/1}^{\varepsilon,\star} \tag{4.2}$$

Calibration in adversarial classification One need to note that the notion of calibration in the adversarial setting is misleading.

TODO.

Hence, one need to focus specifically on the notion of consistency in the adversarial setting. In the next section, we prove some results on adversarial consistency.

4.2 Adversarial Consistency Results

4.2.1 Convex Losses are not Consistent

Following the work on standard classification, natural candidate losses for adversarial consistency would be suprema of standardly calibrared margin losses:

$$L(x, y, f) = \sup_{x', d(x,x')\leq\varepsilon} \phi(yf(x))$$

including a wide range of convex functions ϕ. Next theorem shows the counter-intuitive result that no calibrated convex loss can define a consistent loss for adversarial classification

Proposition 12. *Let ϕ be a convex classification-calibrated margin loss (i.e. $\phi'(0) < 0$). In $\mathbb{R}$, for any $\varepsilon > 0$, there exists a distribution $\mathbb{P}$ such that $x \to \sup_{x', d(x,x')\leq\varepsilon} \phi(yf(x))$ is not consistent for $\mathbb{P}$ with regards to $L_{0/1,\varepsilon}$.*

Proof. Let $\epsilon > 0$. We will construct a distribution $\mathcal{D}$ and a sequence of classifier h_n so that the ϕ-risk converges toward the optimal, while the $0-1$ loss remains constant at a non-optimal value.

Let a such that $\epsilon < a < 2\epsilon$. We define D by :

$$\begin{cases} \mathbb{P}(Y = 1, X = 0) = \frac{1}{2} \\ \mathbb{P}(Y = -1, X = -a) = \frac{1}{4} \\ \mathbb{P}(Y = -1, X = a) = \frac{1}{4} \\ \mathbb{P}(X, Y) = 0 \qquad\qquad \text{otherwise.} \end{cases}$$

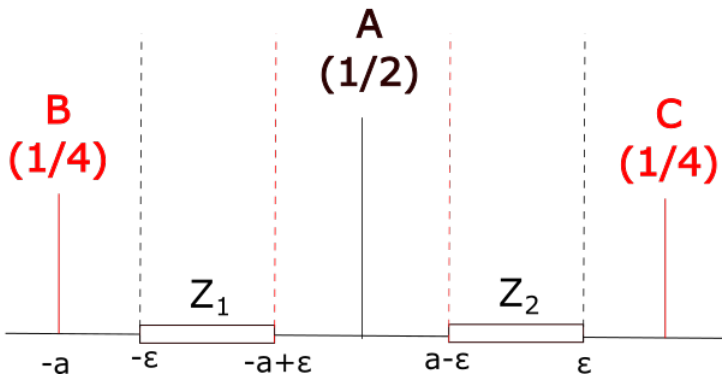


Figure 4.1: The distribution we use as a counter-example. For the $0-1$ loss, there are two optimal classifiers, putting either both Z_1 and Z_2 to 1 (and saving point A), or both to -1 (saving points B and C). With convex surrogate losses however, the optimal is to bring values in both Z_1 and Z_2 to zero, which can be done while maintaining opposite signs on Z_1 and Z_2, and so ensuring the loss of both A and one of B and C for the $0-1$ loss.

Since ϕ is classification-calibrated, it is either non-increasing, or it has a minimum, attained on a point $u > 0$. Let us consider both cases :

First case : ϕ non-increasing We define $Z_1 = [-\epsilon, -a+\epsilon]$ and $Z_2 = [a-\epsilon, \epsilon]$, the zones that can be attacked by two points at once. Let us then define :

$$\forall x \in \mathbb{R}, h_n(x) = \begin{cases} \frac{1}{n} & \text{if } x \in Z_1 \\ -1/n & \text{if } x \in Z_2 \\ 1 & \text{if } x \in [-a+\epsilon, a-\epsilon] \\ -1 & \text{otherwise} \end{cases}$$

Since the central point and the point $x = -a$ are attackable, but not the point $x = a$, we have $\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sup_{z\in\mathcal{B}(x,\epsilon)} l_{0/1}(y, h_n(z))\right] = \frac{3}{4}$, which is worse than the constant classifier h=1, which gives a score of $\frac{1}{2}$. So h_n is not a minimizing sequence for the $0-1$ loss.

Let us now show that h_n is a minimizing sequence for the loss ϕ under attack, which will be proof of the non-consistency.

$$\begin{aligned} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sup_{z\in\mathcal{B}(x,\epsilon)} \phi(y * h_n(z))\right] &= \frac{1}{2}\phi(\frac{-1}{n}) + \frac{1}{4}\phi(\frac{1}{n}) + \frac{1}{4}\phi(\frac{1}{n}) \\ &= \frac{1}{2}\phi(\frac{-1}{n}) + \frac{1}{2}\phi(\frac{1}{n}) \\ &\underset{n\to+\infty}{\longrightarrow} \phi(0) \end{aligned}$$

We now need to show that $\phi(0)$ is a lower bound of the optimal adversarial risk for the loss

ϕ. Let h be any classifier. We define :

$$\begin{aligned}
b &= \inf_{z\in[-a+\epsilon,a-\epsilon]} h(z) \\
c &= \sup_{z\in[-a-\epsilon,-\epsilon]} h(z) \\
d &= \sup_{z\in[\epsilon,a+\epsilon]} h(z) \\
m_i &= \inf_{z\in\mathcal{Z}_i} h(z) \text{ for } i \in \{1,2\} \\
M_i &= \sup_{z\in\mathcal{Z}_i} h(z) \text{ for } i \in \{1,2\} \\
m &= \min(m_1, m_2)
\end{aligned}$$

We then have :

$$\begin{aligned}
\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sup_{z\in\mathcal{B}(x,\epsilon)} \phi(y * h_n(z))\right] &= \frac{1}{2}\sup_{z\in[-\epsilon,\epsilon]}\phi(h(z)) + \frac{1}{4}\sup_{z\in[-a-\epsilon,-a+\epsilon]}\phi(-h(z)) + \frac{1}{4}\sup_{z\in[a-\epsilon,a+\epsilon]}\phi(-h(z)) \\
&= \frac{1}{2}\max\left[\sup_{z\in[-a+\epsilon,a-\epsilon]}\phi(h(z)), \sup_{z\in\mathcal{Z}_1}\phi(h(z)), \sup_{z\in\mathcal{Z}_2}\phi(h(z))\right] \\
&+ \frac{1}{4}\max\left[\sup_{z\in[-a-\epsilon,-\epsilon]}\phi(-h(z)), \sup_{z\in\mathcal{Z}_1}\phi(-h(z))\right] \\
&+ \frac{1}{4}\max\left[\sup_{z\in[\epsilon,a+\epsilon]}\phi(-h(z)), \sup_{z\in\mathcal{Z}_2}\phi(-h(z))\right] \\
&= \frac{1}{2}\max\left[\phi(b), \phi(m_1), \phi(m_2)\right] + \frac{1}{4}\max\left[\phi(-M_1), \phi(-c)\right] \\
&+ \frac{1}{4}\max\left[\phi(-M_2), \phi(-c)\right] \\
&\geq \frac{1}{2}\max\left[\phi(m_1), \phi(m_2)\right] + \frac{1}{4}\phi(-M_1) + \frac{1}{4}\phi(-M_2) \\
&\geq \frac{1}{2}\phi(m) + \frac{1}{4}\phi(-m) + \frac{1}{4}\phi(-m) \\
&\geq \phi(0)
\end{aligned}$$

Hence the result. □

4.2.2 Realisable case

In this realisable case, i.e. $\mathcal{R}^{\varepsilon}_{0/1} = 0$, results about standard classification consistency still hold.

Proposition 13. *Let $\mathbb{P}$ be a Borel probability distribution over $\mathcal{X}\times\mathcal{Y}$ such that $\mathcal{R}^{\varepsilon}_{0/1} = 0$. Then, let ϕ be a standard classification calibrated margin-based loss. Then $\tilde{\phi} : (x, y, f) \mapsto \sup_{x'\in B_\varepsilon(x)}\phi(yf(x'))$ is adversarially consistent at level ε.*

Proof. Coming soon □

Chapter 5

Certification Methods for Adversarial Examples

Contents

In this chapter we focus on noise injection and convex flows to build certifiable defenses on adversarial robustness. A desirable property for a function to be robust against adversarial examples is that the classifier $\mathbf{f} = (f_1, \dots, f_K)$ is a M-Lipschitz function with regards to Euclidean norm i.e. $\|\mathbf{f}(x) - \mathbf{f}(x')\|_2 \leq M\|x - x'\|$ for all $x, x' \in \mathcal{X}$.

Proposition 14 ([Li et al., 2019b])**.** *Let $\mathbf{f}$ be a M-Lipschitz classifier. Then for $\varepsilon > 0$ and for every x with label y such that*

$$\max(0, f_y(x) - \max_{y' \neq y} f_{y'}(x)) > \sqrt{2}M\varepsilon$$

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

$$\operatorname*{argmax}_i f_i(x + \tau) = y$$

This property ensures robustness guarantees for Lipschitz classifiers

5.1 Certification using Noise Injection

As we will inject some noise in our classifier in order to defend against adversarial attacks, we need to introduce the notion of "probabilistic mapping".

Definition 12 (Probabilistic mapping)**.** *Let $\mathcal{X}$ be an arbitrary space, and $(\mathcal{Y}, \mathcal{F}_\mathcal{Y})$ a measurable space. A* probabilistic mapping *from $\mathcal{X}$ to $\mathcal{Y}$ is a randomized classifier, i.e. a mapping $h : \mathcal{X} \to \mathcal{M}_1^+(\mathcal{Y})$. To obtain a numerical output out of this* probabilistic mapping*, one needs to sample y according to $h(x)$.*

Any mapping can be considered as a probabilistic mapping, whether it explicitly injects noise (as in [Dhillon et al., 2018, Lecuyer et al., 2018, Rakin et al., 2018]) or not. In fact, any deterministic mapping can be considered as a probabilistic mapping, since it can be characterized by a Dirac measure. Accordingly, the definition of a probabilistic mapping is fully general and equally treats networks with or without noise injection. There exists no definition of robustness against adversarial attacks that comply with the notion of probabilistic mappings. We settle that by generalizing the notion of prediction-change risk initially introduced in [Diochnos et al., 2018] for deterministic classifiers. Let h be a probabilistic mapping from $\mathcal{X}$ to $\mathcal{Y}$, and $d_{\mathcal{M}_1^+(\mathcal{Y})}$ some metric/divergence on $\mathcal{M}_1^+(\mathcal{Y})$. We define the (ε, α)-radius prediction-change risk of h w.r.t. $\mathcal{D}_\mathcal{X}$ and $d_{\mathcal{M}_1^+(\mathcal{Y})}$ as

$$\text{PC-Risk}_\alpha(h, \alpha) := \mathbb{P}_{x \sim \mathcal{D}_\mathcal{X}} \left[\exists \tau \in B(\varepsilon) \text{ s.t. } d_{\mathcal{M}_1^+(\mathcal{Y})}(h(x + \tau), h(x)) > \alpha \right] .$$

These three generalized notions allow us to analyze noise injection defense mechanisms (Theorems 9, and 10). We can also define adversarial robustness (and later adversarial gap) thanks to these notions.

Definition 13 (Adversarial robustness)**.** *Let $d_{\mathcal{M}_1^+(\mathcal{Y})}$ be a metric/divergence on $\mathcal{M}_1^+(\mathcal{Y})$. A probabilistic mapping h is called $d_{\mathcal{M}_1^+(\mathcal{Y})}$-$(\varepsilon, \alpha, \gamma)$ robust if PC-Risk$_\alpha(h, \alpha) \leq \gamma$, $d_{\mathcal{M}_1^+(\mathcal{Y})}$-$(\varepsilon, \alpha)$ robust if $\gamma = 0$.*

It is difficult in general to show that a classifier is $d_{\mathcal{M}_1^+(\mathcal{Y})}$-$(\varepsilon, \alpha, \gamma)$ robust. However, we can derive some bounds for particular divergences that will ensure robustness up to a certain level (Theorem 9). It is worth noting that our definition of robustness depends on the considered metric/divergence between probability measures. Lemma 5 gives some insights on the monotony of the robustness according to the parameters, and the probability metric/divergence at hand.

Lemma 5. *Let h be a probabilistic mapping, and let d_1 and d_2 be two metrics on $\mathcal{M}_1^+(\mathcal{Y})$. If there exists a non decreasing function $\phi : \mathbb{R} \to \mathbb{R}$ such that $\forall \mu_1, \mu_2 \in \mathcal{M}_1^+(\mathcal{Y})$, $d_1(\mu_1, \mu_2) \leq \phi(d_2(\mu_1, \mu_2))$, then the following assertion holds: h is d_2-$(\varepsilon, \alpha, \gamma)$-robust $\implies$ h is d_1-$(\varepsilon, \phi(\alpha), \gamma)$-robust.*

As suggested in Definition 13 and Lemma 5, any given choice of metric/divergence will instantiate a particular notion of adversarial robustness and it should be carefully selected.

Proposition 15 (Renyi implies TV-robustness)**.** *Let h be a probabilistic mapping, then for all $\lambda \geq 1$, $\alpha > 0$, there exists $\alpha' > 0$ s.t. if h is $d_{R,\lambda}$-$(\varepsilon, \alpha, \gamma)$-robust then h is d_{TV}-$(\varepsilon, \alpha', \gamma)$-robust.*

A crucial property of Renyi-robustness is the *Data processing inequality*. It is a well-known inequality from information theory which states that *"post-processing cannot increase information"* [Beaudry and Renner, 2012, Cover and Thomas, 2012]. In our case, if we consider a Renyi-robust probabilistic mapping, composing it with a deterministic mapping maintains Renyi-robustness with the same level.

Proposition 16 (Data processing inequality)**.** *Let us consider a probabilistic mapping $h : \mathcal{X} \to \mathcal{M}_1^+(\mathcal{Y})$, and denote $\rho : \mathcal{Y} \to \mathcal{Y}'$ a deterministic function. If $U \sim h(x)$ then the probability measure $M'(x)$ s.t. $\rho(U) \sim M'(x)$ defines a probabilistic mapping $M' : \mathcal{X} \to \mathcal{M}_1^+(\mathcal{Y}')$. For any $\lambda > 1$, if h is $d_{R,\lambda}$-$(\varepsilon, \alpha, \gamma)$ robust then M' is also $d_{R,\lambda}$-$(\varepsilon, \alpha, \gamma)$ robust.*

Data processing inequality will allow us later to inject some additive noise in any layer of a neural network and to ensure Renyi-robustness.

5.1.1 Defense mechanisms based on Exponential family noise injection

Robustness through Exponential family noise injection

For now, the question of which class of noise to add is treated *ad hoc*. We choose here to investigate one particular class of noise closely linked to the Renyi divergence, namely Exponential family distributions, and demonstrate their interest. Let us first recall what the Exponential family is.

Definition 14 (Exponential family)**.** *Let Θ be an open convex set of $\mathbb{R}^n$, and $\theta \in \Theta$. Let ν be a measure dominated by μ (either by the Lebesgue or counting measure), it is said to be part of the* Exponential family *of parameter θ (denoted $E_F(\theta, t, k)$) if it has the following p.d.f.*

$$p_F(z, \theta) = \exp\left\{\langle t(z), \theta\rangle - u(\theta) + k(z)\right\}$$

where $t(z)$ is a sufficient statistic, k a carrier measure (either for a Lebesgue or a counting measure) and $u(\theta) = \log \int_z \exp\left\{< t(z), \theta > + k(z)\right\} dz$.

To show the robustness of randomized networks with noise injected from the Exponential family, one needs to define the notion of sensitivity for a given deterministic function:

Definition 15 (Sensitivity of a function)**.** *For any $\varepsilon \geq 0$ and for any $||.||_A$ and $||.||_B$ two norms, the ε-sensitivity of f w.r.t. $||.||_A$ and $||.||_B$ is defined as*

$$\Delta_{\varepsilon}^{A,B}(f) := \sup_{x,y \in \mathcal{X}, ||x-y||_A \leq \varepsilon} ||f(x) - f(y)||_B \ .$$

Let us consider an n-layer feedforward neural network $\mathcal{N}(.) = \phi^n \circ ... \circ \phi^1(.)$. For any $i \in [n]$, we define $\mathcal{N}_{|i}(.) = \phi^i \circ ... \circ \phi^1(.)$ the neural network truncated at layer i. Theorem 9 shows that, injecting noise drawn from an Exponential family distribution ensures robustness to adversarial example attacks in the sense of Definition 13.

Theorem 9 (Exponential family ensures robustness)**.** *Let us denote $\mathcal{N}_X^i(.) = \phi^n \circ ... \circ \phi^{i+1}(\mathcal{N}_{|i}(.) + X)$ with X a random variable. Let us also consider two arbitrary norms $||.||_A$ and $||.||_B$ respectively on $\mathcal{X}$ and on the output space of $\mathcal{N}_X^i$.*

- *If $X \sim E_F(\theta, t, k)$ where t and k have non-decreasing modulus of continuity ω_t and ω_k. Then for any $\varepsilon \geq 0$, $\mathcal{N}_X^i(.)$ defines a probabilistic mapping that is $d_{R,\lambda}$-(ε, α) robust with $\alpha = ||\theta||_2 \omega_t^{B,2}(\Delta_{\varepsilon}^{A,B}(\phi)) + \omega_k^{B,1}(\Delta_{\varepsilon}^{A,B}(\phi))$ where $||.||_2$ is the norm corresponding to the scalar product in the definition of the exponential family density function and $||.||_1$ is the absolute value on $\mathbb{R}$. Notions of continuity modulus is defined in the supplementary material.*
- *If X is a centered Gaussian random variable with a non degenerated matrix parameter Σ. Then for any $\varepsilon \geq 0$, $\mathcal{N}_X^i(.)$ defines a probabilistic mapping that is $d_{R,\lambda}$-(ε, α) robust with $\alpha = \frac{\lambda \Delta_{\varepsilon}^{A,2}(\phi)^2}{2\sigma_{min}(\Sigma)}$ where $||.||_2$ is the canonical Euclidean norm on $\mathbb{R}^n$.*

In simpler words, the previous theorem ensures stability in the neural network when injecting noise w.r.t. the distribution of the output. Intuitively, if two inputs are close w.r.t. $\|\cdot\|_A$, the output distributions of the network will be close in the sense of Renyi divergence. It is well known that in the case of deterministic neural networks, the Lipschitz constant becomes bigger as the number of layers increases [Gouk et al., 2018]. By injecting noise at layer i, the notion of robustness only depends on the sensitivity of the first i layers of the network and not the following ones. In that sense, randomization provides a more precise control on the "continuity" of the neural network. In the next section, we show that thanks to the notion of robustness w.r.t. probabilistic mappings, one can bound the loss of accuracy of a randomized neural network when it is attacked.

Bound on the risk gap under attack and certified accuracy

The notions of risk and adversarial risk can easily be generalized to encompass probabilistic mappings.

Definition 16 (Risks for probabilistic mappings)**.** *Let h be a probabilistic mapping from $\mathcal{X}$ to $\mathcal{Y}$, the risk and the ε-radius adversarial risk of h w.r.t. $\mathcal{D}$ are defined as:*

$$\mathcal{R}(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{E}_{y'\sim h(x)}\left[\mathbb{1}\left(y' \neq y\right)\right]\right]$$
$$\mathcal{R}_{\epsilon}(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sup_{\|\tau\|_{\mathcal{X}} \leq \varepsilon} \mathbb{E}_{y'\sim h(x+\tau)}\left[\mathbb{1}\left(y' \neq y\right)\right]\right] \ .$$

The definition of adversarial risk for a probabilistic mapping can be matched with the concept of Expectation over Transformation (EoT) attacks [Athalye et al., 2018a]. Indeed, EoT

attacks aim at computing the best opponent in expectation for a given random transformation. In the adversarial risk definition, the adversary chooses the perturbation which has the greatest probability to fool the model, which is a stronger objective than the EoT objective. Theorem 10 provides a bound on the gap between the adversarial risk and the regular risk:

Theorem 10 (Adversarial risk gap bound in the randomized setting)**.** *Let h be the probabilistic mapping at hand. Let us suppose that h is $d_{R,\lambda}$-(ε, α) robust for some $\lambda \geq 1$ then:*

$$|\mathcal{R}_\epsilon(h) - \mathcal{R}(h)| \leq 1 - e^{-\alpha}\mathbb{E}_x\left[e^{-H(h(x))}\right]$$

where H is the Shannon entropy $H(p) = -\sum_i p_i \log(p_i)$.

This theorem gives a control on the loss of accuracy under attack w.r.t. the robustness parameter α and the entropy of the predictor. It provides a tradeoff between the quantity of noise added in the network and the accuracy under attack. Intuitively, when the noise increases, for any input, the output distribution tends towards the uniform distribution, then, $\alpha \to 0$ and $H(h(x)) \to \log(K)$, and the risk and the adversarial risk both tends to $\frac{1}{K}$ where K is the number of classes in the classification problem. On the opposite, if no noise is injected, for any input, the output distribution is a Dirac distribution, then, if the prediction for the adversarial example is not the same as for the regular one, $\alpha \to \infty$ and $H(h(x)) \to 0$. Hence, the noise needs to be designed both to preserve accuracy and robustness to adversarial attacks. In the Section 5.1.2, we give an illustration of this bound when h is a neural network with noise injection at input level as presented in Theorem 9. In practice, we do not have access to the real value of the entropy, but we estimate it with classical estimators [Paninski, 2003].

Our framework being general enough it encompasses several known accuracy certificates from the literature, e.g. the one provided in [Lecuyer et al., 2018]. Interestingly, we can introduce the following one, based on *our* definition of robustness.

Theorem 11. *Let $x \in \mathcal{X}$, and h be a probabilistic mapping with values in $\mathbb{R}^K$. If h is $d_{R,\lambda}$-(ε, α) robust, and if there exist k^* and $\delta^* \in (0,1)$ s.t. $\mathbb{E}_{y\sim h(x)}[y_{k^*}] > e^{2\alpha'} \max_{i \neq k^*} \mathbb{E}_{y\sim h(x)}[y_i] + (1+e^{\alpha'})\delta^*$, with $\alpha' = \alpha + \frac{\log(1/\delta^*)}{\lambda - 1}$. Then, for the classifier $f : x \mapsto \operatorname{argmax}_{k\in[K]} \mathbb{E}_{y\sim h(x)}[y_k]$ there is no perturbation $\tau \in B(0, \varepsilon)$ such that $f(x) \neq f(x+\tau)$.*

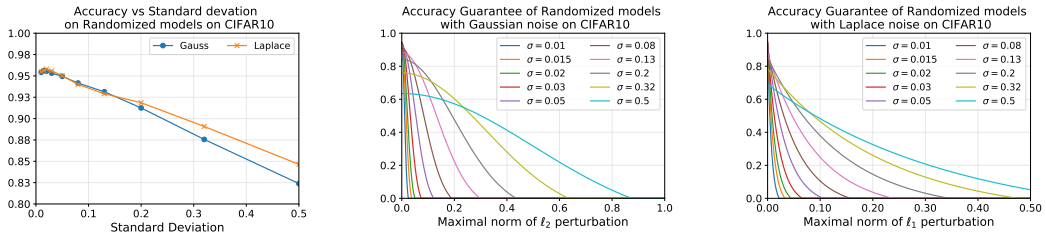
As the main focus of this work is to give theoretical evidence for randomization techniques, numerical experiments will mainly focus on Theorem 9 and 10 and not on certificates (Theorem 11).

5.1.2 Numerical experiments

To illustrate our theoretical findings, we train randomized neural networks with a simple method which consists in injecting a noise drawn from an Exponential family distribution in the image during training and inference.

Experimental setup

We present our results and analysis on CIFAR-10, CIFAR-100 [Krizhevsky et al., 2009] and ImageNet datasets [Deng et al., 2009]. For CIFAR-10 and CIFAR-100 [Krizhevsky et al., 2009], we used a Wide ResNet architecture [Zagoruyko and Komodakis, 2016] which is a variant of the ResNet model from [He et al., 2016a]. We use 28 layers with a widen factor of 10. We train all networks for 200 epochs, a batch size of 400, dropout 0.3 and Leaky Relu activation with



(a) (b) (c)

Figure 5.1: (a) Impact of the standard deviation of the injected noise on accuracy in a randomized model on CIFAR-10 with a Wide ResNet architecture. (b) and (c) illustration of the guaranteed accuracy of different randomized models with Gaussian (b) and Laplace (c) noises given the norm of the adversarial perturbation. The accuracies and entropies and estimated empirically.

a slope on $\mathbb{R}^-$ of 0.1. We minimize the Cross Entropy Loss with Momentum 0.9 and use a piecewise constant learning rate of 0.1, 0.02, 0.004 and 0.00008 after respectively 7500, 15000 and 20000 steps. The networks achieve for CIFAR10 and 100 a TOP-1 accuracy of 95.8% and 79.1% respectively on test images. For ImageNet [Deng et al., 2009], we use an Inception ResNet v2 [Szegedy et al., 2017] which is the sate of the art architecture for this dataset and achieve a TOP-1 accuracy of 80%. For the training of ImageNet, we use the same hyper parameters setting as the original implementation. We train the network for 120 epochs with a batch size of 256, dropout 0.8 and Relu as activation function. All evaluations were done with a single crop on the non-blacklisted subset of the validation set.

To transform these classical networks to probabilistic mappings, we inject noise drawn from Laplace and Gaussian distributions, each with various standard deviations. While the noise could theoretically be injected anywhere in the network, we inject the noise on the image for simplicity. More experiments with noise injected in the first layer of the network are presented in the supplementary material. To evaluate our models under attack, we use three powerful iterative attacks with different norms: *ElasticNet* attack (EAD) [Chen et al., 2018a] with ℓ_1 distortion, *Carlini&Wagner* attack (C&W) [Carlini and Wagner, 2017] with ℓ_2 distortion and *Projected Gradient Descent* attack (PGD) [Madry et al., 2018] with ℓ_∞ distortion. All standard deviations and attack intensities are in between -1 and 1. Precise descriptions of our numerical experiments and of the attacks used for evaluation are deferred to the supplementary material.

Attacks against randomized defenses: It has been pointed out by [Athalye et al., 2018b, Carlini et al., 2019] that in a white box setting, an attacker with a complete knowledge of the system will know the distribution of the noise injected in the network. As such, to create a stronger adversarial example, the attacker can take the expectation of the loss or the logits of the randomized network during the computation of the attack. This technique is called Expectation Over Transformation (EoT) and we use a Monte Carlo method with 80 simulations to approximate the best perturbation for a randomized network.

Experimental results

Trade-off between accuracy and intensity of noise: When injecting noise as a defense mechanism, regardless of the distribution it is drawn from, we observe (as in Figure 5.1(a)) that the accuracy decreases when the noise intensity grows. In that sense, noise needs to be calibrated to preserve both accuracy and robustness against adversarial attacks, i.e. it needs to be large enough to preserve robustness and small enough to preserve accuracy. Figure 5.1(a)

Table 5.1: Accuracy under attack on the CIFAR-10 dataset with a randomized Wide ResNet architecture. We compare the accuracy on natural images and under attack with different noise over 3 iterative attacks (the number of steps is next to the name) made with 80 Monte Carlo simulations to compute EoT attacks. The first line is the baseline, no noise has been injected.

Distribution	**Sd**	**Natural**	ℓ_1 **– EAD 60**	ℓ_2 **– C&W 60**	ℓ_∞ **– PGD 20**
-	-	0.958	0.035	0.034	0.384
Normal	0.01	0.954	0.193	0.294	0.408
	0.50	0.824	0.448	0.523	0.587
Laplace	0.01	0.955	0.208	0.313	0.389
	0.50	0.846	0.464	0.494	0.589

shows the loss of accuracy on CIFAR10 from 0.95 to 0.82 (respectively 0.95 to 0.84) with noise drawn from a Gaussian distribution (respectively Laplace) with a standard deviation from 0.01 to 0.5. Figure 5.1(b) and 5.1(c) illustrate the theoretical lower bound on accuracy under attack of Theorem 10 for different distributions and standard deviations. The term in entropy of Theorem 10 has been estimated using a Monte Carlo method with 10^4 simulations. The trade-off between accuracy and robustness from Theorem 10 thus appears w.r.t the noise intensity. With small noises, the accuracy is high, but the guaranteed accuracy drops fast w.r.t the magnitude of the adversarial perturbation. Conversely, with bigger noises, the accuracy is lower but decreases slowly w.r.t the magnitude of the adversarial perturbation. These Figures also show that Theorem 10 gives strong accuracy guarantees against small adversarial perturbations. Next paragraph shows that in practice, randomized networks achieve much higher accuracy under attack than the theoretical bound, and against much larger perturbations.

Performance of randomized networks under attacks and comparison to state of the art: While Figure 5.1(b) and 5.1(c) illustrated a theoretical robustness against growing adversarial perturbations, Table 5.1 illustrates this trade-off experimentally. It compares the accuracy under attack of a deterministic network with the one of randomized networks with Gaussian and Laplace noises both with low (0.01) and high (0.5) standard deviations. Randomized networks with a small noise lead to no loss in accuracy with a small robustness while high noises lead to a higher robustness at the expense of loss of accuracy ($\sim$ 11 points). Table 5.2 compares the accuracy and the accuracy under attack of randomized networks with Gaussian and Laplace distributions for different standard deviations against adversarial training [Madry et al., 2018]. We observe that the accuracy on natural images of both noise injection methods are similar to the one from [Madry et al., 2018]. Moreover, both methods are more robust than adversarial training to PGD and C&W attacks. With all the experiments, to construct an EoT attack, we use 80 Monte Carlo simulations at every step the attacks. These experiments show that randomized defenses can be competitive given the intensity of noise injected in the network. Note that these experiments have been led with EoT of size 80. For much bigger sizes of EoT these results would be mitigated. Nevertheless, the accuracy would never drop under the bounds illustrated in Figure 5.1, since Theorem 10 gives a bound that on the worst case attack strategy (including EoT).

Table 5.2: Accuracy under attack of randomized neural network with different distributions and standard deviations versus adversarial training by Madry et al. [Madry et al., 2018]. The PGD attack has been made with 20 step, an epsilon of 0.06 and a step size of 0.006 (input space between -1 and $+1$). The Carlini&Wagner attack uses 30 steps, 9 binary search steps and a 0.01 learning rate. The first line refers to the baseline without attack.

Attack	**Steps**	[Madry et al., 2018]	**Normal 0.32**	**Laplace 0.32**	**Normal 0.5**	**Laplace 0.5**
-	-	0.873	0.876	0.891	0.824	0.846
ℓ_∞ – PGD	20	0.456	0.566	0.576	0.587	0.589
ℓ_2 – C&W	30	0.468	0.512	0.502	0.489	0.479

5.2 Introduction

Modern neural networks have been known to be sensible against small, imperceptible and adversarially-chosen perturbations of their inputs [Biggio et al., 2013, Szegedy et al., 2014b]. This vulnerability has become a major issue as more and more neural networks have been deployed into production applications. Over the past decade, the research progress plays out like a cat-and-mouse game between the development of more and more powerful attacks [Carlini et al., 2017, Croce et al., 2020b, Goodfellow et al., 2015, Kurakin et al., 2016] and the design of empirical defense mechanisms [Cohen et al., 2019, Madry et al., 2018, Moosavi-Dezfooli et al., 2019]. Finishing the game calls for certified adversarial robustness [Raghunathan et al., 2018, Wong et al., 2018]. While recent work devised defenses with theoretical guarantees against adversarial perturbations, they share the same limitation, i.e.,, the tradeoffs between expressivity and robustness, and between scalability and accuracy.

A natural approach to provide robustness guarantees on a classifier is to enforce Lipschitzness properties. To achieve such properties, researchers mainly focused on two different kinds of approaches. The first one is based on randomization [Cohen et al., 2019, Lecuyer et al., 2018, Pinot et al., 2019] and consists in convolving the input with with a predefined probability distribution. While this approach offers some level of scalability (i.e.,, currently the only certified defense on the ImageNet dataset), it suffers from significant impossibility results Yang et al. [2020]. A second approach consists in building 1-Lipschitz layers using specific linear transform [Anil et al., 2019, Cisse et al., 2017, Li et al., 2019b,b, Singla and Feizi, 2021, Singla et al., 2021a, Trockman et al., 2021]. Knowing the Lipschitz constant of the network, it is then possible to compute a certification radius around any points.

A large line of work explored the interpretation of residual neural networks He et al. [2016b] as a parameter estimation problem of nonlinear dynamical systems [E, 2017, Haber et al., 2017, Lu et al., 2018]. Reconsidering the ResNet architecture as an Euler discretization of a continuous dynamical system yields to the trend around Neural Ordinary Differential Equation [Chen et al., 2018b]. For instance, in the seminal work of Haber et al. [2017], the continuous formulation offers more flexibility to investigate the stability of neural networks during inference, knowing that the discretization will be then implemented by the architecture design. The notion of stability, in our context, quantifies how a small perturbation on the initial value impacts the trajectories of the dynamical system.

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.

5.3 Background and Related Work

In this paper, we aim at devising *certified* defense mechanisms against adversarial attacks, in the following, we formally define an adversarial attacks and a robustness certificate. We consider a classification task from an input space $\mathcal{X} \subset \mathbb{R}^d$ to a label space $\mathcal{Y} := \{1, \dots, K\}$. To this end, we aim at learning a classifier function $\mathbf{f} := (f_1, \dots, f_K) : \mathcal{X} \to \mathbb{R}^K$ such that the predicted label for an input x is $\text{argmax}_k f_k(x)$. For a given couple input-label (x, y), we say that x is correctly classified if $\text{argmax}_k f_k(x) = y$.

Definition 17 (**Adversarial Attacks**). *Let $x \in \mathcal{X}$, $y \in \mathcal{Y}$ the label of x and let $\mathbf{f}$ be a classifier. An adversarial attack at level ε is a perturbation τ s.t. $\|\tau\| \leq \varepsilon$ such that:*

$$\underset{k}{\text{argmax}}\, f_k(x + \tau) \neq y$$

Let us now define the notion of robust certification. For $x \in \mathcal{X}$, $y \in \mathcal{Y}$ the label of x and let $\mathbf{f}$ be a classifier, a classifier $\mathbf{f}$ is said to be *certifiably robust at radius $\varepsilon \geq 0$* at point x if for all τ such that $\|\tau\| \leq \varepsilon$:

$$\underset{k}{\text{argmax}}\, f_k(x + \tau) = y$$

The task of robust certification is then to find methods that ensure the previous property. A key quantity in this case is the Lipschitz constant of the classifier.

5.3.1 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., 2014b, 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 17 (Tsuzuku et al. [2018])**.** *Let $\mathbf{f}$ 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 $\|\tau\|_2 \leq \varepsilon$:

$$\underset{k}{\text{argmax}}\, f_k(x + \tau) = y$$

From Proposition 17, 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.

5.3.2 Certified Adversarial Robustness

Mainly two kinds of methods have been developed to come up with certified adversarial robustness. The first category relies on randomization and consists of convolving the input with a predefined probability distribution during both training and inference phases. Several works that rely on the method have proposed empirical Cao and Gong [2017], Liu et al. [2018], Pinot et al. [2019, 2020] and certified defenses Cohen et al. [2019], Lecuyer et al. [2018], Li et al. [2019a], Salman et al. [2019], Yang et al. [2020]. These methods are model-agnostic, in the sense they do not depend on the architecture of the classifier, and provide "high probability" certificates. However, this approach suffers from significant impossibility results: the maximum radius that can be certified for a given smoothing distribution vanishes as the dimension increases Yang et al. [2020]. Furthermore, in order to get non-vacuous provable guarantees, such approaches often require to query the network hundreds of times to infer the label of a single image. This computational cost naturally limits the use of these methods in practice.

The second approach directly exploits the Lipschitzness property with the design of built-in 1-Lipschitz layers. Contrarily to previous methods, these approaches provide deterministic guarantees. Following this line, 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., 2019b, Singla and Feizi, 2021, Trockman et al., 2021]. The work of Li et al. [2019b], Trockman et al. [2021] and Singla and Feizi [2021] (denoted BCOP, Cayley and SOC respectively) are considered the most relevant approach to our work. Indeed, their 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. The BCOP layer (Block Convolution Orthogonal Parameterization) uses an iterative algorithm proposed by Björck et al. [1971] to orthogonalize the linear transform performed by a convolution. The SOC layer (Skew Orthogonal Convolutions) uses the property that if A is a skew symmetric matrix then $Q = \exp A$ is an orthogonal matrix. To approximate the exponential, the authors proposed to use a finite number of terms in its Taylor series expansion. Finally, the 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 $Q = (I - A)^{-1}(I + A)$. Both methods are well adapted to convolutional layers and are able to reach high accuracy levels on CIFAR datasets. 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.

5.3.3 Residual Networks

To prevent from gradient vanishing issues in neural networks during the training phase [Hochreiter et al., 2001], He et al. [2016b] proposed the Residual Network (ResNet) architecture. Based on this 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:*

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

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[1], the stability can be characterized by the eigenvalues of the Jacobian matrix $\nabla_x F_t(x_t)$: 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., 2020] 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.

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

5.4 A Framework to design Lipschitz Layers

The continuous time interpretation of Definition 18 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 $\|x_t - z_t\|$ only involves the symmetric part $S(\nabla_x F_t(x))$ (see Appendix 5.9.1 for details).

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 in Appendix 5.9.1 the following proposition.

Proposition 18. *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:

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

The symmetric part plays even a more important role since one can show that a function whose Jacobian is always skew-symmetric is actually linear (see Appendix 5.10.1 for more details). However, 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. See proof in Appendix 5.9.2

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:

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

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.

5.4.1 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:

$$\|x_{t+1} - z_{t+1}\| - \|x_t - z_t\| = \|A_t (x_t - z_t)\|.$$

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.

5.4.2 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 19. *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

$$\|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}}\|_2 \leq \|x_t - z_t\|_2$$

In Section 5.5, 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 5. *Another solution relies on the implicit Euler scheme: $x_{t+\frac{1}{2}} = x_t - \nabla_x f_t(x_{t+\frac{1}{2}})$. We show in Appendix 5.10.2 that this strategy defines a 1-Lipschitz flow without further assumption on f_t than convexity. We propose an implementation. However preliminary experiments did not show competitive results and the training time is prohibitive. We leave this solution for future work.*

5.4.3 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].

5.5 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 paper we focus on the parametrization of symmetric update with the convex potentials proposed in 19. For that purpose, the Input Convex Neural Network (ICNN) [Amos et al., 2017] provide a relevant starting point that we will extend.

5.5.1 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 $\mathbf{W}$, i.e.,, the greatest singular value of $\mathbf{W}$ defined as:

$$\|\mathbf{W}\|_2 := \max_{x \neq 0} \frac{\|\mathbf{W}x\|_2}{\|x\|_2}$$

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.

5.5.2 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[2] instead of generic linear transforms represented by a dense matrix.

Remark 6. *When $\mathbf{W} \in \mathbb{R}^{1 \times d}$, $b = 0$ and $\sigma =$ ReLU, the* Convex Potential Layer *is equivalent to the HouseHolder activation function introduced in Singla et al. [2021a].*

Residual Networks [He et al., 2016b] 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.

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.

[2]For instance, one can leverage the `Conv2D` and `Conv2D_transpose` functions of the PyTorch framework [Paszke et al., 2019]

Algorithm 3: Computation of a Convex Potential Layer

Require: **Input:** x**, vector:** u**, weights:** $\mathbf{W}$**,** b
Ensure: Compute the layer z and return u

$$\left.\begin{array}{l} v \leftarrow \mathbf{W}u/\|\mathbf{W}u\|_2 \\ u \leftarrow \mathbf{W}^\top v/\|\mathbf{W}^\top v\|_2 \\ h \leftarrow 2/\left(\sum_i (\mathbf{W}u \cdot v)_i\right)^2 \end{array}\right\} \begin{array}{l} \text{1 iter. for training} \\ \text{100 iter. for inference} \end{array}$$

return $x - h\left[\mathbf{W}^\top \sigma(\mathbf{W}x + b)\right], u$

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. In this paper, we simply implement it as the truncation of the output. This obviously maintains the Lipschitz constant.

5.5.3 Computing spectral norms

Our Convex Potential Layer, described in Equation 5.5.2, 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 converges along with the training process[3]. We describe with more details in Algorithm 3, 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

[3]Note that a typical training requires approximately 200K steps where 100 steps of PM is usually enough for convergence

each layer[4] at inference time. Also note that at inference time, the computation of the spectral norm only needs to be performed once for each layer.

[4] 100 iterations of Power Method is sufficient to converge with a geometric rate.

#	S	M	L	XL
Conv. Layers	20	30	90	120
Channels	45	60	60	70
Lin. Layers	7	10	15	15
Lin. Features	2048	2048	4096	4096

Table 5.3: 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. In the paper, they will be reported respectively as CPL-S, CPL-M, CPL-L and CPL-XL.

5.6 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 5.6.1. By computing the certified and empirical adversarial accuracy of our networks on CIFAR10 and CIFAR100 classification tasks [Krizhevsky et al., 2009], we show that our architecture is competitive with state-of-the-art methods (Sections 5.6.3). In Appendix 5.11, 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.

5.6.1 Training and Architectural Details

We demonstrate the effectiveness of our approach on a classification task with CIFAR10 and CIFAR100 datasets [Krizhevsky et al., 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 5.3, respectively named CPL-S, CPL-M, CPL-L and CPL-XL, ranked according to the 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.

5.6.2 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

	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 5.4: 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.

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 paper is not open-sourced yet, so we just reported the results from their paper in ours results (Tables 5.4 and 5.5) 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).

5.6.3 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 5.4 and 5.5. We also plotted certified accuracy in function of ε on Figure 5.2. 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.

	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 5.5: 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.

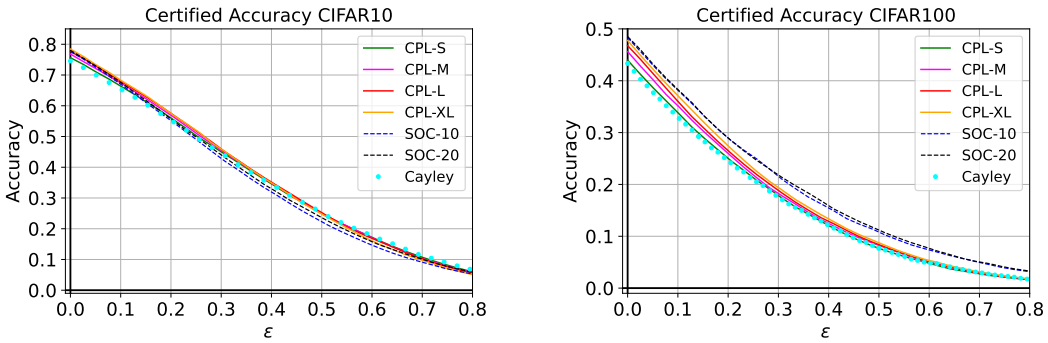


Figure 5.2: Certifiably robust accuracy in function of the perturbation ε for our CPL networks and its concurrent approaches (SOC and Cayley models) on CIFAR10 and CIFAR100 datasets.

Empirical Adversarial Robustness. We also reported in Figure 5.3 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 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.

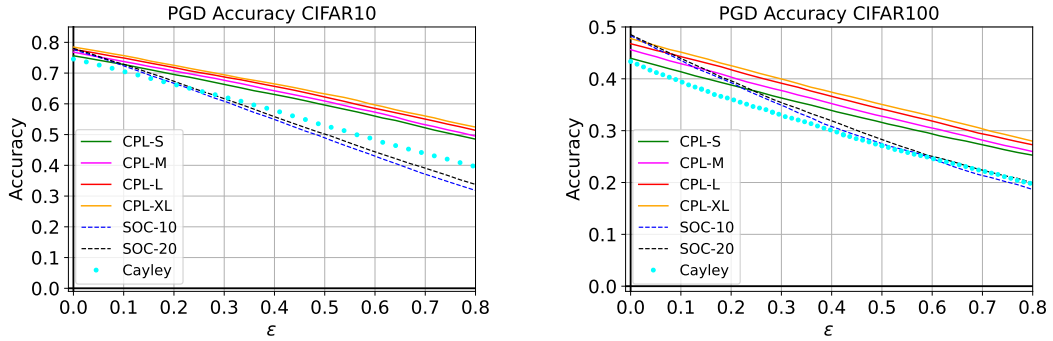


Figure 5.3: Accuracy against PGD attack with 10 iterations in function of the perturbation ε for our CPL networks and its concurrent approaches on CIFAR10 and CIFAR100 datasets.

5.7 Conclusion

In this paper, 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.

Further work. Propoisition 18 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 (see Appendix 5.10.3). This is an important question that requires further investigation. 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., 2021] exhibit interesting properties. Finally, we hope to use similar approaches to build robust Recurrent Neural Networks [Sherstinsky, 2020] and Transformers [Vaswani et al., 2017].

5.8 Further Related Work

Lipschitz Regularization for Robustness. 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. 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., 2021, 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.

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.

5.9 Proofs

5.9.1 Proof of Proposition 18

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. □

5.9.2 Proof of Corollary 3

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 18, we deduce $\|x_t - y_t\| \leq \|x_0 - y_0\|$ for all trajectories starting from x_0 and y_0. □

5.9.3 Proof of Proposition 19

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. □

5.10 Additional Results

5.10.1 Functions whose gradient is skew-symmetric everywhere

Let $F := (F_1, \ldots, F_d) : \mathbb{R}^d \to \mathbb{R}^d$ 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 $F(x) = Ax$

5.10.2 Implicit discrete convex potential flows

Let us define the implicit update $x_{t+\frac{1}{2}} = x_t - \nabla_x f_t(x_{t+\frac{1}{2}})$. 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.

To compute such a layer, one could solve the proximal operator strongly convex-minimization optimization problem. This strategy is not computationally efficient and not scalable.

5.10.3 Expressivity of discretized convex potential flows

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[5] 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 19. 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 $M \in \mathcal{P}(\mathbb{R}^{d\times d})$.

5.11 Additional experiments

5.11.1 Training stability: scaling up to 1000 layers

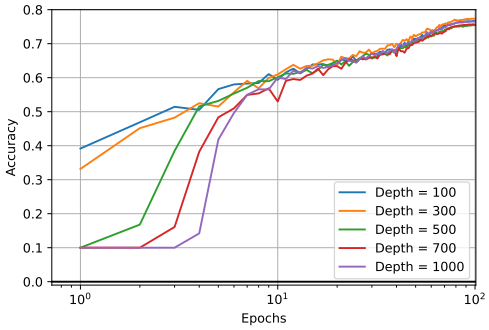


Figure 5.4: Standard test accuracy in function of the number of epochs (log-scale) for various depths for our neural networks $(100, 300, 500, 700, 1000)$.

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

[5] 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`

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 5.4 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.

5.11.2 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 19. 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.

5.11.3 Effect of Batch Size in Training

In Tables 5.6 and 5.7, we tried three different batch sizes (64, 128 and 256) for training our networks on CIFAR10 and CIFAR100 datasets, we remark 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.

	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 5.6: 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.

5.11.4 Effect of the Margin Parameter

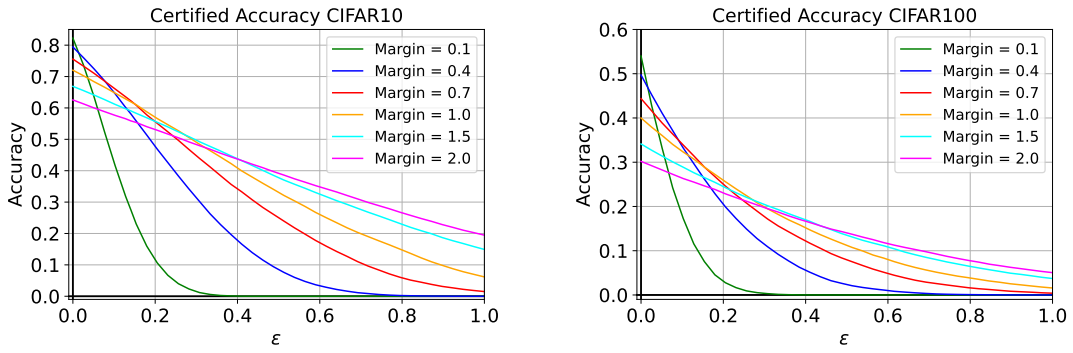


Figure 5.5: Certifiably robust accuracy in function of the perturbation ε for our CPL-S network with different margin parameters on CIFAR10 and CIFAR100 datasets.

In these experiments we varied the margin parameter in the margin loss in Figures 5.5 and 5.6. 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.

	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 5.7: 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.

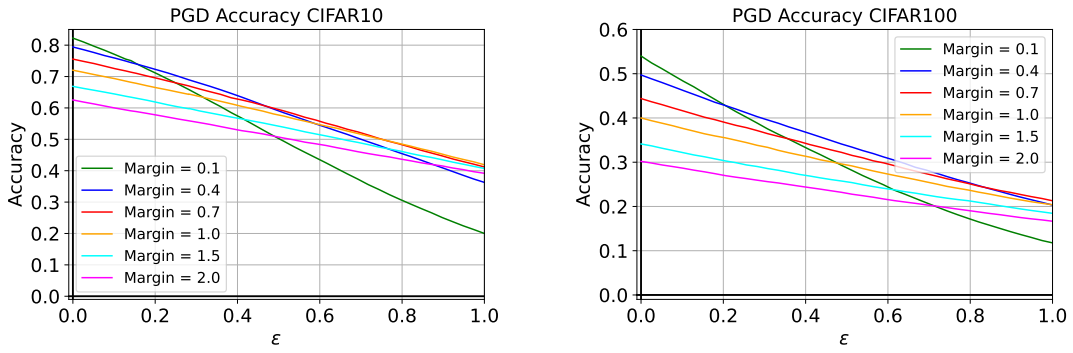


Figure 5.6: Certifiably robust accuracy in function of the perturbation ε for our CPL-S network with different margin parameters on CIFAR10 and CIFAR100 datasets.

Chapter 6

Conclusion

Contents

6.1 Open Questions

6.1.1 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

6.1.2 Loss Calibration General Results

- The non realisable case is difficult: showing either negative/positive general results
- Further developing the margin loss analysis

6.1.3 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.

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, 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`.

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, Anqi Mao, Mehryar Mohri, and Yutao Zhong. A finer calibration analysis for adversarial robustness. *arXiv preprint arXiv:2105.01550*, 2021b.

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.

Normand J. Beaudry and Renato Renner. An intuitive proof of the data processing inequality. *Quantum Info. Comput.*, 12(5-6):432–441, May 2012. ISSN 1533-7146.

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.

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*, 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.

Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287, 2017.

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, 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.

Nicholas Carlini et al. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

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 *AAAI*, 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.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, 2017.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 2019.

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.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Zac Cranko, Aditya Menon, Richard Nock, Cheng Soon Ong, Zhan Shi, and Christian Walder. Monge blunts bayes: Hardness results for adversarial training. 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 1406–1415. PMLR, 09–15 Jun 2019. URL `http://proceedings.mlr.press/v97/cranko19a.html`.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020b.

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.

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.

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.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018.

Jimmy Ba Diederik P. Kingma. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2014.

Dimitrios Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarial risk and robustness: General definitions and implications for the uniform distribution. In *Advances in Neural Information Processing Systems*, pages 10380–10389, 2018.

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.

Farzan Farnia, Jesse Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *International Conference on Learning Representations*, 2019.

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.

Gene H Golub et al. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 2000.

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.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 2021.

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.

Eldad Haber et al. Stable architectures for deep neural networks. *Inverse problems*, 2017.

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016b.

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.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR, 2018.

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.

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.

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.

Michael Kearns and Ming Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 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.

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, 2020.

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/*.

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)*, 2018.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, 2019a.

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, 2020.

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*, 2019b.

Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision*, 2018.

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.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. 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.

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.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

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.

Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6): 1191–1253, 2003.

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*, 2016.

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.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12: 2825–2830, 2011.

Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization. In *Advances in Neural Information Processing Systems*, 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. In *International Conference on Machine Learning*. 2020.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. *arXiv preprint arXiv:1811.09310*, 2018.

Jeremy Rapin and Olivier 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.

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.

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.

Michael E. Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Momentum residual neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Hanie Sedghi, Vineet Gupta, and Philip Long. The singular values of convolutional layers. In *International Conference on Learning Representations*, 2018.

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.

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.

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.

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*, 2014a.

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*, 2014b.

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.

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.

Zhuozhuo Tu, Jingwei Zhang, and Dacheng Tao. Theoretical analysis of adversarial learning: A minimax approach. *arXiv preprint arXiv:1811.05232*, 2018.

AM Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

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.

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.

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.

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.

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, 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.

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.

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, 2020.

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 *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, 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*, 2019.