XX

xx

Laurent Meunier

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Université Paris-Dauphine – PSL Research University

in collaboration with

Meta AI Research

under the joint supervision of

Pr. Jamal ATIF and Dr. Olivier Teytaud

Abstract

This thesis investigates the problem of classification in presence of advesarial attacks. An adversarial attack is a small and humanly imperceptible perturbation of input designed to fool start-of-the-art machine learning classifiers. In particular, deep learning systems, used in safety critical AI systems as self-driving cars are at stake with the eventuality of such attacks. What is even more striking is the ease to create such adversarial examples and the difficulty to defend against them while keeping a high level of accuracy. Robustness to adversarial perturbations is a still misunderstood field in academics. In this thesis, we aim at understanding better the nature of the adversarial attacks problem from a theoretical perspective.

Can we find a principled way to defend against adversarial examples?

In a first part, we tackle the problem of adversarial examples from a game theoretic point of view. We study the open question of the existence of mixed Nash equilibria in the zero-sum game formed by the attacker and the classifier. To that extent, we consider a randomized classifier and we introduce a more general attacker that can move each point randomly in the vinicity of original points. While previous game theoretic approaches usually allow only one player to use randomized strategies, we show the necessity of considering randomization for both the classifier and the attacker. We demonstrate that this game has no duality gap, meaning that it always admits approximate Nash equilibria. We also provide the first optimization algorithms to learn a mixture of a finite number of classifiers that approximately realizes the value of this game, i.e. procedures to build an optimally robust randomized classifier.

In a second part, we study the problem of surrogate losses in the adversarial examples case. In classification, the goal is to maximize the accuracy, but in practice, the accuracy is not efficiently optimizable. Instead, it is usual to minimize a convex and continuous loss that satisfy what is called the *consistency property*. In the adversarial case, we tackle this problem and show that a wide range of usually consistent losses cannot be consistent. In particular, convex losses are not good surrogate losses for the adversarial attack problem. Finally, we pave a way towards designing a class of consistent losses, but this question is partially treated and left as further work.

In a final section, we study the robustness of neural networks from a dynamical system perspective. Residual Networks can indeed be interepreted as a discretization of a first order parametric differential equation. By studying this system, we provide a generic method to build 1-Lipschitz Neural Networks and show that some previous approaches are special cases of this framework. We extend this reasoning and show that ResNet flows derived from convex potentials define 1-Lipschitz transformations, that lead us to define the Convex Potential Layer (CPL).

Contents

List of Figures and Tables

Notations and Symbols

We use bold lower-case to denote vectors and functions with multidimensional outputs and standard lower-case to denote scalars and real-value functions. Depending on the context, we either use calligraphic font or upper-case to denote ensembles – most of the times calligraphic, sometimes upper-case to denote sub-sets or elements of a set of sets.

Algebra

$\mathbb{R}$	Set of real numbers	
$\mathbb{N}$	Set of natural integers	
$\mathbb{R}^d$	Set of d-dimensional real-valued vectors	
$\mathbb{R}^{d\times d'}$	Set of $d \times d'$ real-valued matrices	
I_d	$d \times d$ identity matrix	
$(\mathcal{Z}, d)$	$\mathcal{Z}$ space endowed with metric d	
$[\![a]\!]$	Set of integers between 1 and a	$[a] := \{1, \dots, a\}$
Δ_K	K dimensional simplex	$\Delta_K := \{\mathbf{p} \in \mathbb{R}_+^K \mid \|\mathbf{p}\|_1 = 1\}$
$\|x\|_p$	ℓ_p-norm of $x \in \mathbb{R}^d$ for $p \in [1, +\infty)$	$\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^d \|\boldsymbol{v}_i\|^p\right)^{1/p}$
$\|\boldsymbol{v}\|_\infty$	Infinite norm of $\boldsymbol{v} \in \mathbb{R}^d$	$\|\boldsymbol{v}\|_\infty = \max_{i\in[d]}(\|\boldsymbol{v}_i\|)$
$B_d(x, \alpha)$	d-ball with center $x \in \mathcal{X}$ and radius $\alpha \geq 0$	$\{y \mid \|y - x\|_p \leq \alpha\}$

Probability

$\mathcal{B}(\mathcal{Z})$	Borel σ-algebra of a space $(\mathcal{Z}, d)$
$\mathcal{M}(\mathcal{Z})$	Set of radon measures distribution over $(\mathcal{Z}, d)$
$\mathcal{M}_1^+(\mathcal{Z})$	Set of Borel probability distributions over $(\mathcal{Z}, d)$
$\mathcal{F}(\mathcal{Z}, \mathcal{Z}')$	Set of measurable functions from $\mathcal{Z}$ to $\mathcal{Z}'$
$\boldsymbol{\psi}\#\rho$	Push-forward of $\rho \in \mathcal{P}(\mathcal{Z})$ by $\boldsymbol{\psi} \in \mathcal{F}(\mathcal{Z}, \mathcal{Z}')$
$\mathbb{P}[.]$	Probability of a random event
$\mathbb{E}_{\mathbb{P}}[.]$	Expectation of a random event under the probability $\mathbb{P}$
$\mathcal{N}(.,.)$	Gaussian distribution
$\mathrm{Lap}(.,.)$	Laplace distribution
Φ	cdf of the standard Gaussian distribution $\mathcal{N}(0, 1)$

Classification and Learning theory

$\mathcal{X}$	Input space
d	distance on the input space
$\mathcal{Y}$	Output (Label) space
K	Number of classes
$\mathbb{P}$	Ground-truth distribution
S	Training sample
$\mathcal{H}$	Hypothesis space
L	Loss function

Functions

$\mathbf{1}_{\{.\}}$	Indicator function of an event	$\mathbf{1}_A = 1$ if A is true, 0 otherwise
$\mathrm{sign}(x)$	Sign function applied on x	$\mathrm{sign}(x) = 1$ if $x > 0$, -1 if $x < 0$ and 1 if $x = 0$
$\nabla_x f$	Gradient of f with regards to x	

Abbreviations

a.k.a.	**a**lso **k**nown **a**s
cdf	**c**umulative **d**ensity **f**unction
C & W	**C**arlini and **W**agner (attack)
e.g.	***e**xempli **g**ratia*
DLR	**D**ifference of **L**ogit **R**ation (attack)
Eq.	**Eq**uation
FGM	**F**ast **G**radient **M**ethod (attack)
i.e.	***i**d **e**st*
i.i.d.	**i**dentically and **i**ndependently **d**istributed
PGD	**P**rojected **G**radient **D**escent (attack)
resp.	**resp**ectively
s.t.	**s**uch **t**hat
std	**st**andard **d**eviation
w.r.t.	**w**ith **r**espect **t**o

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 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 lot 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 specialized 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 theorized 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 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 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.

In response 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., 2014]**: small imperceptible perturbations at inference time. We will refer them to *"adversarial attacks"*.

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

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

As part of the introduction: *"Cybersecurity plays a crucial role in ensuring that AI systems are resilient against attempts to alter their use, behavior, 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 unauthorized 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

[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

AI is referred to any autonomous system than can endanger human lives. This text aims at dealing with many threats in Learning Systems. Two direct references are made to adversarial attacks, underlying the need for companies to deal with them. The difficulty is to unify and create precise rules in a domain where results and certificates are mostly empirical. As mentioned earlier, it is known that robust models are often less performing and can make autonomous systems unusable in real world scenarii. Thus, this text is a first step towards a unified regulation on autonomous systems but might miss precise requirements for models to be used in production.

1.2.2 Adversarial attacks against Machine Learning Systems

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

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

[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.

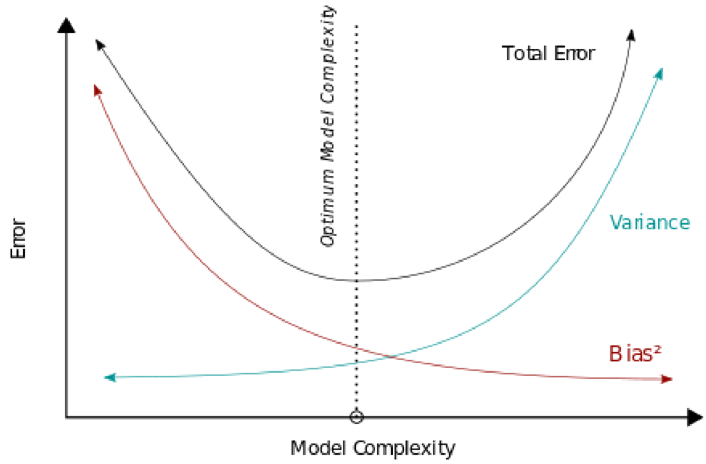


Figure 1.1: Bias-Variance tradeoff. A model with low complexity will have a low variance but a high bias. A model with high complexity will have a low bias but a high variance.

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 learned model, we evaluate it on other images that constitute the test set.

1.3.1 A Learning Approach for Classification

From now, we will assume that the inputs are in some space $\mathcal{X}$ and the labels form a set $\mathcal{Y} := \{1, \dots, K\}$. To learn an adequate classification model, we denote $\{(x_1, y_1), \dots, (x_N, y_N)\}$ 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 of 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-complexity tradeoff (see Figure 1.1). A penalty term $\Omega_{\mathcal{H}}(\mathbf{f})$ can also be added to the ERM objective to prevent from overfitting. This tradeoff was recently questioned by the double descent [Belkin et al., 2019] phenomenon where overparametrized (i.e. number of parameters largely over the number of training samples) regimes lower the risk.

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

1.3.2 Classification in Presence of Adversarial Attacks

Though a model can be very well performing on natural samples, small perturbations of these natural samples can lead to unexpected and critical behaviors of classification models [Biggio et al., 2013, Szegedy et al., 2014]. To formalize that, we will assume the existence of a "perception" distance $d : \mathcal{X}^2 \to \mathbb{R}$ such that a perturbation x' of an input x remains imperceptible if $d(x, x') \leq \varepsilon$ for some constant $\varepsilon \geq 0$. This "perception" distance is difficult to define in practice. For images, the $\|\cdot\|_\infty$ distance over pixels is often used, but is not able to capture all imperceptible perturbations. This choice is purely arbitrary: for instance, we will highlight in the manuscript that $\|\cdot\|_2$ perturbations can also be imperceptible while having a large $\|\cdot\|_\infty$. Image classification algorithms are also vulnerable to geometric perturbations, i.e. rotations and translations [Kanbak et al., 2018, Engstrom et al., 2019]. A typical example of an adversarial attack is shown in Figure 1.2.

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:

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

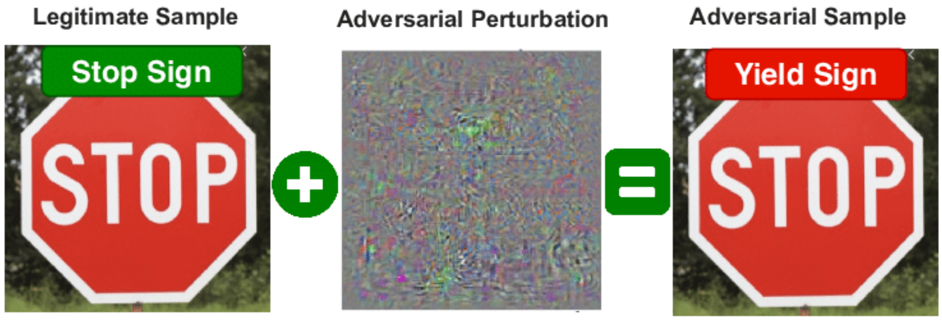


Figure 1.2: Example of a pixel-level adversarial attack on a Stop Sign. It underlines the safety issues triggered by the possibility of such attacks.

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

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

This problem is more challenging to tackle than the standard risk minimization since it involves a hard inner supremum problem [Madry et al., 2018b]. 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., 2015b, Madry et al., 2018b]. It consists in alternating inner and outer optimization problems. Such a technique improves in practice adversarial robustness but lacks theoretical guarantees. So far, most results and advances in understanding and harnessing adversarial attacks are empirical [Ilyas et al., 2019, Rice et al., 2020], leaving many theoretical and practical questions open. Moreover, robust models suffer from a performance drop and vulnerability of models is currently still very high (see Table 1.3), 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 1.3: State of the art accuracies on adversarial tasks on a WideResNet 28x10 [Zagoruyko and Komodakis, 2016]. Results are reported from [Croce et al., 2020a]

1.4 Outline and Contributions

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

1.4.1 A Game Theoretic Approach to Adversarial Attacks

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

Question 1

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

In game theory, there are many types of equilibria. In this manuscript, we will focus on Stackelberg and Nash equilibria. We will show the existence of both when both the classifier and the attacker play randomized strategies. To reach such equilibria, the classifier will be random, and the attacker will move randomly the samples at a maximum distance of ε. Then, we will propose two different algorithms to compute the optimal randomized classifier in the case of a finite number of possible classifiers. We will finally propose a heuristic algorithm to train a mixture of neural networks and show experimentally the improvements we achieve over standard methods. This work was published at ICML2021 [Meunier et al., 2021e].

1.4.2 Loss Consistency in Classification in Presence of an Adversary

In standard classification, consistency with regard 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 $(f_n)_{n\in\mathbb{N}}$ 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 case is much harder.

Question 2

Which losses are consistent with regard 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. We pave a way towards solving this question by proposing candidate losses and giving arguments for their consistency. This work has not been submitted for peer-reviewing yet.

1.4.3 Building Certifiable Models

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

Question 3

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

To this end, we propose a general method that enforces Lipschitzness on the predictions of neural networks. This method draws its inspiration from the continuous flow interpretation of residual networks. We provide discretization strategies and recover many existing methods to build 1-Lipschitz layers in neural networks. In particular, we show that using a gradient flow of a convex function, our network is 1-Lipschitz. Based on this insight, we design a Lipschitz layer, that we call Convex Potential Layer (CPL). We show empirically and theoretically the robustness benefits of this approach. This work is under review and its preprint is available [Meunier et al., 2021a]

1.4.4 Additional Works

In addition 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 additional works complete the study we lead in the main document:

- **On the robustness of randomized classifiers to adversarial examples (see Appendix A):** 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 was published at NeurIPS2019 [Pinot et al., 2019] and under review in an extended journal version [Pinot et al., 2021].

- **Yet another but more efficient black-box adversarial attack: tiling and evolution strategies (see Appendix B):** we provide a method based on evolutionary strategies to craft black-box adversarial attacks. This work is a preprint and has not been published [Meunier et al., 2019].

- **Advocating for Multiple Defense Strategies against Adversarial Examples (see Appendix C):** We show that, in high-dimensional settings, the balls overlaps for two different ℓ^p norms are fundamentally different. This induces to rethink robustness against attacks using different norms. This work was published at a workshop at ECML2020 [Araujo et al., 2020].

- **Adversarial Attacks on Linear Contextual Bandits (see Appendix D):** we build provable attacks against online recommendation systems, namely Linear Contextual Bandits. This work was published at NeurIPS2020 [Garcelon et al., 2020].

- **ROPUST: Improving Robustness through Fine-tuning with Photonic Processors and Synthetic Gradients (see Appendix E):** we use an Optical Processor Unit (OPU) over existing state-of-the-art defenses to improve adversarial robustness. This work was published at ICASSP2022 [Cappelli et al., 2021b].

We published a paper in optimal transport on **Equitable and Optimal Transport with Multiple Agents (see Appendix F)** at AISTATS2021 [Scetbon et al., 2021a] where we introduce a way to deal with multiple costs in optimal transport by equitably partitioning transport among costs. We also a published a paper at xxx on Conditional Independence Testing [Scetbon et al., 2021b] on **an ℓ^p-based Kernel Conditional Independence Test (see Appendix G)**. In this paper we present a new kernel-based Conditional Independence Test. Its advantages are its computational simplicity and a very simple asymptotic distribution under null hypothesis. Moreover, it performs competitively with other test for Conditional Independence.

With Olivier Teytaud, research scientist at Meta AI and co-supervisor of this thesis, we also published some works in the field of evolutionary algorithms:

- **Variance Reduction for Better Sampling in Continuous Domains (see Appendix H)**: 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 [Meunier et al., 2020c].

- **On averaging the best samples in evolutionary computation (see Appendix I):** we prove mathematically that a single parent leads to a suboptimal 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 [Meunier, Chevaleyre, Rapin, Royer, and Teytaud, 2020a].

- **Asymptotic convergence rates for averaging strategies (see Appendix J):** we extend the results from the previous paper to a wide class of functions including C^3 functions with unique optima. This work was published at FOGA2021 [Meunier et al., 2021b].

- **Black-Box Optimization Revisited: Improving Algorithm Selection Wizards through Massive Benchmarking (see Appendix K):** We propose a wide range of benchmarks integrated in Nevergrad [Rapin and Teytaud, 2018] platform. This work was published in the journal TEVC [Meunier et al., 2021c].

2 Background

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

Contents

2.1 Supervised Classification

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

2.1.1 Notations

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

- Consider an input space $\mathcal{X}$, typically images. We assume this space is endowed with an arbitrary metric d, possibly a perception distance or any ℓ_p norm. In the following of the manuscript, unless it is specified, $(\mathcal{X}, d)$ will be a *proper* (i.e. closed balls are compact) *Polish* (i.e. completely separable) metric space. Note that for any norm $\|\cdot\|$, $(\mathbb{R}^d, \|\cdot\|)$ is a proper Polish metric space.

- Each input $x \in \mathcal{X}$ has to be associated with a label y. A label is a descriptor of the input. The set of labels is discrete, and we designate it by $\mathcal{Y} := \{1, \dots, K\}$. $\mathcal{Y}$ is endowed with the trivial metric $d'(y, y') = \mathbf{1}_{y \neq y'}$. Note that $(\mathcal{X} \times \mathcal{Y}, d \oplus d')$ is also a proper Polish space.

The purpose of classification is to learn a classifier $h : \mathcal{X} \to \mathcal{Y}$. It is usual to learn a function $f : \mathcal{X} \to \mathbb{R}^K$ such that: $h(x) = \operatorname{argmax}_{k \in \mathcal{Y}} f_k(x)$. In a classification problem in machine learning, the data is assumed to be sampled from an unknown probability distribution $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$. We will assume from now that all the probability distributions we consider are Borel distributions. For any Polish Space $\mathcal{Z}$, we will denote $\mathcal{B}(\mathcal{Z})$ the Borel σ-algebra and the set of Borel distributions over $\mathcal{Z}$ will be denoted $\mathcal{M}_1^+(\mathcal{Z})$. We recall that on Polish space, all Borel probability distributions are Radon measures. We also recall the notion of *universal measurability*: a set $A \subset \mathcal{Z}$ is said to be universally measurable if it is measurable for every *complete* Borel probability measure. We also recall the notion of *weak topology* on the space of probability distribution: we say that $\mathbb{P}_n$ converges weakly towards $\mathbb{P}$ if for every bounded continuous function f, $\int f d\mathbb{P}_n \to \int f d\mathbb{P}$. Note that calling this property weak topology is an abuse of language, it is closer to a notion of weak-$\star$ topology.

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

2.1.2 Classification Task in Supervised Learning

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

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

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

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

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

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

Note that this quantity is well-defined when h or f is Borel or universally measurable. The optimal classifier is called the *Optimal Bayes classifier* and is defined as $h^\star(x) = \operatorname{argmax}_k \mathbb{P}(y =$

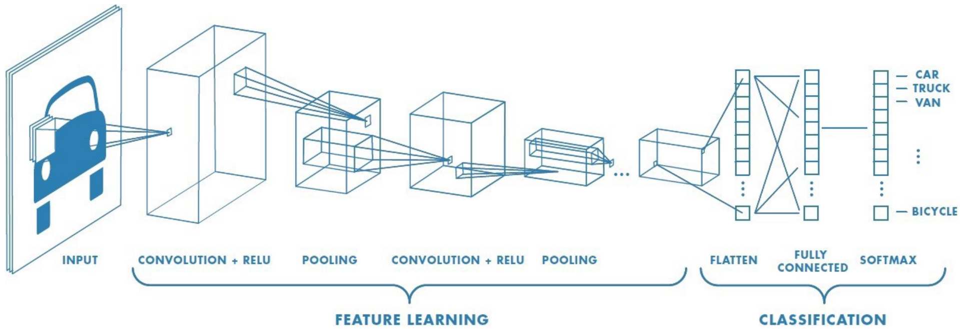


Figure 2.1: Illustration of a convolutional neural network: stacking convolutional operators and non-linear activation functions.

$k \mid x)$. We remark that the disintegration theorem ensures that $x \mapsto \mathbb{P}(y = k \mid x)$ is indeed Borel measurable.

In practice, the access to the Optimal Bayes classifier is not possible because it requires full knowledge of the probability distribution $\mathbb{P}$ which is not the case in general. Instead, in the supervised learning setting, the learner has access to data points $\{(x_1, y_1), \ldots, (x_N, y_N)\}$, that constitutes the *training set*. Knowing the Optimal Bayes classifier on training points is not sufficient to generalize on points out of the training set. Hence, one needs to reduce the search space of measurable functions to a much smaller one, denoted $\mathcal{H}$ in the sequel. The $0/1$ loss is not convex neither continuous, and minimizing directly the 0/1 loss risk on $\mathcal{H}$ might be NP-hard even for simple set of hypotheses as linear classifiers. We usually minimize a well-chosen surrogate loss function L. A *loss function* $L : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ is a non-negative Borel measurable function. An example of such a loss is the cross entropy loss defined as $L(f(x), y) = -\sum_{i=1}^{K} \mathbf{1}_{y=i} \log f_i(x)$ where $f_i(x)$ is the probability learned by the model with input x belonging to the class i. Hence, the objective is to minimize the empirical risk associated with $\mathcal{H}$ using the loss L defined as:

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

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

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

where W_i are called the weight matrices and b_i the biases. In the case of image recognition, the weights may have a special structure of convolution: such networks are called *Convolutional Networks*. We illustrate a convolutional layer in Figure 2.1.

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

2.1.3 Surrogate losses, consistency and calibration

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

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

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

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

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

We remark the notion of calibration is basically a pointwise notion of consistency with η corresponding to $\mathbb{P}(y = 1|x)$. Zhang [2004b], Bartlett et al. [2006], Steinwart [2007] proved the equivalence of the two notions in the case of standard-binary classification. In particular, they show that a wide range of convex margin losses are actually classification-consistent: if ϕ is convex and differentiable at 0, then ϕ is calibrated if and only if $\phi'(0) < 0$.

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

2.1.4 Empirical Risk Minimization and Generalization

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

where σ_i are independent samples from Rademacher law: $P[\sigma_i = +1] = P[\sigma_i = -1] = \frac{1}{2}$. The Rademacher complexity satisfy the following composition property of Rademacher complexity: when ϕ is a M-Lipschitz function:

$$\widehat{\mathrm{Rad}}_S(\phi \circ \mathcal{H}) \leq M\widehat{\mathrm{Rad}}_S(\mathcal{H})$$

This property is particularly useful because it allows to study the Rademacher complexity regardless the loss function. When $\mathcal{H}$ is not too complex (for instance, finite set or linear classifiers), one can bound the Rademacher complexity by $O(n^{-1/2})$. It was proven by Bartlett and Mendelson [2002] that the Rademacher complexity upperbounds the uniform risk error as follows:

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

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

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

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

2.2 Introduction to Adversarial Classification

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

2.2.1 What is an adversarial example?

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

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

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

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

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

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

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

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

A first property is that the adversarial risk is well-defined in the sense of measurability stated below. While this result seems trivial, it requires advanced arguments from measure theory.

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

Proof. For $h \in \mathcal{F}(\mathcal{X},\mathcal{Y})$, we define $\phi_\varepsilon(x,y,h) = \sup_{x'\in B_\varepsilon(x)} \mathbf{1}_{h(x)\neq y}$. We have :

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

Then,

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

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

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

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

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

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

2.2.2 Casting Adversarial examples

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

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

In this case the attack is said to be *untargeted*, i.e. the classifier aims at evading the label y. On the other side, a *targeted attack* aims at perturbing a label x to make it classify to a target label y. In this case, the attacker objective writes: $\min_{x' \in \mathcal{X} \text{ s.t. } d(x,x') \leq \varepsilon} L(f(x'), y))$. An attacker may also target at finding the smallest perturbation problem [Moosavi-Dezfooli et al., 2016, Carlini and Wagner, 2017]. 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 f and its parameters. Hence, these attacks often takes advantages of the differentiability of f and the loss function L. Then, such attacks usually takes the gradient $\nabla_x L\big(f\big(x^t\big), y\big)$ 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. [2018b], FGSM attack [Goodfellow et al., 2015b], 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 to design 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. [2018b] of radius ε, recursively computes*

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

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

Black box attacks: In this setting, the attacker has limited knowledge of the classifier. The attacker does not have 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 in the ZOO attack [Chen et al., 2017] and in the NES attack [Ilyas et al., 2018a, 2019]. Attacks might also be based on other optimization methods such as combinatorial methods [Moon et al., 2019] or evolutionary computation [Andriushchenko et al., 2019].

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

- **Image Segmentation and Object Detection:** Xie et al. [2017] proposed to attack image segmentation and object detection. The goal of such attack is to enforce an undesirable detection or segmentation in an image.
- **Video classification:** Videos are series of images. Adversarial attacks against video classification systems are close to adversarial examples in standard Image Classification. Adversarial attacks might aim at changing either a bit many frames [Jiang et al., 2019] or a lot only a few frames [Mu et al., 2021].
- **Audio systems:** Audio systems can be fooled by adding inaudible adversarial noise to an audio file [Carlini and Wagner, 2018]. These attacks raise issues in the massive use of personal vocal assistants [Zhang et al., 2019b].
- **NLP classification tasks:** Adversaries change some words in a text to make it misclassified. However, such examples can also change the meaning of the text and consequently change its classification also to humans. Examples of attempts for adversarial examples against NLP systems can be either black box [Jin et al., 2019, Li et al., 2020a] or gradient-based [Guo et al., 2021]
- **Recommender Systems** A recent line of work Jun et al. [2018], Liu and Shroff [2019], 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 choose the wrong arms a linear number of times. While these works are mostly theoretical, their potential use in practical settings might raise issues for businesses in a close future.

2.2.3 Defending against adversarial examples

Defending against adversarial examples is still an open research question with only a few answers to it. One can classify 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 be *certifiably robust at level* ε at input x with label y if there exist no adversarial example of level ε on h at the point (x, y), i.e. for all x' such that

$d(x, x') \leq \varepsilon, h(x') = y$. Researchers have focused on finding ways to certify robustness. The first categories of defenses rely on convex relaxation of layers [Wong and Kolter, 2018, Wong et al., 2018]. It consists in considering a convex outer approximation of the set of activations reachable through a norm-bounded perturbation of an input. In the case of ReLU activation, the robust optimization problem that minimizes the worst case loss over this outer region writes as a linear program. Another developed method is noise injection to the input [Lecuyer et al., 2019, Cohen et al., 2019, Pinot et al., 2019, Salman et al., 2019]. By adding a noise, the inputs can be seen as distributions. The certificates are derived by determining which classifier would be the most powerful to distinguish two inputs. This idea is closely related to the notions of statistical tests [Cohen et al., 2019], information theory [Pinot et al., 2019] and differential privacy [Lecuyer et al., 2018]. Finally, a last trend to develop provably robust neural networks is to enforce Lipschitzness property [Tsuzuku et al., 2018]. Many papers have worked on designing Lipschitz layers [Li et al., 2019b, Trockman et al., 2021, Singla and Feizi, 2021] and activations [Anil et al., 2019, Singla et al., 2021a, Huang et al., 2021b].

Algorithm 1: Adversarial Training algorithm

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

Empirical defenses. Defenses against adversarial examples often have no theoretical guarantees and are based on training heuristics. The first defense that was proposed is *Adversarial Training* [Goodfellow et al., 2015b, Madry et al., 2018b]. This defense is a heuristic to minimize the adversarial risk. We describe the adversarial training defense in Algorithm 1 to training a classifier f_θ parametrized by θ. It consists in alternating minimization steps and attacks on the classifier to make it more robust. To our knowledge there exists no proof of convergence for this defense. Many other empirical defenses are variants of Adversarial Training, e.g. TRADES [Zhang et al., 2019a] or MART [Wang et al., 2019b]. For instance, TRADES aims at minimizing the following objective:

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

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

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

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

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

State-of-the-art in Image Classification To evaluate the performance of an attack of a classification algorithm, one needs to train and evaluate on datasets. In image classification evaluation, three datasets are mainly used:

- **MNIST [LeCun]:** A dataset of black and white low-quality images representing the 10 digits. The training set contains 50000 images and test set 10000 images. These images are of dimension $28 \times 28 \times 1$ (784 in total). This dataset is known to be easy ($> 99\%$ can be obtained using simple classifiers). In adversarial classification, the problem is also easy to be solved. Evaluation on MNIST is not sufficient to assess the performance of a classifier or even a defense against adversarial examples.

- **CIFAR10 and CIFAR100 [Krizhevsky and Hinton, 2009]:** Datasets of colored low-quality images representing the 10 labels and 100 labels for respectively CIFAR10 and CIFAR100. Each training set contains 50000 images and test set 10000 images. These images are of dimension $32 \times 32 \times 1$ (3072 in total). The current state-of-the-art on CIFAR10 in standard classification is $> 99\%$ of accuracy with most recent methods. 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%. It is worth noting 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 31st, 2021.

2.2.4 Theoretical knowledge in Adversarial classification

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

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

Generalization Bounds in Adversarial Learning. Similarly to the standard classification case, research has focused on computing uniform bounds for adversarial classification. These works are often inspired from generalizations of standard tools such as VC-dimension [Cullina et al., 2018] or Rademacher complexity [Yin et al., 2019, Khim and Loh, 2018, Awasthi et al., 2020] in the adversarial case. They exhibit generalization bounds that are highly dependent on the dimension of the input. Indeed the Rademacher complexity for classes adapted to the adversarial case adds a polynomial term in the dimension D of the input. However, for randomized classifiers, it is difficult to adapt PAC-Bayes bounds to the adversarial setting [Viallard et al., 2021]. Indeed, the proof schemes cannot be used in the adversarial setting. Moreover, there is still a lack of understanding of the bias-complexity tradeoff in the adversarial case [Wang et al., 2018].

Adversarial Bayes Risk. The adversarial Bayes risk has been studied only very recently by the community Bhagoji et al. [2019], Pydi and Jog [2021a], Trillos and Murray [2020] expressed the adversarial risk as an optimal transport problem for a suitable cost. Another approach was to study

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

the adversarial risk from a game theoretic perspective. We will explain in details these contributions in Section 3.1.1.

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

2.3 Game Theory in a Nutshell

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

2.3.1 Two-player zero-sum games

An important subclass of game theoretic problems are two-person zero-sum games. In such a game there are two players namely Player 1 and Player 2 with opposite objectives. When Player 1 plays an action x in some space $\mathcal{A}_1$ and Player 2 plays 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 feature is the order of play in the game: the strategies might be different if the player knows 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 the best responses for Player 2 to Player 1 strategy is a strategy that maximizes the utility: $\arg\max_\nu \mathbb{E}_{x\sim\mu, y\sim\nu}[u_1(x, y)]$. We denote this set $BR_2(\mu)$. Game theory aims at studying and computing the nature of strategies in response to other players strategies.

2.3.2 Equilibria in two-player zero-sum games

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

Nash Equilibria. In a Nash equilibrium, each player is assumed to know the equilibrium strategies of the 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 a Nash equilibrium is not always guaranteed. Formally, a Nash equilibrium is a tuple of actions $(x^\star, y^\star)$ for Players 1 and 2 such that for all other actions x for Player 1 and y for Player 2 we have:

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

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

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

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

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

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

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

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

2.3.3 Strong Duality Theorems

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.4 Optimal Transport concepts

Optimal Transport have gained interest in Machine Learning applications during the past years. Indeed, Optimal Transport has the ability to model many problems, e.g. Generative Adversarial Networks [Arjovsky et al., 2017], or Adversarial Learning [Sinha et al., 2017, Pydi and Jog, 2021a, Bhagoji et al., 2019]. In particular, it will be a central tool in this thesis with the notion of distributionally robust optimization introduced in Section 3.1.2. The computation methods for optimal transport problems have also been considerably improved recently. Originally introduced by Monge, this Optimal Transport was a problem where the aim was to move some quantity x to some places y while minimizing the total cost of transport. Let $\mathcal{Z}$ be a Polish space. Let $\mathbb{P}$ and $\mathbb{Q}$ be two Borel probability distributions over $\mathcal{Z}$ and $c : \mathcal{Z}^2 \to \bar{\mathbb{R}}_+$ be a non-negative function. Formally, the problem was posed as follows:

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

where T is a measurable mapping and $T_\sharp\mathbb{P}$ defines the pushforward measure of $\mathbb{P}$ by the function T:

$$T_\sharp\mathbb{P}(A) = \mathbb{P}\big[T^{-1}(A)\big]$$

for all measurable sets A.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Kantorivch Duality. A fundamental theorem in Optimal Transportation is the Kantorovich duality theorem which writes as follows.

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

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

where for all $x, y \in \mathcal{Z}$, $f \oplus g(x, y) := f(x) + g(y)$ and $\mathcal{C}(\mathcal{Z})$ is the set of continuous functions over $\mathcal{Z}$.

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

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

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

where for all $x, y \in \mathcal{Z}$, $f \oplus g(x, y) := f(x) + g(y)$ *and* $\mathcal{C}(\mathcal{Z})$ *is the set of continuous functions over* $\mathcal{Z}$.

3 Related Work

Contents

3.1 A Game Theoretic Approach to Adversarial Classification

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

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

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

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

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

However, this setting is limited due to the convexity assumptions. As we will see in Chapter 5, one can prove that no convex loss can be a good surrogate for the $0/1$ loss in the adversarial setting.

The goal of the paper is to build a framework to design new zero-shot black-box adversarial attacks from generative attackers. Such an attack is called a *No Box attack*.

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

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

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

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

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

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

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

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

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

3.1.1 Adversarial Risk Minimization and Optimal Transport

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

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

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

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

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

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

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

3.1.2 Distributionally Robust Optimization

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

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

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

$$\min_{\theta\in\Theta} \sup_{\mathbb{Q}\in\mathcal{M}_1^+(\mathcal{Z}) \mid D(\mathbb{P},\mathbb{Q})\leq\varepsilon} \mathbb{E}_{z\sim\mathbb{Q}}[f(z)]$$

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

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

$$\min_{\theta\in\Theta} \sup_{\mathbb{Q}\in\mathcal{M}_1^+(\mathcal{Z}) \mid W_c(\mathbb{P},\mathbb{Q})\leq\varepsilon} \mathbb{E}_{z\sim\mathbb{Q}}[f(z)]$$

The Wasserstein balls writes as

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

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

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

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

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

$$\sup_{\mathbb{Q} \in \mathcal{M}_1^+(\mathcal{Z}) \mid W_c(\mathbb{P}, \mathbb{Q}) \leq \varepsilon} \mathbb{E}_{z \sim \mathbb{Q}}[f(z)] = \inf_{\lambda \geq 0} \mathbb{E}_{z \sim \mathbb{P}} \left[\sup_{z' \in \mathcal{Z}} f(z') - \lambda c(z, z') \right] + \lambda \varepsilon$$

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

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

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

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

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

Proof. Let us define:

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

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

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

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

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

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

We have then

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

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

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

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

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

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

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

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

3.2 Surrogate losses in the Adversarial Setting

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

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

However, the adversarial version of these surrogate losses needs not to have the same consistency properties with respect to the adversarial $0/1$ loss. In fact, most existing results in the standard framework rely on a reduction of the global consistency problem to a local point-wise problem, called *calibration*. However, the same approach is not feasible in the adversarial setting, because the new losses are by nature non-point-wise. Then the optimum for a given input may depend on yet a whole other set of inputs [Awasthi et al., 2021a,c]. Studying the concepts of calibration and consistency in the adversarial setting remains an open and understudied problem. Furthermore, this is a complex and technical area of research, that requires a rigorous analysis, since small tweaks in definitions can quickly make results meaningless or inaccurate. This difficulty is illustrated in the literature, where articles published in high profile conferences tend to contradict or refute each other Bao et al. [2020], Awasthi et al. [2021a,c].

Setting. In this section, let us consider a classification task with input space $\mathcal{X}$ and output space $\mathcal{Y} = \{-1, +1\}$. Let $(\mathcal{X}, d)$ be a proper Polish (i.e. completely separable) metric space representing the inputs space. For all $x \in \mathcal{X}$ and $\delta > 0$, we denote $B_\delta(x)$ the closed ball of radius δ and center x. We also assume that for all $x \in \mathcal{X}$ and $\delta > 0$, $B_\delta(x)$ contains at least two points[1]. Let

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

us also endow $\mathcal{Y}$ with the trivial metric $d'(y, y') = \mathbf{1}_{y \neq y'}$. Then the space $(\mathcal{X} \times \mathcal{Y}, d \oplus d')$ is a proper Polish space. For any Polish space $\mathcal{Z}$, we denote $\mathcal{M}_+^1(\mathcal{Z})$ the Polish space of Borel probability measures on $\mathcal{Z}$. We will denote $\mathcal{F}(\mathcal{Z})$ the space of real valued Borel measurable functions on $\mathcal{Z}$. Finally, we denote $\bar{\mathbb{R}} := \mathbb{R} \cup \{\infty, +\infty\}$.

3.2.1 Notions of Calibration and Consistency

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

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

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

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

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

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

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

In the literature [Zhang, 2004b, Bartlett et al., 2006, Steinwart, 2007], the notion of surrogate losses has been studied as a consistency problem. Formally, the notion of consistency is as follows.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Connection between calibration and consistency. Calibration is a necessary condition for consistency. In general, the converse is not true. However, in the specific context of standard classification with a well-defined 0/1-loss, the notions of consistency and calibration have been shown to be equivalent [Zhang, 2004b, Bartlett et al., 2006, Steinwart, 2007]. The next section discusses existing results on calibration and consistency in the standard classification setting.

3.2.2 Existing Results in the Standard Classification Setting

In binary classification, h is often defined as the sign of a real valued function $f \in \mathcal{F}(\mathcal{X})$. The loss usually used to characterize classification tasks corresponds to the accuracy of the classifier h. When h is defined as above, this loss is defined as follows.

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

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

with a convention for the sign, e.g. $sign(0) = 1$. We will denote $\mathcal{R}_{\mathbb{P}}(f) := \mathcal{R}_{L_{0/1},\mathbb{P}}(f)$, $\mathcal{R}^\star_{\mathbb{P}} := \mathcal{R}^\star_{L_{0/1},\mathbb{P}}$, $\mathcal{C}(x, \eta, f) := \mathcal{C}_{L0/1}(x, \eta, f)$ and $\mathcal{C}^\star(x, \eta) := \mathcal{C}^\star_{L_{0/1}}(x, \eta)$.

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

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

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

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

Without loss of generality, we will say that ϕ is a margin loss function, and we will denote $\mathcal{R}_\phi$ the risk associated with the margin loss ϕ and $\mathcal{C}_\phi$ the calibration function. Notably, it has been demonstrated in several previous works [Zhang, 2004b, Bartlett et al., 2006, Steinwart, 2007] that, for a margin loss ϕ, we always have $\mathcal{C}^\star_\phi(x, \eta) = \inf_{\alpha \in \mathbb{R}} \eta\phi(\alpha) + (1 - \eta)\phi(-\alpha)$. This is in particular one of the main observation allowing to show the following strong result about the connection between consistency and calibration.

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

1. *ϕ is calibrated with regard to $L_{0/1}$,*
2. *ϕ is uniformly calibrated w.r.t. $L_{0/1}$,*
3. *ϕ is consistent with regard to $L_{0/1}$.*

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

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

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

3.2.3 Calibration and Consistency in the Adversarial Setting.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Definition 16 ($\mathcal{H}$-calibration function)**.** *Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X})$. Let L be a loss function. The* optimal $\mathcal{H}$-calibration function *is defined as*

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

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

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

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

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

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

3.3 Robustness and Lipchitzness

In this section, we overview the deep link that exist between adversarial examples and Lipschitzness. Indeed, a Lipschitz function is a function that does not vary a lot when changing its input and a classifier is robust if a small perturbation does not change the prediction. Formally, we recall a classifier h is *certifiably robust at level* ε at input x with label y if there exists a property depending on h, x, y and ε that implies that for all x' such that $d(x, x') \leq \varepsilon$, $h(x') = y$. We first recall a property linking Lipschitzness to Robustness. Then, we present the existing methods for building Lipschitz Neural Networks.

3.3.1 Lipschitz Property of Neural Networks

The Lipschitz constant has seen a growing interest in the last few years in the field of deep learning [Virmaux and Scaman, 2018, Fazlyab et al., 2019, Combettes and Pesquet, 2020, Béthune et al., 2021]. 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 [Szegedy et al., 2014, Farnia et al., 2019, Tsuzuku et al., 2018], better training stability [Xiao et al., 2018, Trockman et al., 2021], 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:

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

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

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

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

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

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

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

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

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

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

$$f(x) = (A_L\sigma(A_{L-1}\ldots\sigma(A_1x + b_1)\ldots) + b_L)$$

Assuming that σ is 1-Lipschitz, we have:

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

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

$$\|A\|_2 = \max_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sqrt{\lambda_{max}(A^TA)}$$

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

Lipschitz Regularization of Neural Networks. Based on the insight that Lipschitz Neural Networks are more robust to adversarial attacks, researchers have developed several techniques to regularize and constrain the Lipschitz constant of neural networks by adding a regularization $\Omega(f)$ to the classification objective to encourage a smaller Lipschitz constant. However, the computation of the Lipschitz constant of neural networks has been shown to be NP-hard [Virmaux and Scaman, 2018]. Most methods therefore tackle the problem by reducing or constraining the Lipschitz constant at the layer level. For instance, the work of Cisse et al. [2017], Huang et al. [2020a] and Wang et al. [2020a] exploit the orthogonality of the weights matrices to build Lipschitz layers. Other approaches [Gouk et al., 2018, Jia et al., 2017, Sedghi et al., 2018, Singla et al., 2021b, Araujo et al., 2021] 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.

Algorithm 2: Spectral normalization algorithm

Require: **Matrix W, Nb. Iter.** n

Initialize u and v

$$\left.\begin{array}{l} v \leftarrow \mathbf{W}u/\|\mathbf{W}u\|_2 \\ u \leftarrow \mathbf{W}^\top v/\|\mathbf{W}^\top v\|_2 \\ h \leftarrow 2/(\sum_i (\mathbf{W}u \cdot v)_i)^2 \end{array}\right\} n \text{ iterations}$$

return h

3.3.2 Learning 1-Lipschitz layers

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

The first natural idea to learn 1-Lipschitz layers is to normalize the matrices in the forward pass of a Neural Networks : $A_i \leftarrow \frac{A_i}{\|A_i\|_2}$. This natural idea was exploited by Miyato et al. [2018]. A key difficulty is the computation of the spectral norm $\|A_i\|_2$. The authors proposed to use the power iteration method to compute the spectral norm (see Algorithm 2). The number of iterations might be prohibitive, hence the authors proposed to use only one step in the training phase to make it faster. This method effectively approximates well the spectral norm of the last layer. However, this method presents some disadvantages. The spectral normalization has for side effect reducing the importance of smaller singular values. A consequence is the gradient vanishing that is very present in this structure.

Other approaches [Anil et al., 2019, Singla et al., 2021a, Huang et al., 2021b] proposed methods leveraging the properties of activation functions to constraint the Lipschitz constant of Neural Networks. These works are usually useful to help to improve the performance of linear orthogonal layers. We now present how methods that focus on learning orthogonal layers.

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

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

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

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

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

SOC layers use the exponential mapping of a skew symmetric matrix defined using the following Taylor expansion:

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

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

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

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

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

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

3.3.3 Residual Networks

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

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

where $F_t(x_t)$ is typically a two layer neural networks:

$$F_t(x_t) = W_{2,t}\sigma(W_{1,t}x_t)$$

for some weight matrices $W_{1,t}, W_{2,t}$ and activation function σ. The ResNet uses residual connection that have the effect of limiting gradient vanishing issues. Combined with batch normalization, the issue of gradient explosion can also be mitigated, hence opening the possibility to very deep and stable architecture.

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

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

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

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

It was argued by Haber et al. [2017] that when F_t does not depend on t or vary slowly with time[3], the stability can be characterized by the eigenvalues of the Jacobian matrix $\nabla_x F_t(x_t)$: the dynamical system is stable if the real part of the eigenvalues of the Jacobian remains negative throughout the propagation. This property however only relies on intuition and this condition might be difficult to verify in practice. In Chapter 6, in order to derive stability properties, we study gradient flows and convex potentials, which are subclasses of Residual networks.

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

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

4 Game Theory of Adversarial Examples

Contents

In this chapter, we answer **Question 1: "xxx"** by proving the existence of Mixed Nash equilibria in the adversarial example game when both the adversary and the classifier can use randomized strategies. First, we motivate in Section 4.1 the necessity for using randomized strategies both with the attacker and the classifier. Then, we extend the work of Pydi and Jog [2021a], by rigorously reformulating the adversarial risk as a linear optimization problem over distributions. In fact, we cast the adversarial risk minimization problem as a Distributionally Robust Optimization (DRO) [Blanchet and Murthy, 2019] problem for a well suited cost function. This formulation naturally leads us, in Section 4.2, to analyze adversarial risk minimization as a zero-sum game. We demonstrate that, in this game, the duality gap always equals 0, meaning that it always admits approximate mixed Nash equilibria.

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

4.1 The Adversarial Attack Problem

4.1.1 A Motivating Example

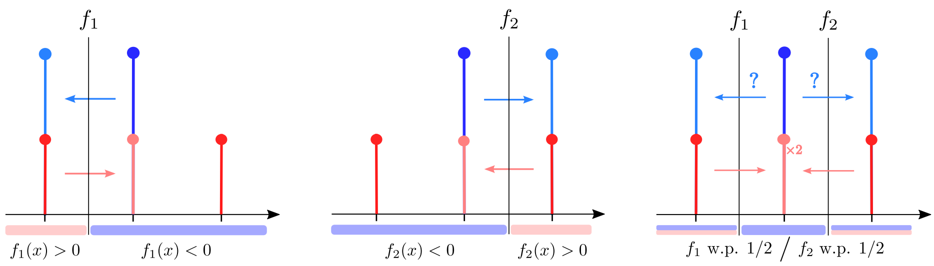


Figure 4.1: Motivating example: blue distribution represents label -1 and the red one, label $+1$. The height of columns represents their mass. The red and blue arrows represent the attack on the given classifier. On left: deterministic classifiers (f_1 on the left, f_2 in the middle) for whose, the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

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

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

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 ways, 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 $3/4$.

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

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

4.1.2 General setting

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

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

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

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

4.1.3 Measure Theoretic Lemmas

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

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 $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Let M be an upper bound on the loss L. For all $\theta \in \Theta$, $M - L(\theta, \cdot)$ is non negative and lower semi-continuous. Then by Fatou's Lemma :

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

$$\leq \liminf_{n\to\infty} \int M - L(\theta, (x_n, y_n))d\mu(\theta)$$

We then have: $\int M - L(\theta, \cdot)d\mu(\theta)$ 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 the 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. Letus denote v this function. Let $(v_n)_n$ be a non-decreasing sequence of continuous bounded functions such that $v_n \to v$. Let $(\mathbb{Q}_k)_k$ converge weakly towards $\mathbb{Q}$. Then by monotone convergence theorem:

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

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. □

4.1.4 Adversarial Risk Minimization

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

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

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

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

[2] For the well-posedness, see Lemma **??**.
[3] This risk is also well posed (see Lemma **??**).

For instance, for the $0/1$ loss, the inner maximization problem, consists in maximizing the probability of misclassification for a given pair (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

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

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

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

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

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

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

where essinf denotes the essential infimum. □

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

4.1.5 Distributional Formulation of the Adversarial Risk

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

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

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

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

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

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

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

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

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

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

Proposition 5. *Let $\mathbb{P}$ be a Borel probability distribution on $\mathcal{X} \times \mathcal{Y}$ and $\varepsilon > 0$ and $\eta \ge 0$, 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, $d(x,x') \le \varepsilon$, $y = y'$ γ-almost surely, and $\Pi_{1\sharp}\gamma = \mathbb{P}$, and $\Pi_{2\sharp}\gamma = \mathbb{Q}$. Then $\int c_\varepsilon d\gamma = 0 \le \eta$. Then, $W_{c_\varepsilon}(\mathbb{P},\mathbb{Q}) \le \eta$, 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 $\int c_\varepsilon d\gamma \le \eta$. Since $c_\varepsilon((x,x'),(y,y')) = +\infty$ when $d(x,x') > \varepsilon$ and $y \ne y'$, we deduce that, $d(x,x') \le \varepsilon$ 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 immediate from the relation with the Wasserstein uncertainty set.

Let us show first that $\mathcal{A}_\varepsilon(\mathbb{P})$ is relatively compact for the weak topology. To do so we will show that $\mathcal{A}_\varepsilon(\mathbb{P})$ is tight and apply Prokhorov 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 $\mathbb{P}(K_\delta) \ge 1 - \delta$. Let $\tilde{K}_\delta := \{(x',y') \mid \exists (x,y) \in K_\delta,\ d(x',x) \le \varepsilon,\ y = y'\}$. Recalling that $(\mathcal{X}, d)$ is proper (i.e. the closed balls are compact), so $\tilde{K}_\delta$ is compact. Moreover for $\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})$, $\mathbb{Q}(\tilde{K}_\delta) \ge \mathbb{P}(K_\delta) \ge 1 - \delta$. And then, Prokhorov's theorem holds, and $\mathcal{A}_\varepsilon(\mathbb{P})$ is relatively compact for the 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 $d(x,x') \le \varepsilon$ and $y = y'$ γ_n-almost surely and $\Pi_{1\sharp}\gamma_n = \mathbb{P}$, $\Pi_{2\sharp}\gamma_n = \mathbb{Q}_n$. $\{\mathbb{Q}_n, n \ge 0\}$ 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 $d(x,x') \le \varepsilon$ 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. $\square$

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

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

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

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

Proof. Let $\mu \in \mathcal{M}_+^1(\Theta)$. Let us define $\tilde{f}$ as

$$\tilde{f} : ((x, y), (x', y')) \mapsto \mathbb{E}_{\theta\sim\mu}[L(\theta, (x, y))] - c_\varepsilon((x, y), (x', y')) \quad .$$

$\tilde{f}$ is upper-semi continuous, hence upper semi-analytic. Then, by upper semi continuity of $\mathbb{E}_{\theta\sim\mu}[L(\theta, \cdot)]$ 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 $\mathbb{E}_{\theta\sim\mu}[L(\theta, T(x, y))] = \sup_{(x',y'),\, d(x,x')\leq\varepsilon, y=y'} \mathbb{E}_{\theta\sim\mu}[L(\theta, (x, y))]$. Let $\mathbb{Q} = T_\sharp\mathbb{P}$, then $\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})$. And then

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

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

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

Then we deduce the expected result:

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

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

□

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

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

4.2 Nash Equilibria in the Adversarial Game

4.2.1 Adversarial Attacks as a Zero-Sum Game

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

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

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

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

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

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

4.2.2 Dual Formulation of the Game

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

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

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

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

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

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

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

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

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

4.2.3 Nash Equilibria for Randomized Strategies

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.3 Finding the Optimal Classifiers

4.3.1 An Entropic Regularization

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

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

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

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

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

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

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

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

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

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

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

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

$$S^{(\varepsilon)}_{(x,y)} := \left\{(x',y') \mid c_\varepsilon((x,y),(x',y')) = 0\right\},$$

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

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

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

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

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

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

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

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 $\mathbb{Q}^\star_{\alpha,i}$. Moreover, L is upper semi-continuous (u.s.c), thus $\mathbb{Q} \to \mathbb{E}_{\mathbb{Q},\mu}[L(\theta,(x,y))]$ is also u.s.c.[a]

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. [b], therefore, by considering the limit superior as n goes to infinity we obtain that

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

from which we deduce that $\mathbb{Q}^\star_{\alpha,i}$ 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 $\mathbb{Q}^\star_{\alpha_i^{(\ell)},i}$ and $\mathbb{Q}^\star_i$ the solutions of respectively:

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

and

$$
\max_{\mathbb{Q}_i \in \Gamma_{i,\varepsilon}} \mathbb{E}_{\mathbb{Q}_i,\mu}[L(\theta,(x,y))] \quad .
$$

Since $\Gamma_{i,\varepsilon}$ is tight, $(\mathbb{Q}^\star_{\alpha_i^{(\ell)},i})_{\ell\geq 0}$ is also tight and we can extract by Prokhorov's theorem a subsequence which converges towards $\mathbb{Q}^\star$. Moreover we have

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

from which follows that

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

Then by considering the limit superior we obtain that

$$
\limsup_{\ell\to+\infty} \mathbb{E}_{\mathbb{Q}^\star_{\alpha_i^{(\ell)},i},\mu}[L(\theta,(x,y))] = \mathbb{E}_{\mathbb{Q}^\star_i,\mu}[L(\theta,(x,y))]
$$

from which follows that

$$
\mathbb{E}_{\mathbb{Q}^\star_i,\mu}[L(\theta,(x,y))] \leq \mathbb{E}_{\mathbb{Q}^\star,\mu}[L(\theta,(x,y))]
$$

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

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

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

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

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

Finally, we end up with the following problem:

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

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

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

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

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

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

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

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

where $C_i := \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\theta\in\Theta}\sum_{j=1}^m \sigma_j L(\theta, u_j^{(i)})\right]$ *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 $\mu \in \mathcal{M}_1^+(\Theta)$,

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

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

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

Since $\widehat{\mathcal{R}}^{\mathbf{m}}_{\varepsilon,\boldsymbol{\alpha}} \leq \widehat{\mathcal{R}}^{\mathbf{m}}_{\varepsilon,\boldsymbol{\alpha}}(\mu_n)$, we remark that

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

and by considering the limit, we obtain that

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

Similarly we have that

$$\begin{aligned}
\widehat{\mathcal{R}}^{\star}_{\varepsilon,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}^{\mathbf{m}}_{\varepsilon,\boldsymbol{\alpha}} &\leq \widehat{\mathcal{R}}^{\star}_{\varepsilon,\boldsymbol{\alpha}} - \widehat{\mathcal{R}}_{\varepsilon,\boldsymbol{\alpha}}(\mu_n^{(\boldsymbol{m})}) \\
&\quad + \widehat{\mathcal{R}}_{\varepsilon,\boldsymbol{\alpha}}(\mu_n^{(\boldsymbol{m})}) - \widehat{\mathcal{R}}^{\mathbf{m}}_{\varepsilon,\boldsymbol{\alpha}}(\mu_n^{(\boldsymbol{m})}) \\
&\quad + \widehat{\mathcal{R}}^{\mathbf{m}}_{\varepsilon,\boldsymbol{\alpha}}(\mu_n^{(\boldsymbol{m})}) - \widehat{\mathcal{R}}^{\mathbf{m}}_{\varepsilon,\boldsymbol{\alpha}}
\end{aligned}$$

from which follows that

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

Therefore we obtain that

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

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

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

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

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

and let us define

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

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

$$\begin{aligned}
|f(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(N)}) &- f(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(i-1)},\boldsymbol{z}^{(i)},\boldsymbol{u}^{(i+1)},\dots,\boldsymbol{u}^{(N)})| \\
&\leq \frac{\alpha}{N}\Bigg| \sup_{\mu \in \mathcal{M}_1^+(\Theta)} \left|\widehat{C}_i(\mu, \boldsymbol{u}^{(i)}) - C_i(\mu)\right| \\
&- \sup_{\mu \in \mathcal{M}_1^+(\Theta)} \left|\widehat{C}_i(\mu, \boldsymbol{z}^{(i)}) - C_i(\mu)\right|\Bigg| \\
&\leq \sup_{\mu \in \mathcal{M}_1^+(\Theta)} |\widehat{C}_i(\mu, \boldsymbol{u}^{(i)}) - \widehat{C}_i(\mu, \boldsymbol{z}^{(i)})|
\end{aligned}$$

$$= \frac{\alpha}{N}\Big|\frac{1}{m}\left[\exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,u_k^{(i)}))\right]}{\alpha}\right)-\exp\left(\frac{\mathbb{E}_{\theta\sim\mu}\left[L(\theta,z^{(i)}))\right]}{\alpha}\right)\right]\Big|$$
$$\leq \frac{2\alpha\exp(M/\alpha)}{Nm}$$

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

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

We have that

$$\mathbb{E}(f(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(N)}))=\frac{\alpha}{n}\sum_{i=1}^{N}\mathbb{E}\left(\sup_{\mu\in\mathcal{M}_1^+(\Theta)}\left|\widehat{C}_i(\mu,\boldsymbol{u}^{(i)})-C_i(\mu)\right|\right)\quad .$$

From the properties of Rademacher complexity (see Section 2.1.4), we have for every i :

$$\mathbb{E}\left(\sup_{\mu\in\mathcal{M}_1^+(\Theta)}\left|\widehat{C}_i(\mu)-C_i(\mu)\right|\right)\leq 2\mathbb{E}(\mathrm{Rad}(\mathcal{T}\circ\boldsymbol{u}^{(i)}))$$

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

$$\mathcal{H}\circ\boldsymbol{z}:=\left\{(f(z_1),\dots,f(z_q)),f\in\mathcal{F}\right\}\quad,$$
$$\mathrm{Rad}(\mathcal{T}\circ\boldsymbol{z}):=\frac{1}{q}\mathbb{E}_{\boldsymbol{\sigma}\sim\{\pm1\}}\left[\sup_{f\in\mathcal{H}}\sum_{i=1}^{q}\sigma_i f(z_i)\right],$$
$$\mathcal{T}:=\left\{u\to\exp\left(\frac{\mathbb{E}_{\theta\sim\mu}[L(\theta,u))]}{\alpha}\right),\ \mu\in\mathcal{M}_1^+(\Theta)\right\}.$$

Moreover, as mentioned in Section 2.1.4, $x\mapsto\exp(x/\alpha)$ is $\frac{\exp(M/\alpha)}{\alpha}$-Lipschitz on $(-\infty,M]$, we have

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

where

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

Let us now define

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

We observe that

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

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

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

Remarks also that

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

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 $1-\delta$

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

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

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

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

where

$$S^{(\varepsilon)}_{(x,y)} := \Big\{(x',y') \mid c_\varepsilon((x,y),(x',y')) = 0\Big\},$$

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

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

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 true 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 as for the proof of Proposition 9, let $(\mu_n^\varepsilon)_{n\geq 0}$ and $(\mu_n)_{n\geq 0}$ be two sequences such that

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

Remarks that

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

Then by considering the limit we obtain that

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

Similarly, we obtain that

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

from which follows that

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

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

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

Note that $(A^{(x_i,y_i)}_{\beta,\mu})^c$ denotes the complementary set of $A^{(x_i,y_i)}_{\beta,\mu}$. □

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

4.3.2 Proposed Algorithms

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

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

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 $\widehat{\mathcal{R}}_\varepsilon(\cdot)$ is a continuous convex function, hence $\min_{\boldsymbol{\lambda}\in\Delta_l} \mathcal{R}(\boldsymbol{\lambda})$ is attained for a certain $\boldsymbol{\lambda}^\star$. Then there exists a non-approximate Nash equilibrium $(\boldsymbol{\lambda}^\star, \mathbb{Q}^\star)$ in the adversarial game when Θ is finite. Here, we present two algorithms to learn the optimal mixture of the adversarial risk minimization problem.

Algorithm 3: Oracle-based Algorithm

$\boldsymbol{\lambda}_0 = \frac{\mathbf{1}_l}{l}; T; \eta = \frac{2}{M\sqrt{lT}}$

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

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

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

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

end

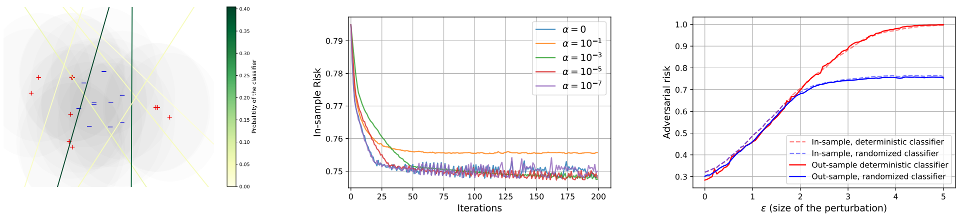


Figure 4.2: On left, 40 data samples with their set of possible attacks represented in shadow and the optimal randomized classifier, with a color gradient representing the probability of the classifier. In the middle, convergence of the oracle ($\alpha = 0$) and regularized algorithm for different values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

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

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

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

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

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

Proposition 11. *Let $\Theta = (\theta_1, \dots, \theta_l)$, $L : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to [0, \infty)$ be a loss satisfying Assumption 1, M bed defined as in Assumption 1 and $T \geq 1$. Then, Algorithm 3 satisfies:*

$$\min_{t\in[\![T-1]\!]} \widehat{\mathcal{R}}_\varepsilon(\boldsymbol{\lambda}_t) - \widehat{\mathcal{R}}^\star_\varepsilon \leq 2\delta + \frac{2M\sqrt{l}}{\sqrt{T}}$$

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

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

is a subgradient of $\boldsymbol{\lambda} \to \mathcal{R}(\boldsymbol{\lambda})$. In particular for every $\boldsymbol{\lambda}^\star$ optimal classifier:

$$\langle \boldsymbol{g}_t, \boldsymbol{\lambda}^\star - \boldsymbol{\lambda}_{t-1}\rangle \leq \mathcal{R}_\varepsilon(\boldsymbol{\lambda}^\star) - \mathcal{R}_\varepsilon(\boldsymbol{\lambda}_{t-1}) \quad .$$

Moreover, we also have

$$\begin{aligned}
|\langle \boldsymbol{g}_t^\star - \boldsymbol{g}_t, \boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^\star\rangle| &\leq \|\boldsymbol{g}_t^\star - \boldsymbol{g}_t\|_\infty \|\boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^\star\|_1 \\
&\leq \delta(\|\lambda_{t-1}\|_1 + \|\lambda^\star\|_1) \\
&\leq 2\delta \quad .
\end{aligned}$$

We also have that $\|\boldsymbol{g}_t\|_2 \leq \sqrt{l}\delta$. Let $\eta \geq 0$ be the learning rate. Then we have for all $t \geq 1$:

$$\begin{aligned}
\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 &\leq \|\boldsymbol{\lambda}_{t-1} - \eta\boldsymbol{g}_t - \boldsymbol{\lambda}^\star\|^2 \\
&= \|\boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^\star\|^2 - 2\eta\langle \boldsymbol{g}_t, \boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^\star\rangle + \eta^2\|\boldsymbol{g}_t\|_2^2 \\
&\leq \|\boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^\star\|^2 - 2\eta\langle \boldsymbol{g}_t^\star, \boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^\star\rangle \\
&+ 2\eta\langle \boldsymbol{g}_t^\star - \boldsymbol{g}_t, \boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^\star\rangle + \eta^2 M^2 l \\
&\leq \|\boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}^\star\|^2 - 2\eta(\mathcal{R}_\varepsilon(\boldsymbol{\lambda}_{t-1}) - \mathcal{R}_\varepsilon(\boldsymbol{\lambda}^\star)) + 4\eta\delta + \eta^2 M^2 l
\end{aligned}$$

We then deduce by summing:

$$2\eta \sum_{t=0}^{T-1} \mathcal{R}_\varepsilon(\boldsymbol{\lambda}_t) - \mathcal{R}_\varepsilon(\boldsymbol{\lambda}^\star) \leq 4\delta\eta T + \|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^\star\|^2 + \eta^2 M^2 l T$$

Then we have:

$$\min_{t \in [\![T-1]\!]} \mathcal{R}_\varepsilon(\boldsymbol{\lambda}_t) - \mathcal{R}_\varepsilon(\boldsymbol{\lambda}^\star) \leq 2\delta + \frac{4}{\eta T} + M^2 l \eta$$

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

$$\min_{t \in [\![T-1]\!]} \mathcal{R}_\varepsilon(\boldsymbol{\lambda}_t) - \mathcal{R}_\varepsilon(\boldsymbol{\lambda}^\star) \leq 2\delta + \frac{2M\sqrt{l}}{\sqrt{T}}$$

□

.

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. The entropic regularized algorithm is made to find an approximate the solution and do not require access to an oracle.

4.3.3 A General Heuristic Algorithm

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

4.4 Experiments

4.4.1 Synthetic Dataset

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

Algorithm 4: Adversarial Training for Mixtures

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

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

Finally, to demonstrate the improvement randomized techniques offer against deterministic defenses, we plot in Figure 4.2 (right) the minimum adversarial risk for both randomized and deterministic classifiers w.r.t. ε. The adversarial risk is strictly better for randomized classifier whenever the adversarial budget ε is bigger than 2. This illustration corroborates our analysis of Theorem 9, and motivates an in-depth study of a more challenging framework, namely image classification with neural networks.

4.4.2 CIFAR Datasets

Experimental Setup. We now implement our heuristic algorithm (Alg. 4) on CIFAR-10 and CIFAR-100 datasets for both Adversarial Traning [Madry et al., 2018b] and TRADES [Zhang et al., 2019a] loss. To evaluate the performance of Algorithm 4, we trained from 1 to 4 ResNet18 [He et al., 2016b] models on 200 epochs per model[4]. 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 adaptative version of PGD for mixtures of classifiers with 10 steps. Note that for one single model, Algorithm 4 exactly corresponds to adversarial training [Madry et al., 2018b] 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.

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

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 (APGD) 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.

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:

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

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

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

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

4.4.3 Effect of the Regularization

In this subsection, we experimentally investigate the effect of the regularization. In Figure 4.4, we notice that the regularization has the effect of stabilizing, reducing the variance and improving the level of the robust accuracy for adversarial training of mixtures (Algorithm 4). The standard accuracy curves are very similar in both cases.

Results. The results are presented in Figure 4.3. We remark our algorithm outperforms the standard adversarial training procedure in all the cases by more 1% on CIFAR-10 and CIFAR-100, without additional loss of standard accuracy as it is shown in the left figures. On TRADES, the gain is even more important by more than 2% in robust accuracy. Moreover, it seems that 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 costly, and it is difficult to draw precise conclusions. Further, hyperparameter tuning [Gowal et al., 2020a] such as architecture, unlabeled data [Carmon et al., 2019b] or activation function may still improve the results.

4.4.4 Additional Experiments on WideResNet28x10

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

4.4.5 Overfitting in Adversarial Robustness

We further investigate the overfitting of our heuristic algorithm. We plotted in Figure 4.6 the robust accuracy on ResNet18 with 1 to 5 models. The most robust mixture of 5 models against PGD with 20 iterations occurs 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 AutoPGD 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 future works.

4.5 Discussions and Open Questions

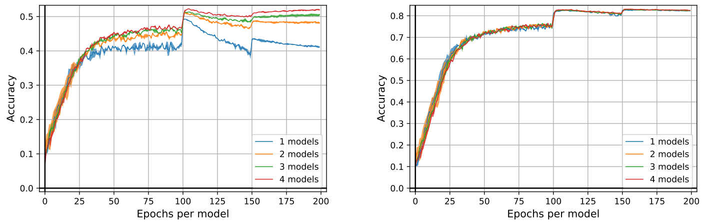
On the need of Randomization. While we give a concrete example where randomization of classifiers is needed to be optimal in Section 4.1.1, [Pydi and Jog, 2021b] show there is no duality gap when the classifier is allowed to play a deterministic measurable classifier. In other words, randomization would not be useful for this game. We conjecture, as the hypothesis class Θ grows, the duality gap decreases to 0. However, in finite samples cases, it is not realistic to optimize over the space of measurable functions. One may ask if we could find conditions on the space of classifiers and the distribution $\mathbb{P}$ such that randomization is required. Pinot et al. [2020] partially answered this question when the attacker is regularized, but the general case is still an open question.

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

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

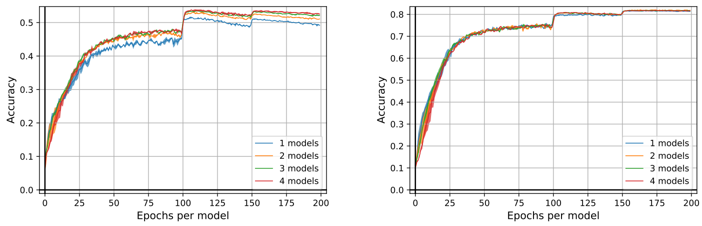
Adversarial Training, CIFAR-10 dataset results

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



TRADES, CIFAR-10 dataset results

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



Adversarial Training, CIFAR-100 dataset results

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

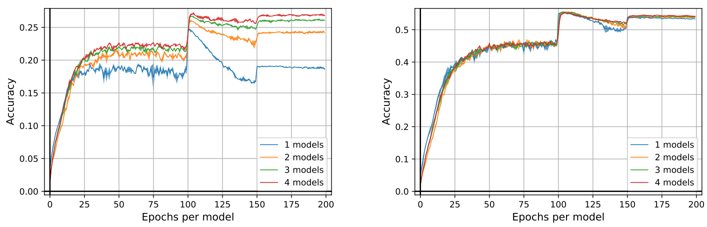


Figure 4.3: Upper plots: Adversarial Training, CIFAR-10 dataset results. Middle plots: TRADES, CIFAR-10 dataset results. Bottom plots: CIFAR-100 dataset results. On left: Comparison of our algorithm with a standard adversarial training (one model). We reported the results for the model with the best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

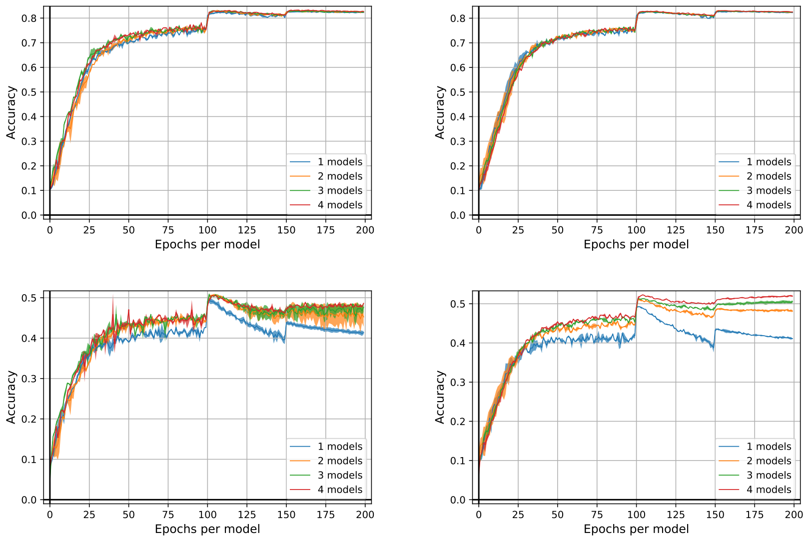


Figure 4.4: On top: Standard accuracies over epochs with respectively no regularization and regularization set to $\alpha = 0.001$. On bottom: Robust accuracies for the same parameters against PGD attack with 20 iterations and $\varepsilon = 0.03$.

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

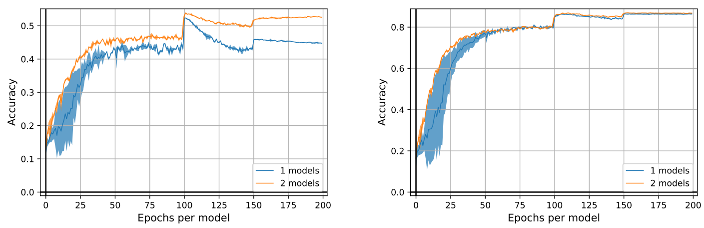
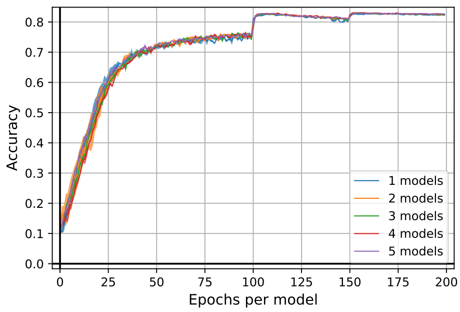


Figure 4.5: Comparison of our algorithm with a standard adversarial training (one model) on WideResNet28x10. We reported the results for the model with the best robust accuracy obtained over two independent runs because adversarial training might be unstable. Standard and Robust accuracy (respectively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



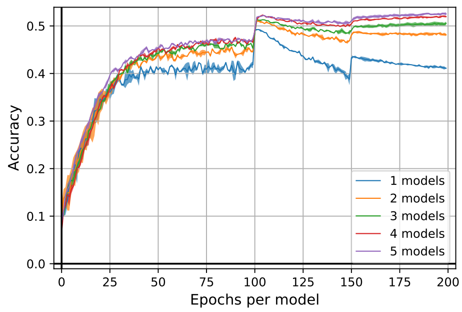


Figure 4.6: Standard and Robust accuracy (respectively on left and on right) on CIFAR-10 test images in function of the number of epochs per classifier with 1 to 5 ResNet18 models. The performed attack is PGD with 20 iterations and $\varepsilon = 8/255$. The best mixture for 5 models occurs at the end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

5 Calibration and Consistency in Presence of Adversarial Attacks

Contents

The objective of this chapter is to study the problem of calibration and consistency in presence of adversaries and answer **Question 2: xxx**. We study, in Section 5.1, the problem of calibration in the adversarial setting and provide both necessary and sufficient conditions for a loss to be calibrated in this setting. It is also worth noting that our results are easily extendable to $\mathcal{H}$-calibration (see Section 5.1.4). One on the main takeaways of our analysis is that no convex surrogate loss can be calibrated in the adversarial setting. We however characterize a set of non-convex loss functions, namely *shifted odd functions* that solve the calibration problem in the adversarial setting. Finally, we focus on the problem of consistency in the adversarial setting in Section 5.2. Based on min-max arguments, we provide insights that might help paving the way to prove consistency of shifted odd functions in the adversarial setting. Specifically, we proved strong duality results for these losses and show tight links with the $0/1$-loss. From these insights, we are able to provide a close but weaker property to consistency.

Setting. Let us consider a classification task with input space $\mathcal{X}$ and output space $\mathcal{Y} = \{-1, +1\}$. Let $(\mathcal{X}, d)$ be a proper Polish (i.e. completely separable) metric space representing the inputs space. For all $x \in \mathcal{X}$ and $\delta > 0$, we denote $B_\delta(x)$ the closed ball of radius δ and center x. We also assume that for all $x \in \mathcal{X}$ and $\delta > 0$, $B_\delta(x)$ contains at least two points[1]. Let us also endow $\mathcal{Y}$ with the trivial metric $d'(y, y') = \mathbf{1}_{y \neq y'}$. Then the space $(\mathcal{X} \times \mathcal{Y}, d \oplus d')$ is a proper Polish space. For any Polish space $\mathcal{Z}$, we denote $\mathcal{M}_1^+(\mathcal{Z})$ the Polish space of Borel probability measures on $\mathcal{Z}$. We will denote $\mathcal{F}(\mathcal{Z})$ the space of real valued Borel measurable functions on $\mathcal{Z}$. Finally, we denote $\bar{\mathbb{R}} := \mathbb{R} \cup \{\infty, +\infty\}$. Moreover, we take back the definitions introduced in Section 3.2.

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

5.1 Solving Adversarial Calibration

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

5.1.1 Necessary and Sufficient Conditions for Calibration

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

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

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

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

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

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

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

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

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

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

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

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

We recall thanks to Proposition 2 that for every $u, v \in \mathcal{X}$,

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

Then in particular, for all $x' \in B_\varepsilon(x)$, we have:

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

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

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

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

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

$$\mathcal{C}_{\phi_\varepsilon}(x_n, \frac{1}{2}, f_n) - \mathcal{C}^\star_{\phi_\varepsilon}(x_n, \frac{1}{2}) \leq \frac{1}{n}$$

and there exists $u_n, v_n \in B_\varepsilon(x_n)$, such that

$$f_n(u_n) f_n(v_n) \leq 0$$

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

$$0 \leq \frac{1}{2}\phi(\alpha_n) + \frac{1}{2}\phi(-\alpha_n) - \inf_{u\in\mathbb{R}}\left[\frac{1}{2}\phi(u) + \frac{1}{2}\phi(u)\right] \leq \mathcal{C}_{\phi_\varepsilon}(x, \frac{1}{2}, f_n) - \mathcal{C}^\star_{\phi_\varepsilon}(x, \frac{1}{2}) \\ \leq \frac{1}{n}$$

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

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

so that we deduce:

$$\frac{1}{2}\max(\phi(\alpha_n), \phi(\beta_n)) + \frac{1}{2}\max(\phi(-\alpha_n), \phi(-\beta_n)) \longrightarrow \inf_{u\in\mathbb{R}}\left[\frac{1}{2}\phi(u) + \frac{1}{2}\phi(u)\right] \tag{5.1}$$

Since, for n sufficiently large, $\alpha_n < 0 < \beta_n$ and ϕ is decreasing and strictly decreasing in a neighbourhood of 0, we have that:

$$\max(\phi(\alpha_n), \phi(\beta_n)) = \phi(\alpha_n)$$

and

$$\max(\phi(-\alpha_n), \phi(-\beta_n)) = \phi(-\beta_n)$$

. Moreover, there exists $\lambda > 0$ such that for n sufficiently large $\phi(\alpha_n) - \phi(\beta_n) \geq \lambda$. Then for n sufficiently large:

$$\begin{aligned} \frac{1}{2}\max(\phi(\alpha_n), \phi(\beta_n)) + \frac{1}{2}\max(\phi(-\alpha_n), \phi(-\beta_n)) \\ &= \frac{1}{2}\phi(\alpha_n) + \frac{1}{2}\phi(-\beta_n) \\ &= \frac{1}{2}(\phi(\alpha_n) - \phi(\beta_n)) + \frac{1}{2}\phi(-\beta_n) + \frac{1}{2} + \phi(\beta_n) \\ &\geq \frac{1}{2}\lambda + \inf_{u\in\mathbb{R}}\left[\frac{1}{2}\phi(u) + \frac{1}{2}\phi(u)\right] \end{aligned}$$

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

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

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

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

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

Then ϕ is adversarially uniformly calibrated at level ε □

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

5.1.2 Negative results

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

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

A convex loss satisfies $\frac{1}{2}\phi(\alpha) + \frac{1}{2}\phi(-\alpha) \geq \phi(0)$, hence $0 \in \operatorname{argmin}_{\alpha\in\mathbb{R}} \phi(\alpha) + \phi(-\alpha)$. From Theorem 11, we deduce the result. Then, ϕ is not adversarially calibrated at level ε. This result seems counter-intuitive and highlights the difficulty of optimizing and understanding the adversarial risk. Since convex losses are not calibrated, one may hope to rely on famous non-convex losses such as sigmoid and ramp losses. But, unfortunately, such losses are not either calibrated.

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

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

5.1.3 Positive results

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

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

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

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

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

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

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

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

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

Consequently we have:

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

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

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

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

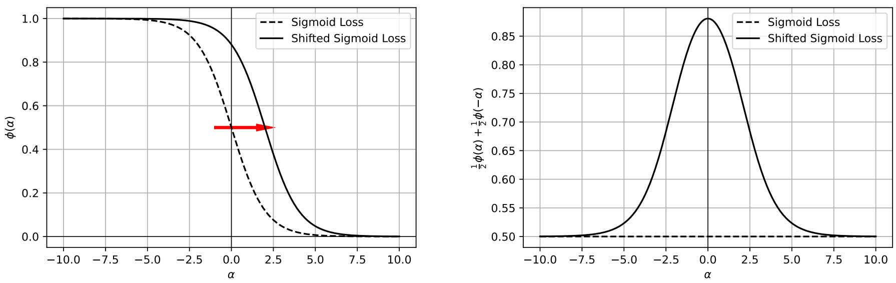


Figure 5.1: Illustration of the a calibrated loss in the adversarial setting. The sigmoid loss satisfy the hypothesis for ψ. Its shifted version is then calibrated for adversarial classification.

$$+\frac{1}{2}(\phi(f(x)-\tau)+\phi(-f(x)-\tau))$$

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

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

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

Without loss of generality, let $\eta>1/2$, then

$$t\mapsto(\eta-\frac{1}{2})(\phi(t-\tau)-\phi(-t-\tau))$$

is strictly decreasing and $\operatorname{argmin}_{t\in\bar{\mathbb{R}}} \frac{1}{2}(\phi(t-\tau)+\phi(-t-\tau))=\{-\infty,+\infty\}$, then we have

$$\operatorname*{argmin}_{t\in\bar{\mathbb{R}}} \lambda+(\eta-\frac{1}{2})(t-\tau)-\phi(-t-\tau))+\frac{1}{2}(\phi(t-\tau)+\phi(-t-\tau))=\{+\infty\}\quad.$$

By continuity of ϕ, we deduce that for $\delta>0$ sufficiently small:

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

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

Finally, we obtain that $\tilde{\phi}$ is calibrated for adversarial classification for every $\varepsilon>0$. $\square$

5.1.4 About $\mathcal{H}$-calibration

Our results naturally extend to $\mathcal{H}$-calibration. With mild assumptions on $\mathcal{H}$, it is possible to recover all the results made on calibration on $\mathcal{F}(\mathcal{X})$. First, it is worth noting that, if $\mathcal{H}$ contains all constant functions, then most results about calibration in the adversarial setting extend. Proposition 2 naturally extends to $\mathcal{H}$-calibration as long as $\mathcal{H}$ contains all constant functions.

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

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

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

The proof is exactly the same as for Proposition 2 since we used a constant function to prove the equality. Under the same assumptions, the notion of $\mathcal{H}$-calibration and uniform $\mathcal{H}$-calibration are equivalent in the standard setting.

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

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

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

Let $\eta \in [0,1]$, $x \in \mathcal{X}$ and $f \in \mathcal{H}$ such that $\mathcal{C}_\phi(x,\eta,f) - \mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) \le \delta$. Thanks to Proposition 5.1.4, $\mathcal{C}^\star_{\phi,\mathcal{H}}(x,\eta) = \mathcal{C}^\star_\phi(x,\eta)$, and $f \in \mathcal{F}(\mathcal{X})$, then $\mathcal{C}_\phi(x,\eta,f) - \mathcal{C}^\star_\phi(x,\eta) \le \delta$ and then:

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

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

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

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

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

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

Then ϕ is uniformly calibrated in standard classification. □

We can now obtain the necessary and sufficient conditions as follows. They are really similar to the adversarial calibration ones.

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

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

The proofs are the same as for the adversarial calibration setting. Note however that the assumptions on $\mathcal{H}$ are very weak: for instance, the set of linear classifiers

$$\mathcal{H} = \left\{ x \mapsto \langle w, x\rangle + b \mid w \in \mathbb{R}^d, b \in \mathbb{R} \right\}$$

satisfies the existence of $x \in \mathcal{X}$ and $(f_n)_n \in \mathcal{H}^\mathbb{N}$ such that $f_n(u) \to 0$ for all $u \in B_\varepsilon(x)$ and for all $n \in \mathbb{N}$, $\sup_{u \in B_\varepsilon(x)} f_n(u) > 0$ and $\inf_{u \in B_\varepsilon(x)} f_n(u) < 0$.

5.2 Towards Adversarial Consistency

We focus our study now on the problem of adversarial consistency. In a first part, taking inspiration from Long and Servedio [2013], Awasthi et al. [2021a], we study the ε-realisable case, i.e. the case where the adversarial risk at level ε equals zero. In a second part, we analyze the behavior of a candidate class of losses, namely the $0/1$-*like margin losses*.

5.2.1 The Realizable Case

The realizable case is important since there are no possible adversaries for the Bayes optimal classifier. Formally, this means that the adversarial risk equals 0, as stated in the following definition.

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

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

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

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

Lemma 4. *Let $\mathbb{P}$ be an ε-realisable distribution and ϕ be a calibrated margin loss in the standard setting. Then $\mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} = \inf_{\alpha\in\mathbb{R}} \phi(\alpha)$.*

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

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

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

$$p \times \mathbb{P}_1\left[\exists x' \in B_\varepsilon(x), f(x') < 0\right] \leq \epsilon'$$

and

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

Let us now define g as:

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

We have:

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

We have:

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

Similarly, we have:

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

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

We are now ready to prove the result of consistency in the realizable case.

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

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

Let now $f \in \mathcal{F}(\mathcal{X})$ be such that $\mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(f) \leq \mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} + \delta\epsilon$. Thanks to Lemma 4, we have:

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

Then by Markov inequality:

$$
\begin{aligned}
\mathbb{P}\left[\sup_{x' \in B_\varepsilon(x)} \phi(yf(x)) - \inf_{\alpha\in\mathbb{R}} \phi(\alpha) \geq \delta\right] & \leq \frac{\mathbb{E}_{\mathbb{P}}\left[\sup_{x' \in B_\varepsilon(x)} \phi(yf(x)) - \inf_{\alpha\in\mathbb{R}} \phi(\alpha)\right]}{\delta} \\
& \leq \epsilon
\end{aligned}
$$

So we have $\mathbb{P}[\forall x' \in B_\varepsilon(x), \phi(yf(x)) - \inf_{\alpha\in\mathbb{R}} \phi(\alpha) \leq \delta] \geq 1 - \epsilon$ and then

$$
\mathbb{P}\left[\forall x' \in B_\varepsilon(x), \mathbf{1}_{y\text{sign}(f(x)) \leq 0} = 0\right] \geq 1 - \epsilon \quad .
$$

Since $\mathbb{P}$ is ε-realisable, we have $\mathcal{R}^\star_{\varepsilon,\mathbb{P}} = 0$ and:

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

which concludes the proof. □

5.2.2 Towards the General Case

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

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

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

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

Based on min-max arguments, we provide below some results better characterizing $0/1$-like margin loss functions in the adversarial setting. Let us first recall the notions of *midpoint property* and *adversarial distributions set* that will be useful from now on as well as an important existing result from Pydi and Jog [2021b].

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

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

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

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

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

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

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

Note that in the original version of the theorem, Pydi and Jog [2021b] did not prove that the supremum is attained.

Proof. To prove that, note that for every Borel probability distribution $\mathbb{Q}$ over $\mathcal{X} \times \mathcal{Y}$,

$$\inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{\mathbb{Q}}(f) = (1-q) + \inf_{f \in \mathcal{C}(\mathcal{X}),\, 0 \leq f \leq 1} \int f d(q\mathbb{Q}_1 + (q-1)\mathbb{Q}_{-1})$$

where $q = \mathbb{Q}[y = 1]$ and $\mathbb{Q}_i = \mathbb{Q}[\cdot \mid y = i]$. When f is continuous and bounded, the function:

$$\mu \in \mathcal{M}(\mathcal{X}) \mapsto \int f d\mu$$

is continuous for the weak topology of measures, then:

$$\mu \in \mathcal{M}(\mathcal{X}) \mapsto \inf_{f \in \mathcal{C}(\mathcal{X}),\, 0 \leq f \leq 1} \int f d\mu$$

is upper semi continuous for the weak topology of measures, as it is the infinum of continuous functions. Then using the compacity of $\mathcal{A}_\varepsilon(\mathbb{P})$, we deduce that the supremum is attained. □

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

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

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

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

Lemma 5. *Let $\mathbb{Q}$ be a Borel probability distribution over $\mathcal{X} \times \mathcal{Y}$ and ϕ be a $0/1$-like shifted odd loss, then: $\mathcal{R}^\star_{\phi,\mathbb{Q}} = \mathcal{R}^\star_{\mathbb{Q}}$.*

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

$$\mathcal{R}^\star_{\phi,\mathbb{Q}} = \inf_{f \in \mathcal{F}(\mathcal{X})} \mathbb{E}_{(x,y)\sim\mathbb{Q}}[\phi(yf(x))]$$

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

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

We can now prove Theorem 14.

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

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

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

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

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

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

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

The last step is a consequence of Theorem 13. Then finally we get that $\mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}}=\mathcal{R}^\star_{\varepsilon,\mathbb{P}}$. □

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

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

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

Moreover the supremum is attained.

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

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

Proof. We have:

$$\begin{aligned}
\inf_{f \in \mathcal{F}(\mathcal{X})} \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}_{\phi,\mathbb{Q}}(f) &= \mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} = \mathcal{R}^\star_{\varepsilon,\mathbb{P}} && \text{by Theorem 14} \\
&= \inf_{f \in \mathcal{F}(\mathcal{X})} \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}_{\mathbb{Q}}(f) \\
&= \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{\mathbb{Q}}(f) \\
&= \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}^\star_{\mathbb{Q}}(f) = \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \mathcal{R}^\star_{\phi,\mathbb{Q}}(f) && \text{by Lemma 5} \\
&= \sup_{\mathbb{Q} \in \mathcal{A}_\varepsilon(\mathbb{P})} \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{\phi,\mathbb{Q}}(f)
\end{aligned}$$

$\mathbb{Q} \mapsto \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{\phi,\mathbb{Q}}(f) = \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{\mathbb{Q}}(f)$ is upper semi-continuous for the weak topology of measures. Moreover, $\mathcal{A}_\varepsilon(\mathbb{P})$ is compact for the weak topology of measures, then $\mathbb{Q} \mapsto \inf_{f \in \mathcal{F}(\mathcal{X})} \mathcal{R}_{\phi,\mathbb{Q}}(f)$ admits a maximum over $\mathcal{A}_\varepsilon(\mathbb{P})$. And $\mathbb{Q}$ is an optimal attack for the loss ϕ if and only if it is an optimal attack for the $0/1$ loss. □

A step towards consistency. From the previous results, we are able to prove a first result toward the demonstration of consistency. This result is much weaker than consistency result, but it guarantees that if a sequence minimizes the adversarial risk, then it minimizes the risk for optimal attacks, i.e. in a game where the attacker plays before the classifier

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

Proof. Let $(f_n)_{n\in\mathbb{N}}$ be a sequence of $\mathcal{F}(\mathcal{X})$ such that $\mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(f_n) \to \mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}}$. Let $\mathbb{Q}^\star$ be an optimal attack of level ε. From Corollary 3, we get that:

$$\mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}} = \mathcal{R}^\star_{\phi,\mathbb{Q}^\star} \quad .$$

Then we get

$$0 \leq \mathcal{R}_{\phi,\mathbb{Q}^\star}(f_n) - \mathcal{R}^\star_{\phi,\mathbb{Q}^\star} \leq \mathcal{R}_{\phi_\varepsilon,\mathbb{P}}(f_n) - \mathcal{R}^\star_{\phi_\varepsilon,\mathbb{P}}$$

from which we deduce that: $\mathcal{R}_{\phi,\mathbb{Q}^\star}(f_n) \to \mathcal{R}^\star_{\phi,\mathbb{Q}^\star}$. Since ϕ is consistent in the standard classification setting, we then have

$$\mathcal{R}_{\mathbb{Q}^\star}(f_n) \to \mathcal{R}^\star_{\mathbb{Q}^\star} \quad .$$

□

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

5.3 Discussions and Open Questions

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

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

Necessary and sufficient conditions for consistency. While we provided necessary and sufficient conditions for calibration in the adversarial setting, it is a difficult and open question to solve the problem of consistency. One may ask if the conditions we found for calibration might be necessary or sufficient for consistency. While there is an intuition that the notion of calibration is much weaker than consistency, we did not prove this. It would be challenging to find a counter-example for a loss that is calibrated but not consistent in the adversarial setting.

6 A Dynamical System Perspective for Lipschitz Neural Networks

Contents

In this chapter, we study the design of Lipschitz Layers under the light of the dynamical system interpretation of Neural Networks, hence answering **Question 3: xxx**. We recall briefly the continuous time interpretation of Residual Networks. 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:

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

Typically, F_t designates a two layer neural network. Note that this can be interpreted as the forward pass of a Neural Networks 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 [Trockman et al., 2021, Singla and Feizi, 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 certifiably-robust accuracy.

6.1 A Framework to design Lipschitz Layers

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

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

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

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

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

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

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

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

$$= 2\int_0^1 \left\langle x_t - z_t, \nabla_x F_{\theta_t}(z_t + s(x_t - z_t))(x_t - z_t)\right\rangle ds$$
$$= 2\int_0^1 \left\langle x_t - z_t, S(\nabla_x F_{\theta_t}(z_t + s(x_t - z_t)))(x_t - z_t)\right\rangle ds$$

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

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

Then by Gronwall Lemma, we have

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

which concludes the proof. □

The symmetric part plays even a more important role since one can show that a twice differentiable function whose Jacobian is always skew-symmetric is actually linear. Indeed, let $F := (F_1, \ldots, F_d) : \mathbb{R}^d \to \mathbb{R}^d$ 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 $F(x) = Ax$. Moreover, constraining $S(\nabla_x F_t(x))$ in the general case is technically difficult and a solution resorts to a more intuitive parametrization of F_t as the sum of two functions $F_{1,t}$ and $F_{2,t}$ whose Jacobian matrix are respectively symmetric and skew-symmetric. Thus, such a parametrization enforces $F_{2,t}$ to be linear and skew-symmetric. For the choice of $F_{1,t}$, we propose to rely on potential functions: a function $F_{1,t} : \mathbb{R}^d \to \mathbb{R}^d$ derives from a simpler family of scalar valued function in $\mathbb{R}^d$, called the *potential*, via the gradient operation. Moreover, since the Hessian of the potential is symmetric, the Jacobian for $F_{1,t}$ is then also symmetric. If we had the convex property to this potential, its Hessian would have positive eigenvalues. Therefore, we introduce the following corollary.

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

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

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

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

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

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

6.1.1 Discretized Flows

To study the discretization of the previous flow, let $t = 1, \ldots, T$ 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 $(f_t)_{t=1,\ldots,T}$ a family of convex differentiable functions on $\mathbb{R}^d$ and $(A_t)_{t=1,\ldots,T}$ a family of skew symmetric matrices. The most basic is method the explicit Euler scheme as defined by:

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

However, if $A_t \neq 0$, this discretized system might not satisfy $\|x_t - z_t\| \leq \|x_0 - z_0\|$. Indeed, consider the simple example where $f_t = 0$. We then have:

$$\|x_{t+1} - z_{t+1}\|^2 - \|x_t - z_t\|^2 = \|A_t(x_t - z_t)\|^2.$$

Thus explicit Euler scheme cannot guarantee Lipschitzness when $A_t \neq 0$. To overcome this difficulty, the discretization step can be split into two parts, one for $\nabla_x f_t$ and one for A_t:

$$\begin{cases} x_{t+\frac{1}{2}} &= \textsc{step}1(x_t, \nabla_x f_t) \\ x_{t+1} &= \textsc{step}2(x_{t+\frac{1}{2}}, A_t) \end{cases}$$

This type of discretization scheme can be found for instance from Proximal Gradient methods where one step is explicit and the other is implicit. Then, we dissociate the Lipschitzness study of both terms of the flow.

6.1.2 Discretization scheme for $\nabla_x f_t$

To apply the explicit Euler scheme to $\nabla_x f_t$, an additional smoothness property on the potential functions is required to generalize the Lipschitzness guarantee to the discretized flows. Recall that a function f is said to be *L-smooth* if it is differentiable and if $x \mapsto \nabla_x f(x)$ is L-Lipschitz.

Proposition 16. *Let $t \in \{1, \cdots, T\}$. Let us assume that f_t is M_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 $0 \leq h_t \leq \frac{2}{M_t}$, then

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

Proof. With $c_t = \|x_t - z_t\|_2^2$, we can write:

$$\begin{aligned} c_{t+\frac{1}{2}} - c_t = &- 2h_t\big\langle x_t - z_t, \nabla_x F_{\theta_t}(x_t) - \nabla_x F_{\theta_t}(z_t)\big\rangle \\ &+ h_t^2\|\nabla_x F_{\theta_t}(z_t) - \nabla_x F_{\theta_t}(z_t)\|_2^2 \end{aligned}$$

This equality allows us to derive the equivalence between $c_{t+1} \leq c_t$ and:

$$\frac{h_t}{2}\|\nabla_x F_{\theta_t}(x_t) - \nabla_x F_{\theta_t}(z_t)\|_2^2 \leq \langle x_t - z_t, \nabla F_{\theta_t}(x_t) - \nabla F_{\theta_t}(z_t)\rangle$$

Moreover, assuming that F_{θ_t} being that:

$$\frac{1}{M_t}\|\nabla_x F_{\theta_t}(x_t) - \nabla_x F_{\theta_t}(z_t)\|_2^2 \leq \big\langle x_t - z_t, \nabla_x F_{\theta_t}(x_t) - \nabla_x F_{\theta_t}(z_t)\big\rangle$$

We can see with this last inequality that if we enforce $h_t \leq \frac{2}{M_t}$, we get $c_{t+\frac{1}{2}} \leq c_t$ which concludes the proof. □

In Section 6.2, we describe how to parametrize a neural network layer to implement such a discretization step by leveraging the recent work on Input Convex Neural Networks Amos et al. [2017].

Remark 7. *Another solution relies on the implicit Euler scheme: $x_{t+\frac{1}{2}} = x_t - \nabla_x f_t(x_{t+\frac{1}{2}})$. Let us remark that $x_{t+\frac{1}{2}}$ is uniquely defined as:*

$$x_{t+\frac{1}{2}} = \underset{x\in\mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2}\|x - x_t\|^2 + f_t(x)$$

We recognized here the proximal operator of f_t that is uniquely defined since f_t is convex. Moreover, we have for two trajectories x_t and z_t:

$$\begin{aligned} \|x_t - z_t\|_2^2 &= \|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}} + \nabla_x f_t(x_{t+\frac{1}{2}}) - \nabla_x f_t(z_{t+\frac{1}{2}})\|_2^2 \\ &= \|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}}\|^2 + 2\langle x_t - z_t, \nabla_x f_t(x_{t+\frac{1}{2}}) - \nabla_x f_t(z_{t+\frac{1}{2}})\rangle \\ &+ \|\nabla_x f_t(x_{t+\frac{1}{2}}) - \nabla_x f_t(z_{t+\frac{1}{2}})\|_2^2 \\ &\geq \|x_{t+\frac{1}{2}} - z_{t+\frac{1}{2}}\|^2 \end{aligned}$$

where the last inequality is deduced from the convexity of f_t. So, without any further assumption on f_t, the discretized implicit convex potential flow is 1-Lipschitz. Then, this strategy defines a 1-Lipschitz flow without further assumption on f_t than convexity. To compute such a layer, one *could compute the proximal operator which is a strongly convex-minimization optimization problem. However, This strategy is not computationally efficient and not scalable and preliminary experiments did not show competitive results while the training time is prohibitive. We leave this solution for future work.*

6.1.3 Discretization scheme for A_t

The second step of discretization involves the term with skew-symmetric matrices 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:

$$x_{t+1} = x_{t+\frac{1}{2}} + A_t \frac{x_{t+1} + x_{t+\frac{1}{2}}}{2}$$
$$\iff x_{t+1} = \left(I - \frac{A_t}{2}\right)^{-1} \left(I + \frac{A_t}{2}\right) x_{t+\frac{1}{2}}.$$

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 since A_t is linear. By taking the value at $s = \frac{1}{2}$, we obtain the following transformation:

$$x_{t+1} := u_{\frac{1}{2}} = e^{\frac{A}{2}} x_{t+\frac{1}{2}}.$$

This step is therefore clearly norm preserving but the matrix exponentiation is challenging and it requires efficient approximations. This trend was recently investigated under the name of Skew Orthogonal Convolution (SOC) Singla and Feizi [2021].

6.2 Parametrizing Convex Potentials Layers

As presented in the previous section, parametrizing the skew symmetric updates has been studied by Trockman et al. [2021], Singla and Feizi [2021]. Here, we focus on the parametrization of symmetric updates with the convex potentials proposed in Proposition 16. For that purpose, the Input Convex Neural Network (ICNN) [Amos et al., 2017] provide a relevant starting point that we will extend.

6.2.1 Gradient of ICNN

We use 1-layer ICNN [Amos et al., 2017] to define an efficient computation of Convex Potentials Flows. For any vectors $w_1, \dots w_k \in \mathbb{R}^d$, and bias terms $b_1, \dots, b_k \in \mathbb{R}$, and for ϕ a convex function, the potential F defined as:

$$F_{w,b} : x \in \mathbb{R}^d \mapsto \sum_{i=1}^{k} \phi(w_i^\top x + b_i)$$

defines a convex function in x as the composition of a linear and a convex function. Its gradient with respect to its input x is then:

$$x \mapsto \sum_{i=1}^{k} w_i \phi'(w_i^\top x + b_i) = \mathbf{W}^\top \phi'(\mathbf{W}x + \mathbf{b})$$

with $\mathbf{W} \in \mathbb{R}^{k \times d}$ and $\mathbf{b} \in \mathbb{R}^k$ are respectively the matrix and vector obtained by the concatenation of, respectively, $w_i^\top$ and b_i, and ϕ' is applied element-wise. Moreover, assuming ϕ' is M-Lipschitz, we have that $F_{w,b}$ is $M\|\mathbf{W}\|_2^2$-smooth. $\|\mathbf{W}\|_2$ denotes the spectral norm of $\mathbf{W}$. The reciprocal also holds: if $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-decreasing M-Lipschitz function, $\mathbf{W} \in \mathbb{R}^{k \times d}$ and $b \in \mathbb{R}^k$, there exists a convex $M\|\mathbf{W}\|_2^2$-smooth function $F_{w,b}$ such that

$$\nabla_x F_{w,b}(x) = \mathbf{W}^\top \sigma(\mathbf{W}x + \mathbf{b}),$$

where σ is applied element-wise. The next section shows how this property can be used to implement the building block and training of such layers.

6.2.2 Convex Potential layers

From the previous section, we derive the following *Convex Potential Layer*:

$$z = x - \frac{2}{\|\mathbf{W}\|_2^2} \mathbf{W}^\top \sigma(\mathbf{W}x + b)$$

Algorithm 5: Computation of a Convex Potential Layer

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

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

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

Written in a matrix form, this layer can be implemented with every linear operation $\mathbf{W}$. In the context of image classification, it is beneficial to use convolutions[1] instead of generic linear transforms represented by a dense matrix.

Remark 8. *When* $\mathbf{W} \in \mathbb{R}^{1\times d}$*,* $b = 0$ *and* $\sigma : x \mapsto$ *ReLU(x)* $= \max(x, 0)$*, 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 scaled or normalized 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 performed: an input x of size $c \times h \times w$ can be extended to some x' of size $c' \times h \times w$, where $c' > c$, which equals x on the c first channels and 0 on the $c' - c$ other channels.

Reducing dimensionsionality. Dimensionality reduction is another essential operation in neural networks. On one hand, its goal is to reduce the number of parameters and thus the amount of computation required to build the network. On the other hand it allows the model to progressively map the input space on the output dimension, which corresponds in many cases to the number of different labels K. In this context, several operations exist: pooling layers are used to extract information present in a region of the feature map generated by a convolution layer. One can adapt pooling layers (*e.g.* max and average) to make them 1-Lipschitz [Bartlett et al., 2017]. Finally, a simple method to reduce the dimension is the product with a non-square matrix. We simply implement it as the truncation of the output. This obviously maintains the Lipschitz constant.

6.2.3 Computing spectral norms

Our Convex Potential Layer, described in Equation 6.2.2, can be adapted to any kind of linear transformations (*i.e.* Dense or Convolutional) but requires the computation of the spectral norm

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

for these transformations. The exact computation of the spectral norm of a linear operator is computationally prohibitive, 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 [Jia et al., 2017, Singla et al., 2021b, Araujo et al., 2021]. While these methods are generally efficient, they are less relevant in 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 of the choice of the starting vector, where $\lambda_1 > \lambda_2$ are the two largest singular values of $\mathbf{W}$. While it can appear 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[2]. We describe with more details in Algorithm 5, the operations performed during a forward pass with a Convex Potential Layer.

However, for evaluation purpose, we need to compute the certified adversarial robustness, and this requires to ensure the convergence of the PM. Therefore, we perform 100 iterations for each layer[3] at inference time. Also note that at inference time, the computation of the spectral norm only needs to be performed once for each layer.

6.3 Experiments

To evaluate our new 1-Lipschitz Convex Potential Layers, we conducted 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 Section 6.3.1. By computing the certified and empirical adversarial accuracy of our networks on CIFAR10 and CIFAR100 classification tasks [Krizhevsky and Hinton, 2009], we show that our architecture is competitive with state-of-the-art methods (Sections 6.3.3). We also study the influence of some hyperparameters and demonstrate the stability and the scalability of our approach by training very deep neural networks up to 1000 layers without normalization tricks or gradient clipping.

6.3.1 Training and Architectural Details

We demonstrate the effectiveness of our approach on a classification task with CIFAR10 and CIFAR100 datasets [Krizhevsky and Hinton, 2009]. We use a similar training configuration to the

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

[3] 100 iterations of Power Method is sufficient to converge with a geometric rate.

#	S	M	L	XL
Conv. Layers	20	30	50	70
Channels	45	60	90	120
Lin. Layers	7	10	15	15
Lin. Features	2048	2048	4096	4096

Table 6.1: Architectures description for our Convex Potential Layers (CPL) neural networks with different capacities. We vary the number of Convolutional Convex Potential Layers, the number of Linear Convex Potential Layers, the number of channels in the convolutional layers and the width of fully connected layers. They will be reported respectively as CPL-S, CPL-M, CPL-L and CPL-XL.

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 evaluate four different variations of different hyperparameters as described in Table 6.1, respectively named CPL-S, CPL-M, CPL-L and CPL-XL, ranked according to the number of parameters they have. In all our experiments, we made 3 independent trainings to evaluate accurately the models. All reported results are the average of these 3 runs.

6.3.2 Concurrent Approaches

We compare our networks with SOC [Singla and Feizi, 2021] and Cayley Trockman et al. [2021] networks which are to our knowledge the best performing approaches for deterministic 1-Lipschitz Neural Networks. Since our layers are fundamentally different from these ones, we cannot compare with the same architectures. We reproduced SOC results for with 10 and 20 layers, that we call respectively SOC-10 and SOC-20 in the same training setting, i.e. normalized inputs, cross entropy loss, SGD optimizer with learning rate 0.1 and multi-step learning rate scheduler. For Cayley layers networks, we reproduced their best reported model, i.e. KWLarge with width factor of 3.

The work of Singla et al. [2021a] propose three methods to improve certifiable accuracies from SOC layers: a new HouseHolder activation function (HH), last layer normalization (LLN), and certificate regularization (CR). The code associated with this approach is not open-sourced yet, so we just reported the results from their paper in ours results (Tables 6.1 and 6.2) 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).

	Standard Accuracy	Provable Accuracy (ε)			Time per epoch (s)
		36/255	72/255	108/255	
CPL-S	75.6	62.3	46.9	32.2	21.9
CPL-M	76.8	63.3	47.5	32.5	40.0
CPL-L	77.7	63.9	48.1	32.9	93.4
CPL-XL	78.5	64.4	48.0	33.0	163
Cayley (KW3)	74.6	61.4	46.4	32.1	30.8
SOC-10	77.6	62.0	45.0	29.5	33.4
SOC-20	78.0	62.7	46.0	30.3	52.2
SOC+-10	76.2	62.6	47.7	34.2	N/A
SOC+-20	76.3	62.6	48.7	36.0	N/A

Table 6.1: 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.

6.3.3 Results

In this section, we present our results on adversarial robustness. We provide results on provable ℓ_2 robustness as well as empirical robustness on CIFAR10 and CIFAR100 datasets for all our models and the concurrent ones

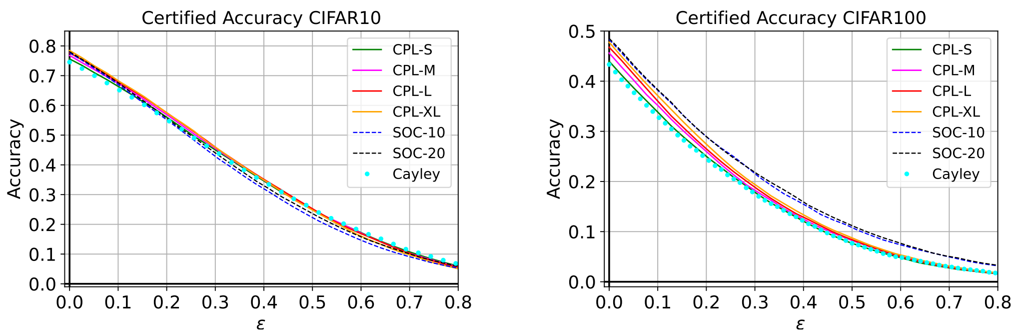


Figure 6.2: Certifiably robust accuracy w.r.t. the perturbation ε for our CPL networks and its concurrent approaches (SOC and Cayley models) on CIFAR10 and CIFAR100 datasets.

Certified Adversarial Robustness. Results on CIFAR10 and CIFAR100 dataset are reported respectively in Tables 6.1 and 6.2. We also plotted certified accuracy w.r.t. ε on Figure 6.2. On CIFAR10, our method outperforms the concurrent approaches in terms of standard and certi-

	Standard Accuracy	Provable Accuracy (ε)			Time per epoch (s)
		36/255	72/255	108/255	
CPL-S	44.0	29.9	19.1	11.0	22.4
CPL-M	45.6	31.1	19.3	11.3	40.7
CPL-L	46.7	31.8	20.1	11.7	93.8
CPL-XL	47.8	33.4	20.9	12.6	164
Cayley (KW3)	43.3	29.2	18.8	11.0	31.3
SOC-10	48.2	34.3	22.7	14.0	33.8
SOC-20	48.3	34.4	22.7	14.2	52.7
SOC+-10	47.1	34.5	23.5	15.7	N/A
SOC+-20	47.8	34.8	23.7	15.8	N/A

Table 6.2: 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.

fied 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.

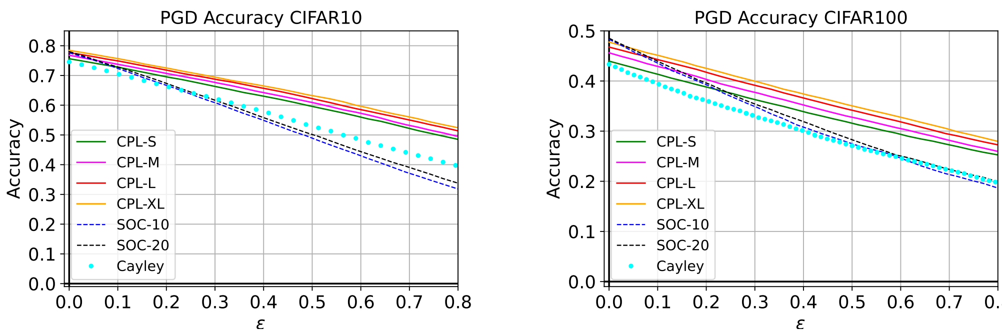


Figure 6.3: Accuracy against PGD attack with 10 iterations w.r.t. the perturbation ε for our CPL networks and its concurrent approaches on CIFAR10 and CIFAR100 datasets.

Empirical Adversarial Robustness. We also reported in Figure 6.3 the accuracy of all the models against PGD ℓ_2-attack [Kurakin et al., 2016, Madry et al., 2018b] for various levels of ϵ. We used 10 iterations for this attack. We remark here that our method 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 such 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 for empirical robustness tasks.

	Batch	**Standard Acc.**	**Provable Accuracy (ε)**			**T./epoch (s)**
			36/255	72/255	108/255	
CPL-S	64	76.5	62.9	47.3	32.0	48
	128	76.1	62.8	47.1	32.3	31
	256	75.6	62.3	46.9	32.2	22
CPL-M	64	77.4	63.6	47.4	32.1	77
	128	77.2	63.5	47.5	32.1	50
	256	76.8	63.2	47.4	32.4	40
CPL-L	64	78.4	64.2	47.8	32.2	162
	128	78.2	64.3	47.9	32.5	109
	256	77.6	63.9	48.1	32.7	93
CPL-XL	64	78.9	64.2	47.2	31.2	271
	128	78.9	64.2	47.5	31.8	198
	256	78.5	64.4	47.8	32.4	163

Table 6.3: Results on the CIFAR10 dataset on standard and provably certifiable accuracies for different values of perturbations ε on CPL (ours) models with various batch sizes. The average time per epoch in seconds is also reported in the last column. All the reported networks use Last Layer Normalization.

Effect of Batch Size in Training. In Tables 6.3 and 6.4, 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 even in some cases we may loose some points in robustness.

Effect of the Margin Parameter. In these experiments we varied the margin parameter in the margin loss in Figures 6.4 and 6.5. It clearly exhibits a tradeoff between standard and robust ac-

	Batch	Standard Acc.	Provable Acc. (ε)			T./epoch (s)
			36/255	72/255	108/255	
CPL-S	64	45.6	30.8	19.3	11.2	47
	128	44.9	30.7	19.2	11.0	31
	256	44.0	29.9	19.1	10.9	23
CPL-M	64	46.6	31.6	19.6	11.6	78
	128	46.3	31.1	19.7	11.5	55
	256	45.6	31.1	19.3	11.3	41
CPL-L	64	48.1	32.7	20.3	11.7	163
	128	47.4	32.3	20.0	11.8	116
	256	46.8	31.8	20.1	11.7	95
CPL-XL	64	49.0	33.7	21.1	12.0	293
	128	48.0	33.7	21.0	12.1	209
	256	47.8	33.4	20.9	12.6	164

Table 6.4: 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.

curacy. 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 obtain 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.

6.3.4 Training stability: scaling up to 1000 layers

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 [Miyato et al., 2018, Farnia et al., 2019] 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 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

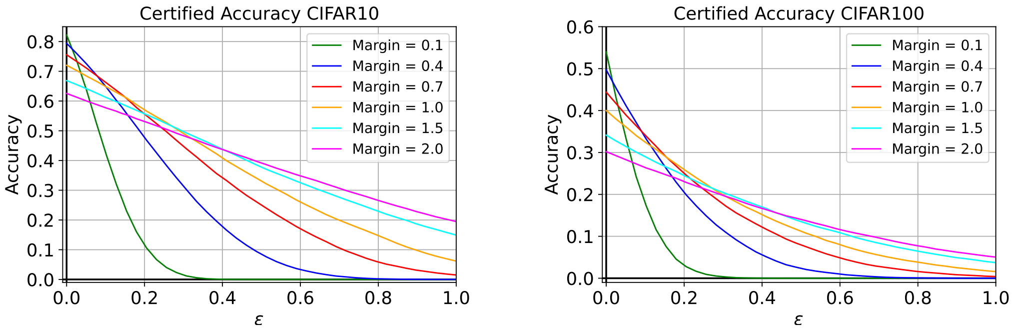


Figure 6.4: Certifiably robust accuracy w.r.t. the perturbation ε for our CPL-S network with different margin parameters on CIFAR10 and CIFAR100 datasets.

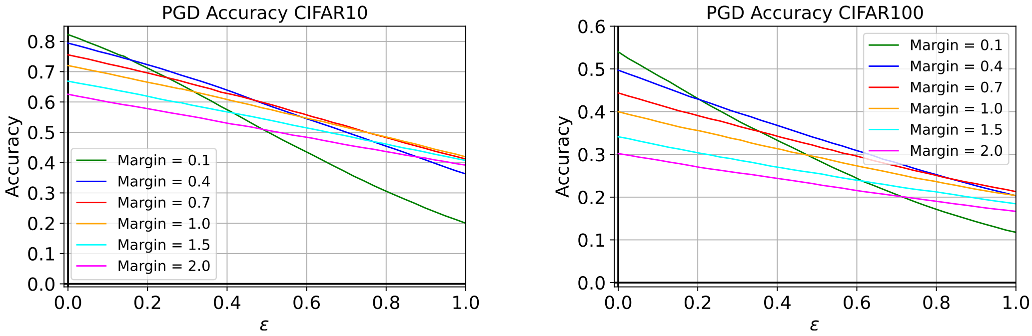


Figure 6.5: Certifiably robust accuracy w.r.t. the perturbation ε for our CPL-S network with different margin parameters on CIFAR10 and CIFAR100 datasets.

both worlds: limiting exploding and vanishing issues while maintaining scalability. To demonstrate the scalability of our approach, we experiment the ability to scale our architecture to very high depth (up to 1000 layers) without any additional normalization/regularization tricks, such as Dropout [Srivastava et al., 2014], Batch Normalization [Ioffe and Szegedy, 2015] or gradient clipping [Pascanu et al., 2013]. With the work done by Xiao et al. [2018], which leverage Dynamical Isometry and a Mean Field Theory to train a 10000 layers neural network, we believe, to the best of our knowledge, to be the second to perform such training. For the sake of computation efficiency, we limit this experiment to architecture with 30 feature maps. We report the accuracy in terms of epochs for our architecture in Figure 6.6 for a varying number of convolutional layers. It is worth noting that for the deepest networks, it may take a few epochs before the start of convergence. As Xiao et al. [2018], we remark there is no gain in using very deep architecture for this task.

6.3.5 Relaxing linear layers

Table 6.5 shows the result of the relaxed training of our CPL architecture, i.e. we fixed the step h_t in the discretized convex potential flow of Proposition 16. Increasing the constant h allows for

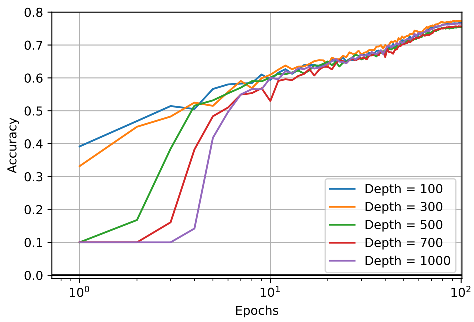


Figure 6.6: Standard test accuracy w.r.t. the number of epochs (log-scale) for various depths for our neural networks (100, 300, 500, 700, 1000).

	h = 1.0	**h = 0.1**	**h = 0.01**
Standard	85.10	82.23	78.53
PGD ($\varepsilon = 36/255$)	61.45	62.99	60.98

Table 6.5: Level of accuracy of CPL networks when the constraints on the step-size is relaxed. We fixed the step-size h to different values and measured standard and empirically robust accuracy. Here the CPL-M model is used.

an important improvement in the standard accuracy, but we loose in robust empirical accuracy. While computing the certified accuracy is not possible in this case due to the unknown value of the Lipschitz constant, we can still notice that the training of the network are still stable without normalization tricks, and offer a non-negligible level of robustness.

6.4 Discussions and Open questions

In this chapter, we presented a new generic method to build 1-Lipschitz layers. We leverage the continuous time dynamical system interpretation of Residual Networks and show that using convex potential flows naturally defines 1-Lipschitz neural networks. After proposing a parametrization based on Input Convex Neural Networks [Amos et al., 2017], we show that our models reach competitive results in classification and robustness in comparison which other existing 1-Lipschitz approaches. We also experimentally show that our layers provide scalable approaches without further regularization tricks to train very deep architectures.

Exploiting the ResNet architecture for devising flows has gained interest 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]. 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 has not been exploited.

Expressivity of discretized convex potential flows. Proposition 15 suggests to constraint the symmetric part of the Jacobian of F_t. We proposed to decompose F_t as a sum of potential gradient and a skew symmetric matrix. Finding other parametrizations is an open challenge. Our models may not express all 1-Lipschitz functions. Knowing which functions can be approximated by our CPL layers is difficult even in the linear case. Indeed, let us define $\mathcal{S}_1(\mathbb{R}^{d\times d})$ the space of real symmetric matrices with singular values bounded by 1. Let us also define $\mathcal{U}_1(\mathbb{R}^{d\times d})$ the space of real matrices with singular values bounded by 1 in absolute value. Let $\mathcal{P}(\mathbb{R}^{d\times d}) = \{A \in \mathbb{R}^{d\times d} | \exists n \in \mathbb{N}, S_1, \dots, S_n \in \mathcal{S}_1(\mathbb{R}^d \times d)$ s.t. $A = S_1 \dots S_n\}$. Then one can prove[4] that $\mathcal{P}(\mathbb{R}^{d\times d}) \neq \mathcal{U}_1(\mathbb{R}^{d\times d})$. Thus there exists $A \in \mathcal{U}_1(\mathbb{R}^{d\times d})$ such that for all matrices n, for all matrices $S_1, \dots, S_n \in \mathcal{S}_1(\mathbb{R}^{d\times d})$ such that $M \neq S_1, \dots, S_n$. Applied to the expressivity of discretized convex potential flows, the previous result means that there exists a 1-Lipschitz linear function that cannot be approximated as a discretized flow of any depth of convex linear 1-smooth potential flows as in Proposition 16. Indeed such a flow would write: $x \mapsto \prod_i (1 - 2S_i)x$ where S_i are symmetric matrices whose eigenvalues are in $[0, 1]$, in other words such transformations are exactly described by $x \mapsto Mx$ for some $M \in \mathcal{P}(\mathbb{R}^{d\times d})$. This is an important question that requires further investigation.

Going beyond ResNets One can also think of extending our work to other dynamical systems. Recent architectures such as Hamiltonian Networks [Greydanus et al., 2019] and Momentum Networks [Sander et al., 2021a] exhibit interesting properties and it is worth digging into these architectures to build Lipschitz layers. Finally, we hope to use similar approaches to build robust Recurrent Neural Networks [Sherstinsky, 2020] and Transformers [Vaswani et al., 2017]. For Transformers, Vuckovic et al. [2020], Sander et al. [2021b] has proposed a dynamical system interpretation of a flow on particles (i.e. the words in the initial sentence for NLP tasks). This can be seen as an interacting flow over a distribution. The question of robustness and Lipschitzness is way more technical since it implies Lipschitzness in the space of distributions. One could imagine to use Wasserstein Gradient flows [Ambrosio et al., 2005] as tools for deriving Lipschitz guarantees for Transformers.

[4] A proof and justification of this result can be found here.

7 Conclusion

Contents

7.1 Summary of the Thesis

In this thesis, we studied the problem of classification in presence of adversaries from different points of view for theoretical and practical purposes. We have tried to analyze the problem using both a high level and a more precise analysis. We summarize our findings as follows.

Summary of contributions

1. We provided a better understanding of the adversarial attacks problem by studying the nature of equilibria in this game and then, proved the existence of mixed Nash equilibria for very general settings. There is a hope this research direction will lead to principled results that can be used in practice for better defending against adversarial examples using randomized classifiers.

2. We studied and closed the problem of calibration in the adversarial binary-classification setting providing necessary and sufficient conditions. We paved the way to prove consistency results, and hope being able to conclude on consistency of shifted odd losses. It remains open to find necessary and sufficient conditions for consistency.

3. We derived a principled way based on dynamical system to build 1-Lipschitz layers. Interestingly, we recovered some existing methods from the literature, but we were also able to build new well-performing layers, namely the Convex Potential Layers. We hope this work would lead to study other possible dynamical systems and provide new provably robust neural networks.

Although this thesis proposed some solutions to the adversarial attacks problem, we also opened many questions that would require further investigation.

7.2 Open Questions

7.2.1 Optimizing the Adversarial Attacks Problem

The optimization of the adversarial attacks problem is still an open from multiple point of views. Recall that the adversarial risk minimization problem writes

$$\inf_{h\in\mathcal{H}} \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{x'\in\mathcal{X}\mid\, d(x,x')\leq\varepsilon} L(h(x),y)\right]$$

In classification, the end-objective is the accuracy, hence one need to optimize the $0/1$ loss. However, optimizing the $0/1$ loss is not computationally tractable. In the adversarial setting, the choice of a good surrogate loss L to the $0/1$ loss is a difficult question. In particular, we have shown that no convex losses can be a good surrogate in Chapter 5. We have seen there might exist continuous and differentiable losses that are consistent with regard to the $0/1$ loss, but it is still an open problem.

Does there exist a simple principled way to train the adversarial attacks problem for both the classifier and the attacker?

Since no convex loss can be a good surrogate for the adversarial classification problem, the optimization of a suitable empirical risk would be a non-convex optimization problem which is misunderstood so far. The difficulty of this problem is also highlighted by the inner supremum which is also non-convex. Then there is still a gap to bridge to understand the optimization of the adversarial problem. In Chapter 4, we proposed the following adversarial problem where the classifier and the attacker compete as follows

$$\inf_{\mu\in\mathcal{M}_1^+(\mathcal{H})}\ \sup_{\mathbb{Q}\in\mathcal{A}_\varepsilon(\mathbb{P})} \mathbb{E}_{\mu\sim\mathcal{H},(x,y)\sim\mathbb{Q}}[L(h(x),y)]$$

This naturally leads to understand the adversarial problem as game between the attacker and the classifier with utility $\mathbb{E}_{\mu\sim\mathcal{H},(x,y)\sim\mathbb{Q}}[L(h(x),y)]$. We showed the existence of Nash equilibria for this game in Chapter 4. Although, we propose a way to learn the optimal mixtures of classifiers when their number is finite, the question of computing equilibria has not been studied and would be a natural further step. On one hand, it would help to build a robust classifier against every attack in $\mathcal{A}_\varepsilon(\mathbb{P})$, and on the other hand, the attacker that would have been built would be robust to change in the mixture of classifiers. This problem is a min-max optimization problem over the set of probability distributions, hence a difficult problem. Although the problem writes as a convex-concave problem over the space of distributions, the utility is not geodisically convex-concave in the Wassertein-2 space. Applying directly results on Wasserstein Gradient Flow is not possible. Deriving a tractable algorithm with convergence guarantees is also difficult. There have been some attempts by the Machine Learning community to find mixed Nash equilibria using optimization over distribution techniques [Hsieh et al., 2019, Domingo-Enrich et al., 2020] with applications to Generative Adversarial Networks for instance. Understanding and finding equilibria in games in Machine Learning such as the adversarial attacks problem and GANs is essential for the community to understand better these problems.

7.2.2 Understanding the Learning Dynamics in the Adversarial Setting

Statistical and Computational learning theory have focused on analyzing what can be inferred on the error outside the training set, often called generalization error. To analyze it, the risk is decomposed in bias-complexity form. This bias complexity tradeoff has been question recently by double descent phenomenon [Belkin et al., 2018, 2019], suggesting that higher complexity models might lead to lower generalization errors. These recent findings underline the lack of understanding we have about generalization of Neural Networks.

Analyzing generalization in the adversarial setting case is still an underdeveloped question. There have been some works using Rademacher complexity [Yin et al., 2019, Awasthi et al., 2020] to craft uniform convergence bounds. But, to our knowledge, very few works have focused on understanding the bias-complexity tradeoff in the adversarial case.

How does statistical generalization work in the adversarial attacks setting?

This problem can be attacked from different angles. First, understanding the need or not of randomization for obtaining optimally robust classifiers is an important problem. From Chapter 4 and Pydi and Jog [2021b], the answer of this question depends mostly on two things: the set of hypotheses $\mathcal{H}$ and the distribution $\mathbb{P}$. If $\mathcal{H}$ is small and cannot be optimal for $\mathbb{P}$, there might be an interest for randomization, while when it is complex and sufficiently expressive, for instance the set of measurable functions [Pydi and Jog, 2021b], there is no need for randomization.

However, choosing complex sets of hypotheses might lead to overfitting, justifying the need of understanding generalization properties of randomized classifiers in the adversarial setting. While the question of uniform convergence bounds have been treated, generalization of randomized classifiers in the adversarial setting has only been partly tackled by Viallard et al. [2021] under the PAC-Bayes framework [Guedj, 2019] .

Beyond PAC-like bounds, convergence rates of optimal classifiers in the adversarial setting as it was done by Fischer and Steinwart [2020] in the case of kernel least squares regression is an important and yet not studied problem. Even the question of the choice of the norm for the convergence is difficult since the adversarial setting involves points outside the support of the distribution.

7.2.3 Scaling Provably Robust Neural Networks

In Chapter 5, we provided a general method to build provably Lipschitz layers. However, every single methods only lead to limited results on CIFAR10 dataset [Krizhevsky et al.] with standard accuracies under 80% and certifiable accuracies under 65% for $\varepsilon = 36/255$. The performances are far under the state-of-the-art on CIFAR10 standard classification task ($> 95\%$). There is still a huge gap we need to bridge to have performant certifiably robust neural networks. Since we are unable to reach decent performances on simple datasets, the question of being robust on larger datasets as ImageNet [Deng et al., 2009] is a bit anticipated.

Is it possible to build non-vacuous certifiable neural networks on highly-dimensional large-scale datasets?

Building robust neural networks with deterministic non-vacuous guarantees is an active research area. Current methods that scale on ImageNet dataset rely on non-deterministic bounds using for instance randomized smoothing [Cohen et al., 2019, Salman et al., 2019]. The advantages and the weaknesses of these methods are the same: while deterministic methods highly depend on the structure of the networks, randomized smoothing methods are agnostic to the structure of neural networks. One may hope using the structure of deep neural networks to get provable strategies. The question of robustness is also understudied for the recent Transformers [Vaswani et al., 2017] neural networks whose basic element is an attention block:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

where d_K is the common dimension of Q and K. The Transformers architectures are today state of the art both in NLP tasks [Devlin et al., 2018] and computer vision tasks [Dosovitskiy et al., 2020]. These approaches are very recent, and their robustness have not been investigated yet. This question definitely worth more attention!

Beyond, this question of scalability and robustness of architectures, one may ask the feasibility of such tasks. Enforcing Lipschitz constraints on the networks may hinder the networks performances. In complex datasets like ImageNet, it might not be possible to get simultaneously good performances and non-vacuous certificates. Moreover, the proposed defenses often rely on a single norm, often the ℓ^2 norm. Designing networks that are "universally" robust for human perception is a utopia, that we may never reach.

A On the Robustness of Randomized Classifiers to Adversarial Examples

This paper investigates the theory of robustness against adversarial attacks. We focus on randomized classifiers (*i.e.* classifiers that output random variables) and provide a thorough analysis of their behavior through the lens of statistical learning theory and information theory. To this aim, we introduce a new notion of robustness for randomized classifiers, enforcing local Lipschitzness using probability metrics. Equipped with this definition, we make two new contributions. The first one consists in devising a new upper bound on the adversarial generalization gap of randomized classifiers. More precisely, we devise bounds on the generalization gap and the adversarial gap (*i.e.* the gap between the risk and the worst-case risk under attack) of randomized classifiers. The second contribution presents a yet simple but efficient noise injection method to design robust randomized classifiers. We show that our results are applicable to a wide range of machine learning models under mild hypotheses. We further corroborate our findings with experimental results using deep neural networks on standard image datasets, namely CIFAR-10 and CIFAR-100. All robust models we trained can simultaneously achieve state-of-the-art accuracy (over 0.82 clean accuracy on CIFAR-10) and enjoy *guaranteed* robust accuracy bounds (0.45 against ℓ_2 adversaries with magnitude 0.5 on CIFAR-10).

A.1 Introduction

In the last few years, there has been a growing concern on adversarial example attacks in machine learning. An adversarial attack refers to a small (humanly imperceptible) change of an input specifically designed to fool a machine learning model. These attacks have recently come to light thanks to works by Biggio et al. [2013] and Szegedy et al. [2014] studying deep neural networks for image classification, although it was an existing topic in spam filter analysis [Dalvi et al., 2004, Lowd and Meek, 2005, Globerson et al., 2006]. The vulnerability of state-of-the-art classifiers to these attacks has genuine security implications especially for deep neural networks used in AI-driven technologies such as self-driving cars, as repetitively demonstrated by Sharif et al. [2016], Sitawarin et al. [2018] and Yao et al. [2020]. Besides security issues, this shows how little we know about the worst-case behaviors of models the industry uses daily. It is essential for the community to understand the very nature of this phenomenon in order to mitigate the threat.

Accordingly, a large body of works has been trying to design new models that would be less vulnerable to the adversarial setting [Goodfellow et al., 2015b, Metzen et al., 2017, Xie et al., 2018, Hu et al., 2019, Verma and Swami, 2019] but most of them were proven (in time) to offer only limited protection against more sophisticated attacks [Carlini et al., 2017, He et al., 2017, Athalye et al., 2018b, Croce et al., 2020b, Tramer et al., 2020]. Among the defense strategies, randomization has proven effective in some contexts [Xie et al., 2018, Dhillon et al., 2018, Liu et al., 2018, Rakin et al., 2018]. Albeit these significant efforts, randomization techniques lack theoretical arguments. In this paper, we generalize the prior results from Pinot et al. [2019] by studying a general class of randomized classifiers, including randomized neural networks, for which we demonstrate adversarial robustness guarantees and analyze their generalization properties.

A.1.1 Supervised learning for image classification

Let us consider the supervised classification problem with an input space $\mathcal{X}$ and an output space $\mathcal{Y}$. In the following, w.l.o.g. we will consider $\mathcal{X} \subset [-1,1]^d$ to be a set of images, and $\mathcal{Y} := \{1,\ldots,K\}$ a set of labels describing them. The goal of a supervised machine learning algorithm is to design classifier that maps any image $x \in \mathcal{X}$ to a label $y \in \mathcal{Y}$. To do so, the learner has access to a *training sample* of n image-label pairs $\mathcal{S} := \{(x_1,y_1),\ldots,(x_n,y_n)\}$. Each training pair (x_i,y_i) is assumed to be drawn *i.i.d.* from a ground-truth distribution $\mathbb{P}$. To build a classifier, the usual strategy is to select a hypothesis function $h : \mathcal{X} \to \mathcal{Y}$ from a pre-defined hypothesis class $\mathcal{H}$ to minimize the *risk* with respect to $\mathbb{P}$. This risk minimization problem writes

$$\inf_{h\in\mathcal{H}} \mathcal{R}(h) := \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[L_{0/1}(h(x),y)\right], \tag{A.1}$$

where $L_{0/1}$, the $0/1$ loss, outputs 1 when $h(x) \neq y$, and zero otherwise.

In practice, the learner does not have access to the ground-truth distribution; hence it cannot estimate the risk $\mathcal{R}(h)$. To find an approximate solution for Problem (A.1), a learning algorithm solves the *empirical risk minimization* problem instead. In this case, we simply replace the risk by its empirical counterpart over the training sample $\mathcal{S} := \{(x_1,y_1),\ldots,(x_n,y_n)\}$. The empirical risk minimization problem writes

$$\inf_{h\in\mathcal{H}} \widehat{\mathcal{R}}(h) := \frac{1}{n}\sum_{i=1}^{n} L_{0/1}(h(x_i),y_i). \tag{A.2}$$

Then, to evaluate how far the selected hypothesis is from the optimum, one wants to upper bound the difference between the risk and the empirical risk of any $h \in \mathcal{H}$. This difference is known as the *generalization gap*.

A.1.2 Classification in the presence of an adversary

Given a hypothesis $h \in \mathcal{H}$ and a sample $(x,y) \sim \mathbb{P}$, the goal of an adversary is to find a perturbation $\tau \in \mathcal{X}$ such that the following assertions *both* hold. First, the perturbation is imperceptible to humans. This means that a human cannot visually distinguish the standard example x from the *adversarial example* $x+\tau$. Second, the perturbation modifies x enough to make the classifier misclassify. More formally, the adversary seeks a perturbation $\tau \in \mathcal{X}$ such that $h(x+\tau) \neq y$.

Although the notion of imperceptible modification is very natural for humans, it is genuinely hard to formalize. Despite these difficulties, in the image classification setting, a sufficient condition to ensure that the attack will remain undetected is to constrain the perturbation τ to have a small ℓ_p norm. This means that for any $p \in [1,\infty]$, there exists a threshold $\varepsilon > 0$ for which any perturbation τ is imperceptible as soon as $\|\tau\|_p \leq \varepsilon$. The literature on adversarial attacks for image classification usually uses either an ℓ_∞ norm akin Madry et al. [2018a] or an ℓ_2 norm akin Carlini et al. [2017] as a surrogate for imperceptibility. Other authors such as Chen et al. [2018a] and Papernot et al. [2016c] also used an ℓ_1 norm or an ℓ_0 semi-norm.

To account for adversaries possibly manipulating the input images, one needs to revisit the standard risk minimization by incorporating the adversary in the problem. The goal becomes to minimize the *worst-case* risk under ε-bounded manipulations. We call this problem the *adversarial risk minimization*. It writes

$$\inf_{h \in \mathcal{H}} \mathcal{R}_\varepsilon(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sup_{\tau \in B_p(\varepsilon)} L_{0/1}(h(x+\tau), y)\right], \tag{A.3}$$

where $B_p(\varepsilon) := \{\tau \in \mathcal{X} \mid \|\tau\|_p \leq \varepsilon\}$. In this new formulation, the adversary focuses on optimizing the inner maximization, while the learner tries to get the best hypothesis from $\mathcal{H}$ "under attack". By analogy with the standard setting, given n training examples $\mathcal{S} := \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to find an approximate solution to the adversarial risk minimization by studying its empirical counterpart, the *empirical adversarial risk minimization*. This optimization problem writes

$$\inf_{h \in \mathcal{H}} \widehat{\mathcal{R}}_N(h) := \frac{1}{n}\sum_{i=1}^{n} \sup_{\tau \in B_p(\varepsilon)} L_{0/1}(h(x_i+\tau), y_i)\,. \tag{A.4}$$

In the presence of an adversary, two major issues appear in the empirical risk minimization. First, as recently pointed out by Madry et al. [2018a], the adversarial generalization error (*i.e.* the gap between the empirical adversarial risk and the adversarial risk) can be much larger than in the standard setting. Indeed, the adversary makes the problem dependent on the dimension of $\mathcal{X}$. Hence, in high-dimension (*e.g.* for images) one needs much more samples to classify correctly as pointed out by Schmidt et al. [2018] as well as Simon-Gabriel et al. [2019]. Moreover, finding an approximate solution to the adversarial risk minimization is not always sufficient. Indeed, recent works by Tsipras et al. [2019] and Zhang et al. [2019a] g ave theoretical evidence that training a robust model may lead to an increase of its standard risk. Hence finding a good approximation for Problem (A.3) may lead to a poor solution for Problem (A.1). Accordingly, it is natural to wonder whether we can ***find a class of models $\mathcal{H}$ for which we can control both the standard and adversarial risks?***

In this paper, we provide answers to the above question by conducting an in depth analysis of a special class of models called randomized classifiers, *i.e.* classifiers that output random variables instead of labels. Our main contributions summarize as follows.

A.1.3 Contributions

Our first contribution consists in studying randomized classifiers. By analogy with the deterministic case, we define a notion of robustness for randomized classifiers. This definition amounts to making the classifier locally Lipschitz with respect to the ℓ_p norm on $\mathcal{X}$, and a probability metric on $\mathcal{Y}$ (*e.g.* the total variation distance or the Renyi divergence). More precisely, if we denote D the probability metric at hand, a randomized classifier m is called (ε, α)-robust w.r.t. D if for any $x, x' \in \mathcal{X}$

$$\|x - x'\|_p \leq \varepsilon \implies D(m(x), m(x')) \leq \alpha.$$

Denoting $\mathcal{M}_D(\varepsilon,\alpha)$ the class of randomized classifiers that respect this local Lipschitz condition, we present the following results.

1. If D is either the total variation distance or the Renyi divergence, we show that for any $m \in \mathcal{M}_D(\varepsilon,\alpha)$, we can upper-bound the gap between the risk and the adversarial risk of m. Notably, if D is the total variation distance, for any $m \in \mathcal{M}_D(\varepsilon,\alpha)$ we have $\mathcal{R}_\varepsilon(m) - \mathcal{R}(m) \leq \alpha$. Hence, α controls the maximal trade-off between robust and standard accuracy for locally Lipschitz randomized classifier. We demonstrate similar results when D is the Renyi divergence showing that $\mathcal{R}_\varepsilon(m) - \mathcal{R}(m) \leq 1 - O(e^{-\alpha})$. This means that, for the class of locally Lipschitz randomized classifiers, solving the risk minimization problem, i.e., Problem (A.1), gives an approximate solution to the adversarial risk minimization problem, i.e., Problem (A.3), up to an additive factor that depends on the robustness parameter α.

2. We devise an upper-bound on the generalization gap of any m in $\mathcal{M}_D(\varepsilon,\alpha)$. In particular, when D is the total variation distance, we demonstrate that for any $m \in \mathcal{M}_D(\varepsilon,\alpha)$ we have
$$\mathcal{R}(m) - \widehat{\mathcal{R}}(m) \leq O\left(\sqrt{\frac{N \times K}{n}}\right) + \alpha,$$
where N is the external ε-covering number of the input samples. This means that, when $N/n \underset{n\to\infty}{\to} 0$, solving the empirical risk minimization problem, i.e., Problem (A.2), on $\mathcal{M}_D(\varepsilon,\alpha)$ provides an approximate solution to the risk minimization problem, i.e., Problem (A.1). Since we can also bound the gap between the adversarial and the standard risk, we can combine the two results to bound the adversarial generalization gap on $\mathcal{M}_D(\varepsilon,\alpha)$. Note however, that this result relies on a strong assumption on $\mathcal{X}$ that does not always avoid dimensionality issues. The problem of finding a subclass of $\mathcal{M}_D(\varepsilon,\alpha)$ that provides tighter generalization bounds is an open question.

For our second contribution, we present a practical way to design this class $\mathcal{M}(\varepsilon,\alpha)$ by using a simple yet efficient noise injection scheme. This allows us to build randomized classifiers from state-of-the-art machine learning models, including deep neural networks. More precisely our contribution is as follows.

1. Based on information-theoretic properties of the total variation distance and the Renyi divergence (e.g., the data processing inequality) we design a noise injection scheme to turn a state-of-the-art machine learning model into a robust randomized classifier. More formally, Let us denote Φ the c.d.f. of a standard Gaussian distribution. Let us consider h a deterministic hypothesis, we show that the randomized classifier $m : x \mapsto h(x+n)$ with $n \sim \mathcal{N}(0, \sigma^2 I_d)$ is both $(\alpha_2, \frac{(\alpha_2)^2}{2\sigma})$-robust w.r.t. the Renyi divergence and $(\alpha_2,\ 2\Phi(\frac{\alpha_2}{2\sigma}) - 1)$-robust w.r.t. the total variation distance. Our results on randomized classifiers are applicable to a wide range of machine learning models including deep neural networks.

2. We further corroborate our theoretical results with experiments using deep neural networks on standard image datasets, namely CIFAR-10 and CIFAR-100 [Krizhevsky and

Hinton, 2009]. These models can simultaneously provide accurate prediction (over 0.82 clean accuracy on CIFAR-10) and reasonable robustness against ℓ_2 adversarial examples (0.45 against ℓ_2 adversaries with magnitude 0.5 on CIFAR-10).

A.2 Related Work

Contrary to other notions such as training corruption, a.k.a. poisoning attacks [Kearns and Li, 1993, Kearns et al., 1994], the theoretical study of adversarial robustness is still in its infancy. So far, empirical observations tend to show that 1) adversarial examples on state-of-the-art models are hard to mitigate and 2) robust training methods give poor generalization performances. Some recent works started to study the problem through the lens of learning theory either to understand the links between robustness and accuracy or to provide bounds on the generalization gap of current learning procedures in the adversarial setting.

A.2.1 Accuracy vs robustness trade-off

A first line of research [Su et al., 2018, Jetley et al., 2018, Tsipras et al., 2019] suggests that designing robust models might be inconsistent with standard accuracy. These works argue with experiments and toy examples that robust and standard classification are two concurrent problems. Following this line, Zhang et al. [2019a] observed that the adversarial risk of any hypothesis h decomposes as follows,

$$\mathcal{R}_\varepsilon(h) = \mathcal{R}(h) + \mathcal{R}_\varepsilon^{>0}(h), \tag{A.5}$$

where $\mathcal{R}_\varepsilon^{>0}(m)$ is the amount of risk that the adversary gets with *non-null* perturbations. Looking at Equation (A.5), we realize that minimizing the adversarial risk is not enough to control standard accuracy, as one could only optimize over the second term. This indicates that adversarial risk minimization, i.e., Problem (A.3), is harder to solve than the standard risk minimization, i.e., Problem (A.1).

While this indicates that both goals maybe difficult be achieve simultaneously, Equation (A.5), along with the empirical studies from the literature do not highlight any fundamental trade-off between robustness and accuracy. Moreover, no upper-bound on $\mathcal{R}_\varepsilon^{>0}(h)$ has been demonstrated yet. Hence the questions whether this trade-off exists and can be controlled remain open. In this paper, we provide a rigorous answer to these questions by identifying classes $\mathcal{M}_D(\varepsilon, \alpha)$ of randomized classifiers for which we can upper bound the trade-off term $\mathcal{R}_\varepsilon^{>0}(m)$ for any $m \in \mathcal{M}_D(\varepsilon, \alpha)$. Hence, we can control the maximum loss of accuracy that the model can suffer in the adversarial setting. It also challenges the intuitions developed by previous works [Su et al., 2018, Jetley et al., 2018, Tsipras et al., 2019] and argues in favor of using randomized mechanisms as a defense against adversarial attacks.

A.2.2 Studying adversarial generalization

To further compare the hardness of the two problems, a recent line of research began to explore the notion of adversarial generalization gap. In this line, Schmidt et al. [2018] presented some first intuitions by studying a simplified binary classification framework where $\mathbb{P}$ is a mixture of multi-dimensional Gaussian distributions. In this framework the authors show that without attacks, we only need $O(1)$ training samples to have a small generalization gap. But against an ℓ_∞ adversary, we need $O(\sqrt{d})$ training samples instead. In the discussion of their work, the authors present the problem of obtaining similar results without making any assumption about the distribution as an open problem.

This issue was recently studied using the Rademacher complexity by Khim and Loh [2018], Yin et al. [2019] and Awasthi et al. [2020]. These papers relate the adversarial generalization error of linear classifiers and one-hidden layer neural networks with the dimension of the problem. They show that the adversarial generalization depends on the dimension of the problem. At a first glance, the difficulty of adversarial generalization seems to contradict previous conclusions on the link between robustness and generalization presented by Xu and Mannor [2012]. But, as we will discuss in the sequel, these results assume that the input space $\mathcal{X}$ can be partitioned in $O(1)$ sub-space in which the classification function has small variations. This assumption may not always hold when dealing with high dimensional input spaces (e.g., images) and very sophisticated classification algorithms (e.g., deep neural networks).

Going further, it should be noted that the generalization gap measures only the difference between empirical and theoretical risks. In practice, the empirical adversarial risk is hard to estimate, since we cannot compute the exact solution to the inner maximization problem. The following question therefore remains open: even if we can set up a learning procedure with a controlled generalization gap, can we give guarantees on the standard and adversarial risks? In this paper, we start answering this question by providing techniques that provably offer both small standard risk and reasonable robustness against adversarial examples (see Section A.1.3 for more details).

A.2.3 Defense against adversarial examples based on noise injection

Injecting noise into algorithms to improve train time robustness has been used for ages in detection and signal processing tasks [Zozor and Amblard, 1999, Chapeau-Blondeau and Rousseau, 2004, Mitaim and Kosko, 1998, Grandvalet et al., 1997]. It has also been extensively studied in several machine learning and optimization fields, e.g., robust optimization [Ben-Tal et al., 2009] and data augmentation techniques [Perez and Wang, 2017]. Concurrently to our work, noise injection techniques have been adopted by the adversarial defense community under the *randomized smoothing* name. The idea of provable defense through noise injection was first proposed by Lecuyer et al. [2019] and refined by Li et al. [2019a], Cohen et al. [2019], Salman et al. [2019] and Yang et al. [2020a]. The rational behind randomized smoothing is very simple: smooth h *after training* by convolution with a Gaussian measure to build a more stable classifier. Our work belongs to the same line of research, but the nature of our results is different. Randomized smoothing is an ensemble method that builds a deterministic classifier by smoothing a pre-trained model with a Gaussian kernel. This scheme requires to compute a Monte-Carlo estimation of the smoothed classifier; hence requiring many rounds of evaluations to output a deterministic label. Our method is based on randomization and only requires one evaluation round for inferring a label, making the prediction randomized and computationally efficient. While randomized smoothing focuses on the construction of certified defenses, we study the generalization properties of randomized mechanisms both in the standard and the adversarial setting. Our analysis presents the fundamental properties of randomized defenses, including (but not limited to) randomized smoothing (c.f. Section A.7).

A.3 Definition of Risk and Robustness for Randomized classifiers

In this work, the goal is to analyze how randomized classifiers can solve the problem of classification in the presence of an adversary. Let us start by defining what we mean by randomized classifiers.

Remark 9 (Note on measurability)**.** *Trough the paper, we assume every spaces $\mathcal{Z}$ to be associated with a σ-algebra denoted $\mathcal{A}(\mathcal{Z})$. Furthermore, we denote $\mathcal{M}_1^+(\mathcal{Z})$ the set of probability distributions defined on the measurable space $(\mathcal{Z}, \mathcal{A}(\mathcal{Z}))$. In the following, for simplicity, we refer to $\mathcal{A}(\mathcal{Z})$ only when necessary.*

Definition 25 (Probabilistic mapping)**.** *Let $\mathcal{Z}$ and $\mathcal{Z}'$ be two arbitrary spaces. A* probabilistic mapping *from $\mathcal{Z}$ to $\mathcal{Z}'$ is a mapping $m : \mathcal{Z} \to \mathcal{M}_1^+(\mathcal{Z}')$, where $\mathcal{M}_1^+(\mathcal{Z}')$ is the space of probability measures on $\mathcal{Z}'$. When $\mathcal{Z} = \mathcal{X}$ and $\mathcal{Z}' = \mathcal{Y}$, m is called a* randomized classifier. *To get a numerical answer for an input x, we sample $\hat{y} \sim m(x)$.*

Any mapping can be considered as a probabilistic mapping, whether it explicitly considers randomization or not. In fact, any deterministic classifier can be considered as a randomized one, since it can be characterized by a Dirac measure. Accordingly, the definition of a randomized classifier is fully general and equally consider classifiers with or without randomization scheme.

A.3.1 Risk and adversarial risk for randomized classifiers

To analyze this new hypothesis class, we can adapt the concepts of risk and adversarial risk for a randomized classifier. The loss function we use is the natural extension of the $0/1$ loss to the randomized regime. Given a randomized classifier m and a sample $(x, y) \sim \mathbb{P}$ it writes

$$L_{0/1}(m(x), y) := \mathbb{E}_{\hat{y} \sim m(x)}[\mathbb{1}\{\hat{y} \neq y\}]. \tag{A.6}$$

This loss function evaluates the probability of misclassification of m on a data sample $(x, y) \sim \mathbb{P}$. Accordingly, the risk of m with respect to $\mathbb{P}$ writes

$$\mathcal{R}(m) := \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[L_{0/1}(m(x), y)\right]. \tag{A.7}$$

Finally, given m and $(x, y) \sim \mathbb{P}$, the adversary seeks a perturbation $\tau \in B_p(\varepsilon)$ that maximizes the expected error of the classifier on x (*i.e.* $\mathbb{E}_{\hat{y}\sim m(x+\tau)}[\mathbb{1}\{\hat{y} \neq y\}]$). Therefore, the adversarial risk of m under ε-bounded perturbations writes

$$\mathcal{R}_\varepsilon(m) := \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau \in B_p(\varepsilon)} L_{0/1}(m(x+\tau), y)\right]. \tag{A.8}$$

By analogy with the deterministic setting, we denote

$$\widehat{\mathcal{R}}(m) := \frac{1}{n}\sum_{i=1}^{n} L_{0/1}(m(x_i), y_i), \text{ and} \tag{A.9}$$

$$\widehat{\mathcal{R}}_N(m) := \frac{1}{n}\sum_{i=1}^{n} \sup_{\tau \in B_p(\varepsilon)} L_{0/1}(m(x_i+\tau), y_i), \tag{A.10}$$

the empirical risks of m for a given training sample $\mathcal{S} := \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

A.3.2 Robustness for randomized classifiers

We could define the notion of robustness for a randomized classifier depending on whether it misclassifies any test sample $(x, y) \sim \mathbb{P}$. But in practice, neither the adversary nor the model provider have access to the ground-truth distribution $\mathbb{P}$. Furthermore, in real-world scenarios, one wants to check before its deployment that the model is robust. Therefore, it is required for the classifier to be stable on the regions of the space where it already classifies correctly. Formally a (deterministic) classifier $c : \mathcal{X} \to \mathcal{Y}$ is called *robust* if for any $(x, y) \sim \mathbb{P}$ such that $c(x) = y$, and for any $\tau \in \mathcal{X}$ one has

$$\|\tau\|_p \leq \varepsilon \implies c(x) = c(x+\tau). \tag{A.11}$$

By analogy with this, we define robustness for a randomized classifier below.

Definition 26 (Robustness for a randomized classifier)**.** *A randomized classifier $m : \mathcal{X} \to \mathcal{M}_1^+(\mathcal{Y})$ is called* (ε, α)-robust *w.r.t. D if for any $x, \tau \in \mathcal{X}$, one has*

$$\|\tau\|_p \leq \varepsilon \implies D(m(x), m(x+\tau)) \leq \alpha\,.$$

Where D is a metric/divergence between two probability measures. Given such a metric/divergence D, we denote $\mathcal{M}_D(\varepsilon, \alpha)$ the set of all randomized classifiers that are (ε, α)-robust *w.r.t. D.*

Note that we did not add the constraint that m classifies well on $(x, y) \sim \mathbb{P}$, since it is already encompassed in the probability distribution itself. If the two probabilities $m(x)$ and $m(x+\tau)$ are close, and if $m(x)$ outputs y with high probability, then it will be the same for $m(x+\tau)$. This formulation naturally raises the question of the choice of the metric D. Any choice of metric/divergence will instantiate a notion of adversarial robustness, and it should be carefully selected. In the present work, we focus our study on the total variation distance and the Renyi divergence. The question whether these metrics/divergences are more appropriate than others remains open but these two divergences are sufficiently general to cover a wide range of other definitions (see Appendix A.11 for more details). Furthermore, these notions of distance comply with both a theoretical analysis (Section A.5) and practical considerations (Section A.8).

A.3.3 Divergence and probability metrics

Let us now recall the definition of total variation distance and Renyi divergence. Let $\mathcal{Z}$ be an arbitrary space, and ρ, ρ' be two measures in $\mathcal{M}_1^+(\mathcal{Z})$[1]. The *total variation distance* between ρ and ρ' is

$$D_{TV}(\rho, \rho') := \sup_{Z \subset \mathcal{A}(\mathcal{Z})} |\rho(Z) - \rho'(Z)| \ , \tag{A.12}$$

[1]Recall from Definition 25 that $\mathcal{M}_1^+(\mathcal{Z})$ is the set of probability measures on $\mathcal{Z}$

where $\mathcal{A}(\mathcal{Z})$ is the σ-algebra associated with the set of measures $\mathcal{M}_1^+(\mathcal{Z})$. The total variation distance is one of the most commonly used probability metrics. It admits several very simple interpretations, and is a very useful tool in many mathematical fields such as probability theory, Bayesian statistics or optimal transport [Villani, 2003, Robert, 2007, Peyré et al., 2019]. In optimal transport, it can be rewritten as the solution of the Monge-Kantorovich problem with the cost function $\text{cost}(z, z') = \mathbb{1}\{z \neq z'\}$,

$$D_{TV}(\rho, \rho') = \inf \int_{\mathcal{Z}^2} \mathbb{1}\{z \neq z'\} d\pi(z, z') \ , \tag{A.13}$$

where the infimum is taken over all joint probability measures π in $\mathcal{M}_1^+(\mathcal{Z} \times \mathcal{Z})$ with marginals ρ and ρ'. According to this interpretation, it seems quite natural to consider the total variation distance as a relaxation of the trivial distance on $[0, 1]$ (for deterministic classifiers).

Let us now suppose that ρ and ρ' admit probability density functions g and g' according to a third measure ν. Then the *Renyi divergence of order* β between ρ and ρ' writes

$$D_\beta(\rho, \rho') := \frac{1}{\beta - 1} \log \int_{\mathcal{Y}} g'(y) \left(\frac{g(y)}{g'(y)} \right)^\beta d\nu(y) \ . \tag{A.14}$$

The Renyi divergence [Rényi, 1961] is a generalized divergence defined for any β on the interval $[1, \infty]$. It equals the Kullback-Leibler divergence when $\beta \to 1$, and the maximum divergence when $\beta \to \infty$. It also has the property of being non-decreasing with respect to β. This divergence is very common in machine learning and Information theory [van Erven and Harremos, 2014], especially in its Kullback-Leibler form as it is widely used as the loss function, i.e., cross entropy, of classification algorithms. In the remaining, we denote $\mathcal{M}_\beta(\varepsilon, \alpha)$ the set of (ε, α)-robust classifiers w.r.t. D_β.

Let us now give some properties of these divergences that will be useful for our analysis. First we recall the probability preservation property of the Renyi divergence, first presented by Langlois et al. [2014].

Proposition 17 (Langlois et al. [2014])**.** *Let ρ and ρ' be two measures in $\mathcal{M}_1^+(\mathcal{Z})$. Then for any $Z \in \mathcal{A}(\mathcal{Z})$, the following holds,*

$$\rho(Z) \leq \left(\exp\left(D_\beta(\rho, \rho')\right) \rho'(Z)\right)^{\frac{\beta-1}{\beta}} .$$

Now thanks to previous works by Gilardoni [2010] and Vajda [1970], we also get the following results relating the total variation distance and the Renyi divergence.

Proposition 18 (Inequality between total variation and Renyi divergence)**.** *Let ρ and ρ' be two measures in $\mathcal{M}_1^+(\mathcal{Z})$, and $\beta \geq 1$. Then the following holds,*

$$D_{TV}(\rho, \rho') \leq \min \left(\frac{3}{2} \left(\sqrt{1 + \frac{4 D_\beta(\rho, \rho')}{9}} - 1 \right)^{1/2}, \frac{\exp(D_\beta(\rho, \rho') + 1) - 1}{\exp(D_\beta(\rho, \rho') + 1) + 1} \right).$$

Proof. Thanks to Gilardoni [2010], one has

$$D_1(\rho, \rho') \geq 2D_{TV}(\rho, \rho')^2 + \frac{4D_{TV}(\rho, \rho')^4}{9}.$$

From which it follows that

$$D_{TV}(\rho, \rho') \leq \frac{3}{2}\left(\sqrt{1 + \frac{4D_1(\rho, \rho')}{9}} - 1\right)^{1/2}.$$

Moreover, using inequality from Vajda [1970], one gets

$$D_1(\rho, \rho') + 1 \geq \log\left(\frac{1 + D_{TV}(\rho, \rho')}{1 - D_{TV}(\rho, \rho')}\right).$$

This inequality leads to the following

$$\frac{\exp(D_1(\rho, \rho') + 1) - 1}{\exp(D_1(\rho, \rho') + 1) + 1} \geq D_{TV}(\rho, \rho').$$

By combining the above inequalities and by monotony of Renyi divergence regarding β, one obtains the expected result. □

From now on, we denote $\mathcal{M}_{TV}(\alpha, \alpha)$ and $\mathcal{M}_{\beta}(\alpha, \alpha)$ the set of (α, α)-*robust* classifiers respectively for D_{TV} and D_{β}. The next section gives bounds on the generalization gap in the standard and the adversarial settings for these specific hypothesis classes.

A.4 Risks' gap and Generalization gap for robust randomized classifiers

As discussed in Section A.2.1, we can always decompose the adversarial risk of a classifier $\mathcal{R}_{\varepsilon}(m)$ in two terms. First the standard risk $\mathcal{R}(m)$ and second the amount of risk the adversary creates with non-zero perturbations $\mathcal{R}_{\varepsilon}^{>0}(m)$. Hence minimizing $\mathcal{R}(m)$ can give poor values for $\mathcal{R}_{\varepsilon}(m)$ and vice-versa. In this section, we upper-bound the risks' gap $\mathcal{R}_{\varepsilon}^{>0}(m)$, *i.e.* the gap between the risk and the adversarial risk of a robust classifier.

A.4.1 Risks' gap for robust classifiers w.r.t. D_{TV}

First, let us consider $m \in \mathcal{M}_{TV}(\varepsilon, \alpha)$. We can control the loss of accuracy under attack of this classifier with the robustness parameter α.

Theorem 15 (Risk's gap for robust classifiers w.r.t D_{TV}). *Let $m \in \mathcal{M}_{TV}(\varepsilon, \alpha)$. Then we have*

$$\mathcal{R}_{\varepsilon}(m) \leq \mathcal{R}(m) + \alpha\,.$$

Proof. Let m be an (ε, α)-robust classifier w.r.t. D_{TV} , $(x, y) \sim \mathbb{P}$ and $\tau \in \mathcal{X}$ such that $\|\tau\|_p \leq \varepsilon$. By definition of the $0/1$ loss we have

$$L_{0/1}(m(x+\tau), y) = \mathbb{E}_{\hat{y}\sim m(x+\tau)}[\mathbb{1}\{\hat{y} \neq y\}].$$

Furthermore, by definition of the total variation distance we have

$$\mathbb{E}_{\hat{y}\sim m(x+\tau)}[\mathbb{1}\{\hat{y} \neq y\}] - \mathbb{E}_{\hat{y}\sim m(x)}[\mathbb{1}\{\hat{y} \neq y\}] \leq D_{TV}(m(x), m(x+\tau)).$$

Since $m \in \mathcal{M}_{TV}(\varepsilon, \alpha)$, the above amounts to write

$$L_{0/1}(m(x+\tau), y) - L_{0/1}(m(x), y) \leq \alpha.$$

Finally, this holds for any $(x, y) \sim \mathbb{P}$ and any ε bounded perturbation τ, then we get

$$\mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} L_{0/1}(m(x+\tau), y)\right] - \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[L_{0/1}(m(x), y)\right] \leq \alpha.$$

The above inequality concludes the proof. □

This result means that if we can design a class $\mathcal{M}_{TV}(\varepsilon, \alpha)$ with small enough α, then minimizing the risk of $m \in \mathcal{M}_{TV}(\varepsilon, \alpha)$ is also sufficient to control the adversarial risk. It is relatively easy to obtain, but it has an interesting consequence on the understanding we have of the trade-off between robustness and accuracy. It says that there exists some classes of randomized classifiers for which robustness and standard accuracy may not be at odds, since we can upper-bound the maximal loss of accuracy the model may suffer under attack. This questions previous intuitions developed on deterministic classifiers by Su et al. [2018], Jetley et al. [2018], Tsipras et al. [2019] and Zhang et al. [2019a] and advocates for the use of randomization schemes as defenses against adversarial attacks. Note, however, that we did not evade the trade-off between robustness and accuracy, we only showed that with certain hypothesis classes it can be controlled.

A.4.2 Risks' gap for robust classifiers w.r.t. D_β

We now extend the previous results the Renyi divergence. We show that, for any randomized classifier in $\mathcal{M}_\beta(\varepsilon, \alpha)$, we can bound the gap between the risk and the adversarial risk of m. Using the Renyi divergence, the factor that controls the classifier's loss of accuracy under attack can be either multiplicative or additive, and depends both on the robustness parameter α and on the divergence parameter β.

Theorem 16 (Multiplicative risks' gap for Renyi-robust classifiers)**.** *Let $m \in \mathcal{M}_\beta(\varepsilon, \alpha)$. Then we have*

$$\mathcal{R}_\varepsilon(m) \leq (e^\alpha \mathcal{R}(m))^{\frac{\beta-1}{\beta}}.$$

Proof. Let m be an (ε, α)-robust classifier w.r.t. D_β, $(x, y) \sim \mathbb{P}$ and $\tau \in \mathcal{X}$ such that $\|\tau\|_p \leq \varepsilon$. With the same reasoning as above, and with Proposition 17, we get

$$\begin{aligned}
L_{0/1}(m(x+\tau), y) &= \mathbb{E}_{\hat{y}\sim m(x+\tau)}[\mathbb{1}\{\hat{y} \neq y\}] \\
&= \mathbb{P}_{\hat{y}\sim m(x+\tau)}[\hat{y} \neq y] \\
&\leq \left(e^{D_\beta(m(x+\tau), m(x))}\mathbb{P}_{\hat{y}\sim m(x)}[\hat{y} \neq y]\right)^{\frac{\beta-1}{\beta}} \quad \text{(Prop. 17)} \\
&= \left(e^{D_\beta(m(x+\tau), m(x))}\mathbb{E}_{\hat{y}\sim m(x)}[\mathbb{1}\{\hat{y} \neq y\}]\right)^{\frac{\beta-1}{\beta}} \\
&\leq \left(e^{\alpha} L_{0/1}(m(x), y)\right)^{\frac{\beta-1}{\beta}}.
\end{aligned}$$

Since this holds for any $(x, y) \sim \mathbb{P}$ and any ε bounded perturbation τ, we get

$$\begin{aligned}
\mathcal{R}_\varepsilon(m) &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sup_{\tau\in B_p(\varepsilon)} L_{0/1}(m(x+\tau), y)\right] \\
&\leq \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[e^{\frac{\beta-1}{\beta}\alpha} L_{0/1}(m(x), y)^{\frac{\beta-1}{\beta}}\right] \\
&\leq e^{\frac{\beta-1}{\beta}\alpha}\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L_{0/1}(m(x), y)^{\frac{\beta-1}{\beta}}\right].
\end{aligned}$$

Finally, using the Jensen inequality, one gets

$$\leq e^{\frac{\beta-1}{\beta}\alpha}\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L_{0/1}(m(x), y)\right]^{\frac{\beta-1}{\beta}} = \left(e^{\alpha}\mathcal{R}(m)\right)^{\frac{\beta-1}{\beta}}.$$

The above inequality concludes the proof. ☐

This first result gives a multiplicative bound on the gap between the standard and adversarial risks. This means that if we can design a class $\mathcal{M}_\beta(\varepsilon, \alpha)$ with small enough α, and big enough β, then minimizing the risk of any $m \in \mathcal{M}_\beta(\varepsilon, \alpha)$ is sufficient to also minimize the adversarial risk of m. Nevertheless, multiplicative factors are not easy to analyze.

Remark 10. *More general bounds can be computed if we assume that for every randomized classifier m there exists a convex function $\mathbf{f}$ such that for all x and τ with $\|\tau\|_p \leq \varepsilon$, we have $m(x)(Z) \leq \mathbf{f}(m(x+\tau)(Z))$ for all measurable sets Z. In this case, we get $\mathcal{R}_\varepsilon(m) \leq \mathbf{f}(\mathcal{R}(m))$. This has a close link with randomized smoothing [Cohen et al., 2019] and f-differential privacy [**?**] where both try to fit the best possible $\mathbf{f}$ using Neyman-Pearson lemma.*

The following result provides an additive counterpart to Theorem 16. It gives a control over the loss of accuracy under attack with respect to the robustness parameter α and the Shannon entropy of m.

Theorem 17 (Additive risks' gap for Renyi-robust classifiers)**.** *Let $m \in \mathcal{M}_\beta(\varepsilon, \alpha)$, then we have*

$$\mathcal{R}_\varepsilon(m) - \mathcal{R}(m) \leq 1 - e^{-\alpha}\mathbb{E}_{x\sim\mathcal{D}_{|\mathcal{X}}}\left[e^{-H(m(x))}\right]$$

*where H is the Shannon entropy (*i.e. *for any $\rho \in \mathcal{M}_1^+(\mathcal{Y})$, $H(\rho) = -\sum_{k\in\mathcal{Y}} \rho_k \log(\rho_k)$) and $\mathcal{D}_{|\mathcal{X}}$ is the marginal distribution of $\mathbb{P}$ for $\mathcal{X}$.*

Proof. Let $m \in \mathcal{M}_\beta(\varepsilon, \alpha)$, then

$$
\begin{aligned}
&\mathcal{R}_\varepsilon(m) - \mathcal{R}(m) \\
&= \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} L_{0/1}(m(x+\tau), y) - L_{0/1}(m(x), y)\right].
\end{aligned}
$$

By definition of the $0/1$ loss, this amounts to write

$$
\begin{aligned}
&= \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} \mathbb{E}_{\hat{y}_{\text{adv}}\sim m(x+\tau), \hat{y}\sim m(x)}[\mathbb{1}(\hat{y}_{\text{adv}} \neq y) - \mathbb{1}(\hat{y} \neq y)]\right] \\
&\leq \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} \mathbb{E}_{\hat{y}_{\text{adv}}\sim m(x+\tau), \hat{y}\sim m(x)}[\mathbb{1}(\hat{y}_{\text{adv}} \neq \hat{y})]\right] \\
&= \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} \mathbb{P}_{\hat{y}_{\text{adv}}\sim m(x+\tau), \hat{y}\sim m(x)}[\hat{y}_{\text{adv}} \neq \hat{y}]\right] \\
&= \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} 1 - \mathbb{P}_{\hat{y}_{\text{adv}}\sim m(x+\tau), \hat{y}\sim m(x)}[\hat{y}_{\text{adv}} = \hat{y}]\right] \\
&= \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} 1 - \sum_{i=1}^{K} m(x)_i \times m(x+\tau)_i\right].
\end{aligned}
$$

Now, note that for any $(x, y) \sim \mathbb{P}$ and $\tau \in \mathcal{X}$, by definition of a probability vector in $\mathcal{M}_1^+(\mathcal{Y})$, and thanks to Jensen inequality we can write

$$
\sum_{i=1}^{K} m(x)_i \times m(x+\tau)_i \geq \exp\left(\sum_{i=1}^{K} m(x)_i \log m(x+\tau)_i\right).
$$

Then by definition of the entropy and the Kullback Leibler divergence we have

$$
\exp\left(\sum_{i=1}^{K} m(x)_i \log m(x+\tau)_i\right) = \exp\big(-D_1(m(x), m(x+\tau)) - H(m(x))\big).
$$

Finally, by combining the above inequalities and since $m \in \mathcal{M}_\beta(\varepsilon, \alpha)$ we get

$$
\mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} \mathbb{P}_{\hat{y}_{\text{adv}}\sim m(x+\tau), \hat{y}\sim m(x)}(\hat{y}_{\text{adv}} \neq \hat{y})\right]
$$

$$\leq \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\sup_{\tau\in B_p(\varepsilon)} 1 - e^{-D_1(m(x),m(x+\tau))-H(m(x))}\right]$$
$$\leq \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[1 - e^{-\alpha-H(m(x))}\right] = 1 - e^{-\alpha}\mathbb{E}_{x\sim\mathbb{P}_{|\mathcal{X}}}\left[e^{-H(m(x))}\right].$$

The above inequality concludes the proof. □

This result is interesting because it relates the accuracy of m with the bound we obtain. In words, when $m(x)$ has large entropy (*i.e.* $H(m(x)) \to \log(K)$) the output distribution tends towards the uniform distribution; hence $\alpha \to 0$. This means that the classifier is very robust but also completely inaccurate, since it outputs classes uniformly at random. On the opposite, if $H(m(x)) \to 0$, then $\alpha \to \infty$. The classifier may be accurate, but it is not robust anymore (at least according to our definition). Hence we need to find a classifier that achieves a trade-off between robustness and accuracy.

A.5 Standard Generalization gap

In this section we devise generalization gap bounds for randomized classifiers when they are robust according either to the total variation distance or the Renyi divergence. To do so, we upper-bound the Rademacher complexity of the loss space for TV-robust classifiers

$$L_{\mathcal{M}_{TV}(\varepsilon,\alpha)} := \{(x,y) \mapsto L_{0/1}(h(x),y) \mid m \in \mathcal{M}_{TV}(\varepsilon,\alpha)\}.$$

The *empirical Rademacher complexity*, first introduced by Bartlett and Mendelson [2002], is one of the standard measures of generalization gap. It is particularly useful to obtain quality bounds for complex classes such as neural networks since it does not depend on the number of parameters in the network contrary to combinatorial notions such as the *VC dimension*.

Definition 27 (Rademacher complexity)**.** *For any class of real-valued functions $\mathcal{F} := \{(x,y) \mapsto \mathbb{R}\}$, given a training sample $\mathcal{S} = \{(x_1,y_1),\ldots,(x_n,y_n)\}$, the* empirical Rademacher complexity *of $\mathcal{F}$ is defined as*

$$Rad_{\mathcal{S}}(\mathcal{F}) := \frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{n} r_i f(x_i,y_i)\right],$$

whith r_i i.i.d. *drawn from a Rademacher measure,* i.e. $\mathbb{P}(r_i = 1) = \mathbb{P}(r_i = -1) = \frac{1}{2}$.

The empirical Rademacher complexity measures the uniform convergence rate of the empirical risk towards the risk on the function class $\mathcal{F}$ as demonstrated by Mohri et al. [2018]. Thanks to this notion of complexity, we can bound with high probability the generalization gap of any hypothesis m in a class $\mathcal{M}$.

Theorem 18 (Mohri et al. [2018]). *Let $\mathcal{M}$ be a class of possibly randomized classifiers and $L_{\mathcal{M}} := \{L_m : (x,y) \mapsto L_{0/1}(m(x),y) \mid m \in \mathcal{M}\}$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$, the following holds for any $m \in \mathcal{M}_{TV}(\varepsilon,\alpha)$,*

$$\mathcal{R}(m) - \widehat{\mathcal{R}}(m) \leq 2Rad_{\mathcal{S}}(L_{\mathcal{M}}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

A.5.1 Generalization error for robust classifiers

Accordingly, we want to upper bound the empirical Rademacher complexity of $L_{\mathcal{M}_{TV}(\varepsilon,\alpha)}$, which motivates the following definition.

Definition 28 (α-covering and external covering number). *Let us consider $(\mathcal{X}, \|.\|_p)$ a vector space equipped with the ℓ_p norm, $B \subset \mathcal{X}$ and $\alpha \geq 0$. Then*

- *$C = \{c_1, \ldots, c_m\}$ is an α-covering of B for the ℓ_p norm if for any $x \in B$ there exists $c_i \in C$ such that $\|x - c_i\|_p \leq \alpha$.*
- *The external covering number of B writes $N\Big(B, \|.\|_p, \alpha\Big)$. It is the minimal number of points one needs to build an α-covering of B for the ℓ_p norm.*

The covering number is a well-known measure that is often used in statistical learning theory [Shalev-Shwartz and Ben-David, 2014] and asymptotic statistics [Van der Vaart, 2000] to evaluate the complexity of a set of functions. Here we use it to evaluate the number of ℓ_p balls we need to cover the training samples, which gives us the following bound on the Rademacher complexity of $L_{\mathcal{M}_{TV}(\varepsilon,\alpha)}$.

Theorem 19 (Rademacher complexity for TV-robust classifiers). *Let $L_{\mathcal{M}_{TV}(\varepsilon,\alpha)}$ be the loss function class associated with $\mathcal{M}_{TV}(\varepsilon,\alpha)$. Then, for any $\mathcal{S} := \{(x_1,y_1),\ldots,(x_n,y_n)\}$, the following holds,*

$$\mathfrak{R}_{\mathcal{S}}\big(L_{\mathcal{M}_{TV}(\varepsilon,\alpha)}\big) \leq \sqrt{\frac{N \times K}{n}} + \alpha.$$

Where $N = N\Big(\{x_1,\ldots,x_n\}, \|.\|_p, \varepsilon\Big)$ is the ε-external covering number of the inputs $\{x_1,\ldots,x_n\}$ for the ℓ_p norm.

Proof. We denote $\mathcal{S} := \{(x_1,y_1),\ldots,(x_n,y_n)\}$ and $N = N\Big(\{x_1,\ldots,x_n\}, \|.\|_p, \varepsilon\Big)$. By definition of a covering number, there exists $C = \{c_1,\ldots,c_N\}$ an ε-covering of $\{x_1,\ldots x_n\}$ for the ℓ_p norm. Furthermore, for $j \in \{1,\ldots,N\}$ and $y \in \{1,\ldots,K\}$, we define

$$E_{y,j} = \left\{ i \in \{1,\ldots,n\} \mid y_i = y \text{ and } \underset{l \in \{1,\ldots,N\}}{\arg\min} \|x_i - c_l\| = j \right\}.$$

We also denote $E_j = \underset{y\in[K]}{\cup} E_{y,j}$. Finally, we denote $L_m : (x, y) \mapsto L_{0/1}(m(x), y)$. Then, by definition of the empirical Rademacher complexity, we can write

$$\mathfrak{R}_{\mathcal{S}}\big(L_{\mathcal{M}_{TV}(\varepsilon,\alpha)}\big) = \frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{m\in\mathcal{M}_{TV}(\varepsilon,\alpha)} \sum_{i=1}^{n} r_i L_m(x_i, y_i)\right].$$

Then we can use E_j to write

$$\mathfrak{R}_{\mathcal{S}}\big(L_{\mathcal{M}_{TV}(\varepsilon,\alpha)}\big) = \frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{m\in\mathcal{M}_{TV}(\varepsilon,\alpha)} \sum_{j=1}^{N}\sum_{i\in E_j} r_i L_m(x_i, y_i)\right].$$

Furthermore for any $m \in \mathcal{M}_{TV}(\varepsilon, \alpha)$ and $i \in E_j$, there exists $\alpha_i \in [-\alpha, \alpha]$ such that: $L_m(x_i, y_i) = L_m(c_j, y_i) + \alpha_i$. Then we have

$$\begin{aligned}\mathfrak{R}_{\mathcal{S}}\big(L_{\mathcal{M}_{TV}(\varepsilon,\alpha)}\big) \le{} & \frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{m\in\mathcal{M}_{TV}(\varepsilon,\alpha)} \sum_{j=1}^{N}\sum_{i\in E_j} r_i L_m(c_j, y_i)\right] \\ & + \frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{\alpha_i\in[-\alpha,\alpha]} \sum_{j=1}^{N}\sum_{i\in E_j} r_i\alpha_i\right].\end{aligned}$$

Let us start by studying the second term. We have

$$\frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{\alpha_i\in[-\alpha,\alpha]} \sum_{j=1}^{N}\sum_{i\in E_j} r_i\alpha_i\right] = \frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{\alpha_i\in[-\alpha,\alpha]} \sum_{i=1}^{n} r_i\alpha_i\right] = \frac{1}{n}\sum_{i=1}^{n}\alpha = \alpha.$$

Now looking at the first term. Since $L_m(x, y) \in [0, 1]$ for all (x, y) we have

$$\begin{aligned}& \frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{m\in\mathcal{M}_{TV}(\varepsilon,\alpha)} \sum_{j=1}^{N}\sum_{i\in E_j} r_i L_m(c_j, y_i)\right] \\ &= \frac{1}{n}\mathbb{E}_{r_i}\left[\sup_{m\in\mathcal{M}_{TV}(\varepsilon,\alpha)} \sum_{j=1}^{N}\sum_{y=1}^{K} L_m(c_j, y) \sum_{i\in E_{y,j}} r_i\right] \\ &\le \frac{1}{n}\mathbb{E}_{r_i}\left[\sum_{j=1}^{N}\sum_{y=1}^{K}\left|\sum_{i\in E_{y,j}} r_i\right|\right].\end{aligned}$$

Finally using the Khintchine inequality and the Cauchy Schartz inequality we get

$$
\begin{aligned}
\frac{1}{n}\mathbb{E}_{r_i}\left[\sum_{j=1}^{N}\sum_{y=1}^{K}\left|\sum_{i\in E_{y,j}} r_i\right|\right] &\leq \frac{1}{n}\sum_{j=1}^{N}\sum_{y=1}^{K}\sqrt{|E_{y,j}|} \quad \text{(Khintchine)}\\
&\leq \frac{1}{n}\sqrt{N\times K}\sqrt{\sum_{j=1}^{N}\sum_{y=1}^{K}|E_{y,j}|} \quad \text{(Cauchy)}\\
&= \sqrt{\frac{N\times K}{n}}.
\end{aligned}
$$

By combining the upper-bounds we have for each term, we get the expected result,

$$
\mathfrak{R}_{\mathcal{S}}\left(L_{\mathcal{M}_{TV}(\varepsilon,\alpha)}\right) \leq \sqrt{\frac{N\times K}{n}} + \alpha.
$$

□

The above result means that, if we can cover the n training samples with $O(1)$ balls, then we can bound the generalization gap of any randomized classifier $m \in \mathcal{M}_{TV}(\varepsilon,\alpha)$ by $O\left(\frac{1}{\sqrt{n}}\right) + \alpha$. Furthermore, a natural corollary of Theorem 19 bounds the Rademacher complexity of the class $L_{\mathcal{M}_\beta(\varepsilon,\alpha)}$.

Corollary 4. *Let $L_{\mathcal{M}_\beta(\varepsilon,\alpha)}$ be the loss function class associated with $\mathcal{M}_\beta(\varepsilon,\alpha)$. Then, for any $\mathcal{S} := \{(x_1,y_1),\ldots,(x_n,y_n)\}$, the following holds,*

$$
\mathfrak{R}_{\mathcal{S}}\left(L_{\mathcal{M}_\beta(\varepsilon,\alpha)}\right) \leq \sqrt{\frac{N\times K}{n}} + \min\left(\frac{3}{2}\left(\sqrt{1+\frac{4\alpha}{9}}-1\right)^{1/2}, \frac{e^{\alpha+1}-1}{e^{\alpha+1}+1}\right).
$$

Where $N = N\left(\{x_1,\ldots,x_n\}, \|.\|_p, \varepsilon\right)$ is the ε-external covering number of the inputs $\{x_1,\ldots,x_n\}$ for the ℓ_p norm.

Proof. This corollary is an immediate consequence of Theorem 19 and Proposition 18. □

Thanks to Theorems 18 and 19 and Corollary 4, one can easily bound the generalization gap of robust randomized classifiers.

A.5.2 Discussion and dimensionality issues

Xu and Mannor [2012] previously studied generalization bounds for learning algorithms based on their robustness. Although we use very different proof techniques, their results and ours are similar. More precisely, both analyses conclude that robust models generalize well if the training samples have a small covering number. Note, however, that we base our formulation on an *adaptive partition* of the samples, while the initial paper from Xu and Mannor [2012] only focuses on a fixed partition of the input space. Wre refer the reader to the discussion section in [Xu and Mannor, 2012] for more details.

These findings seem to contradict the current line of works on the hardness of generalization in the adversarial setting. In fact, if the ground truth distribution is sufficiently concentrated (*e.g.* lies in a low dimensional subspace of x), a small number of balls can cover $\mathcal{S}$ with high probability; hence $N = O(1)$. This means that we can learn robust classifiers with the same sample complexity as in the standard setting. But if the ground truth distribution is not concentrated enough, the training samples will be far one from another; hence forcing the covering number to be large. In the worse case scenario, we need to cover the whole space $[0,1]^d$ giving a covering number $N = O\left(\frac{1}{(\varepsilon)^d}\right)$ which is exponential in the dimension of the problem.

Therefore, in the worst-case scenario, our bound is in $O\left(\frac{1}{(\varepsilon)^d\sqrt{n}}\right) + \alpha$. When ε is small and the dimension of the problem is high, this bound is too large to give any meaningful insight on the generalization gap of the problem. Therefore, we still need to tighten our analysis to show that robust learning for randomized classifiers is possible in high dimensional spaces.

Remark 11. *Note that, we provided a very general result for randomized classifiers under the only assumption that they are robust w.r.t. the total variation distance. Our result applies to any class of classifiers and not only linear classifiers or one-hidden layer neural networks. To build a finer analysis, and to evade the curse of dimensionality, we should consider designing specific sub-classes $\mathcal{M} \subset \mathcal{M}_{TV}(\varepsilon, \alpha)$ and adapt the proofs to make the term N smaller in the worst-case scenario.*

A.6 Building robust randomized classifiers

In this section we present a simple yet efficient way to transform a non-robust, non-randomized classifier into a robust randomized classifier. To do so, we use a key property of both the Renyi divergence and the total variation distance called the *Data processing inequality*. It is a well-known result from information theory which states that *"post-processing cannot increase information"*. The data processing inequality is as follows.

Theorem 20 (Cover and Thomas [2012])**.** *Let us consider two arbitrary spaces $\mathcal{Z}, \mathcal{Z}'$, $\rho, \rho' \in \mathcal{M}_1^+(\mathcal{Z})$ and $D \in \{D_{TV}, D_\beta\}$. Then for any $\psi : \mathcal{Z} \to \mathcal{Z}'$ we have*

$$D\big(\psi\#\rho, \psi\#\rho'\big) \leq D\big(\rho, \rho'\big),$$

where $\psi\#\rho$ denotes the pushforward of distiburtion ρ by ψ.

In the context of robustness to adversarial examples, we use the data processing inequality to ease the design of robust randomized classifiers. In particular, let us suppose that we can build a randomized pre-processing $\mathfrak{p} : \mathcal{X} \to \mathcal{M}_1^+(\mathcal{X})$ such that for any $x \in \mathcal{X}$ and any ε-bounded perturbation τ, we have

$$D(\mathfrak{p}(x), \mathfrak{p}(x+\tau)) \leq \alpha, \text{ with } D \in \{D_{TV}, D_\beta\}. \tag{A.15}$$

Then, thanks to the data processing inequality, we can take any deterministic classifier h to build an (ε, α) robust classifier w.r.t D defined as $m : x \mapsto h\#\mathfrak{p}(x)$. This considerably simplifies the problem of building a class of robust models. Therefore, we want to build $\mathfrak{p}$ a randomized pre-processing for which we can control the Renyi divergence and/or total variation distance between two inputs. To do this, we analyze the simple procedure of injecting random noise directly on the image before sending it to a classifier. Since the Renyi divergence and the total variation distances are particularly well suited to the study of Gaussian distributions, we first use this type of noise injection. More precisely, in this section, we focus on a mapping that writes as follows.

$$\mathfrak{p} : x \mapsto \mathcal{N}(x, \Sigma), \tag{A.16}$$

for some given non-degenerate covariance matrix $\Sigma \in \mathcal{M}_{d\times d}(\mathbb{R})$. We refer the interested reader to Pinot et al. [2019] for more general classes of noise, namely exponential families. Let us now evaluate the maximal variation of Gaussian pre-processing $\mathfrak{p}$ when applied to an image $x \in \mathcal{X}$ with and without perturbation.

Lemma 6. *Let $\beta > 1$, $x, \tau \in \mathcal{X}$ and $\Sigma \in \mathcal{M}_{d\times d}(\mathbb{R})$ a non-degenerate covariance matrix. Let $\rho = \mathcal{N}(x, \Sigma)$ and $\rho' = \mathcal{N}(x + \tau, \Sigma)$, then $D_\beta(\rho, \rho') = \frac{\beta}{2}\|\tau\|^2_{\Sigma^{-1}}$.*

Thanks to the above lemma, we know how to evaluate the level of Renyi-robustness that a Gaussian noise pre-processing brings to a classifier. Now that we have this result, thanks to Proposition 18, we can also upper-bound the total variation distance between $\mathcal{N}(x, \Sigma)$ and $\mathcal{N}(x + \tau, \Sigma)$. But this bound is not always tight. Besides, we can directly evaluate the total variation distance between two Gaussian distributions as follows.

Lemma 7. *Let $x, x' \in \mathcal{X}$ and $\Sigma \in \mathcal{M}_{d\times d}(\mathbb{R})$ a non-degenerate covariance matrix. Let $\rho = \mathcal{N}(x, \Sigma)$ and $\rho' = \mathcal{N}(x + \tau, \Sigma)$, then $D_{TV}(\rho, \rho') = 2\Phi\left(\frac{\|\tau\|_{\Sigma^{-1}}}{2}\right) - 1$ with Φ the cumulative density function of the standard Gaussian distribution.*

Note that both bounds increase with the Mahalanobis norm of τ. Furthermore, we see that the greater the entropy of the Gaussian noise we inject, the smaller the distance between distributions. If we simplify the covariance matrix by setting $\Sigma = \sigma^2 I_d$, it means that we can build more or less robust randomized classifiers against ℓ_2 adversaries, depending on σ.

Theorem 21 (Robustness of Gaussian pre-processing)**.** *Let us consider $c : \mathcal{X} \to \mathcal{Y}$ a deterministic classifier, $\sigma > 0$ and $\mathfrak{p} : x \mapsto \mathcal{N}(x, \sigma^2 I_d)$ a pre-processing probabilistic mapping. Then the randomized classifier $m := c\#\mathfrak{p}$ is*

- *$(\alpha_2, \frac{(\alpha_2)^2\beta}{2\sigma})$-robust w.r.t. D_β against ℓ_2 adversaries.*
- *$(\alpha_2,\ 2\Phi(\frac{\alpha_2}{2\sigma}) - 1)$-robust w.r.t. D_{TV} against ℓ_2 adversaries.*

Proof. Let $x, \tau \in \mathcal{X}$ such that $\|\tau\|_2 \le \alpha_2$. Thanks to Lemma 6 we have

$$D_\beta(\mathfrak{p}(x), \mathfrak{p}(x+\tau)) = \frac{\beta}{2}\|\tau\|^2_{\Sigma^{-1}} = \frac{\beta}{2\sigma^2}\|\tau\|_2^2 \le \frac{\beta(\alpha_2)^2}{2\sigma^2}.$$

Similarly, thanks to Lemma 7, we get

$$D_{TV}(\mathfrak{p}(x), \mathfrak{p}(x+\tau)) = 2\Phi\left(\frac{\|\tau\|_{\Sigma^{-1}}}{2}\right) - 1 \leq 2\Phi\left(\frac{\alpha_2}{2\sigma}\right) - 1.$$

Finally, from the data processing inequality, i.e., thm 20, we get both

$$D_\beta(m(x), m(x+\tau)) \leq \frac{\beta(\alpha_2)^2}{2\sigma^2},$$

and

$$D_{TV}(m(x), m(x+\tau)) \leq 2\Phi\left(\frac{\alpha_2}{2\sigma}\right) - 1.$$

The above inequalities conclude the proof. □

Theorem 21 means that we can build simple noise injection schemes as pre-processing of state-of-the-art image classification models and keep track of the maximal loss of accuracy under attack of the resulting randomized classifier. These results also highlight the profound link between randomized classifiers and randomized smoothing as presented by Cohen et al. [2019]. Even though our findings are of different nature, both techniques use the same base mechanism (Gaussian noise injection). Therefore, Gaussian pre-processing is a principled defense method that can be analyzed through several standpoints, including certified robustness and statistical learning theory.

A.7 Discussion: Mode preservation property and Randomized Smoothing

Even though randomized classifiers have some interesting properties regarding generalization error, we can also study them through the prism of deterministic robustness. Let us for example consider the classifier that outputs the class with the highest probability for $m(x)$, a.k.a. the mode of $m(x)$. It writes

$$h_{\text{rob}} : x \mapsto \operatorname*{argmax}_{k \in [K]} m(x)_k \tag{A.17}$$

Then checking whether h_{rob} is robust boils down to demonstrating that the mode of $m(x)$ does not change under perturbation. It turns out that D_{TV} robust classifiers have this property. We call it the mode preservation property of $\mathcal{M}_{TV}(\varepsilon, \alpha)$.

Proposition 19 (Mode preservation for D_{TV}-robust classifiers)**.** *Let $m \in \mathcal{M}_{TV}(\varepsilon, \alpha)$ be a robust randomized classifier and $x \in \mathcal{X}$ such that $m(x)_{(1)} \geq m(x)_{(2)} + 2\alpha$. Then, for any $\tau \in \mathcal{X}$, the following holds,*

$$\|\tau\|_p \leq \varepsilon \implies h_{rob}(x) = h_{rob}(x+\tau) \ .$$

Proof. Let $x, \tau \in \mathcal{X}$ such that $\|\tau\|_p \leq \varepsilon$ and $m \in \mathcal{M}_{TV}(\varepsilon, \alpha)$ such that

$$m(x)_{(1)} \geq m(x)_{(2)} + 2\alpha.$$

By definition of $\mathcal{M}_{TV}(\varepsilon, \alpha)$, we have that

$$D_{TV}(m(x), m(x+\tau)) \leq \alpha.$$

Then, for all $k \in \{1, \ldots, K\}$ we have

$$m(x)_k - \alpha \leq m(x+\tau)_k \leq m(x)_k + \alpha\,.$$

Let us denote k^* the index of the biggest value in $m(x)$, i.e., $m(x)_{k^*} = m(x)_{(1)}$. For any $k \in \{1, \ldots, K\}$ with $k \neq k^*$, we have $m(x)_{k^*} \geq m(x)_k + 2\alpha$. Finally, for any $k \neq k^*$, we get

$$m(x+\tau)_{k^*} \geq m(x)_{k^*} - \alpha \geq m(x)_k + \alpha \geq m(x+\tau)_k.$$

Then, $\underset{k\in[K]}{\operatorname{argmax}}\, m(x)_k = \underset{k\in[K]}{\operatorname{argmax}}\, m(x+\tau)_k$. This concludes the proof. □

Similarly, we can demonstrate a mode preservation property for robust classifiers w.r.t. the Renyi divergence.

Proposition 20 (Mode preservation for Renyi-robust classifiers). *Let $m \in \mathcal{M}_\beta(\varepsilon, \alpha)$ be a robust randomized classifier and $x \in \mathcal{X}$ such that*

$$\left(m(x)_{(1)}\right)^{\frac{\beta}{\beta-1}} \geq \exp\left((2 - \frac{1}{\beta})\alpha\right)\left(m(x)_{(2)}\right)^{\frac{\beta-1}{\beta}}.$$

Then, for any $\tau \in \mathcal{X}$, the following holds,

$$\|\tau\|_p \leq \varepsilon \implies h_{rob}(x) = h_{rob}(x+\tau),$$

where $h_{rob}(x) := \underset{k\in[K]}{\operatorname{argmax}}\, m(x)_k$.

Proof. Let $x, \tau \in \mathcal{X}$ such that $\|\tau\|_p \leq \varepsilon$ and $m \in \mathcal{M}_\beta(\varepsilon, \alpha)$ such that

$$\left(m(x)_{(1)}\right)^{\frac{\beta}{\beta-1}} \geq \exp\left((2 - \frac{1}{\beta})\alpha\right)\left(m(x)_{(2)}\right)^{\frac{\beta-1}{\beta}}.$$

Then by definition of $\mathcal{M}_\beta(\varepsilon, \alpha)$, we have

$$D_\beta(m(x), m(x+\tau)) \leq \alpha.$$

Furthermore, by using Proposition 17, for any $k \in \{1, \ldots, K\}$ we have

$$(*) m(x)_k \leq (\exp(\alpha) m(x+\tau)_k)^{\frac{\beta-1}{\beta}} \text{ and } (**) m(x+\tau)_k \leq (\exp(\alpha) m(x)_k)^{\frac{\beta-1}{\beta}}.$$

Let us denote k^* the index such that $m(x)_{k^*} = m(x)_{(1)}$. Then using $(*)$ we get

$$m(x+\tau)_{k^*} \geq \exp(-\alpha)(m(x)_{k^*})^{\frac{\beta}{\beta-1}}.$$

Furthermore for any $k \in \{1, \ldots, K\}$ where $k \neq k^*$, we can use the assumption we made on m to get

$$\exp(-\alpha)(m(x)_{k^*})^{\frac{\beta}{\beta-1}} \geq \exp\left(\frac{\beta-1}{\beta}\alpha\right)(m(x)_k)^{\frac{\beta-1}{\beta}}.$$

Finally, using $(**)$ we have

$$\exp\left(\frac{\beta-1}{\beta}\alpha\right)(m(x)_k)^{\frac{\beta-1}{\beta}} \geq m(x+\tau)_k.$$

The above gives us $\operatorname*{argmax}_{k\in[K]} m(x)_k = \operatorname*{argmax}_{k\in[K]} m(x+\tau)_k$. This concludes the proof. ☐

Coming back to the decomposition in Equation (A.5), with the above result, we can bound the risk the adversary induces with non-zero perturbations by the mass of points on which the classifier h_{rob} gives the good response but based on a low probability of success, i.e., with small confidence

$$\mathcal{R}_{\varepsilon}^{>0}(m) \leq \mathbb{P}_{(x,y)\sim\mathbb{P}}\left[h_{\text{rob}}(x) = y \textit{ and } m(x)_{(1)} < m(x)_{(2)} + 2\alpha\right]. \tag{A.18}$$

This means that the only points on which the adversary may induce misclassification are the points on which m already has a high risk. Once more, this says something fundamental about the behavior of robust randomized classifiers. On undefended models, the adversary could change the decision on any point it wanted; now it is limited to changing points on which the classifier is already inaccurate. This considerably mitigates the threat model we should consider. Furthermore, for any deterministic classifier designed as in Equation (A.17), we can also bound the maximal loss of accuracy under attack the classifier may suffer. This bound may, however, be harder to evaluate since it now depends on both the classifier and the dataset distribution. The classifier we define in Equation (A.17) and the mode preservation property of m are closely related to provable defenses based on randomized smoothing. The core idea of randomized smoothing is to take a hypothesis h and to build a robust classifier that writes

$$c_{rob} : x \mapsto \operatorname*{argmax}_{k\in[K]} \mathbb{P}_{z\sim\mathcal{N}(0,\sigma^2 I)}[h(x+z) = k]. \tag{A.19}$$

From a probabilistic point of view, for any input x, randomized smoothing amounts to output the most probable class of the probability measure $m(x) := h\#\mathcal{N}(x, \sigma^2 I)$. Hence, randomized smoothing uses the mode preservation property of m to build a provably robust (deterministic) classifier. Therefore, the above results (Proposition 19 and Equation A.18) also hold for provable defenses based on randomized smoothing. Studying randomized smoothing from our point of view could give an interesting new perspective on that method. So far no results have been published on the generalisation gap of this defense in the adversarial setting. We could devise generalization bounds by similarity with our analysis. Furthermore, the probabilistic interpretation stresses that randomized smoothing is somewhat restrictive since it only considers probability measures which are the expectation on a simple noise injection scheme. The mode preservation property explains the behavior of randomized smoothing, but also presents fundamental properties of randomized defenses that could be used to construct more general defense schemes.

A.8 Numerical validations against ℓ_2 adversary

To illustrate our findings, we train randomized neural networks with Gaussian pre-processing during training and inference on CIFAR-10 and CIFAR-100. Based on this randomized classifier, we study the impact of randomization on the standard accuracy of the network, and observe the theoretical trade-off between accuracy and robustness.

A.8.1 Architecture and training procedure

All the neural networks we use in this section are WideResNets [Zagoruyko and Komodakis, 2016] with 28 layers, a widen factor of 10, a dropout factor of 0.3 and LeakyRelu activation with a 0.1 slope. To train an undefended standard classifier we use the following hyper-parameters[2].

- *Number of Epochs:* 200
- *Batch size:* 400
- *Loss function:* Cross Entropy Loss
- *Optimizer :* Stochastic gradient descent algorithm with momentum 0.9, weight decay of 2×10^{-4} and a learning rate that decreases during the training as follows:

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

To transform these standard networks into randomized classifiers, we inject noise drawn from Gaussian distributions, each with various standard deviations directly on the image before passing it through the network. Both during training and test, for computational efficiency, we evaluate the performance of the the algorithm over a single run for every images; hence no Monte Carlo

[2]Reusable code can be found in the following repository: `https://github.com/MILES-PSL/Adversarial-Robustness-Through-Randomization`

estimator is used. However, in practice, the test-time accuracy is stable when evaluated over the entire test dataset.

A.8.2 Results

Figures A.1 and A.2 show the accuracy and the minimum level of accuracy under attack of our randomized neural network for several levels of injected noise. We can see (Figure A.1) that the precision decreases as the noise intensity grows. In that sense, the noise must be calibrated to preserve both accuracy and robustness against adversarial attacks. This is to be expected, because the greater the entropy of the classifier, the less precise it gets.

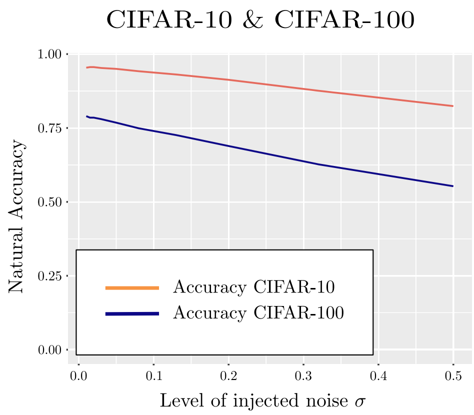


Figure A.1: Impact of the standard deviation of the Gausian noise on accuracy in a randomized model on CIFAR-10 and CIFAR-100 dataset.

Furthermore, when injecting Gaussian noise as a defense mechanism, the resulting randomized network m is both $(\alpha_2, \frac{(\alpha_2)^2}{2\sigma})$-robust w.r.t. D_1 and $(\alpha_2, 2\Phi(\frac{\alpha_2}{2\sigma}) - 1)$-robust w.r.t. D_{TV} against ℓ_2 adversaries. Therefore thanks to thms 15 and 17 we have that

$$\mathcal{R}_\varepsilon(m) - \mathcal{R}(m) \leq 2\Phi\left(\frac{\alpha_2}{2\sigma}\right) - 1, \text{ and} \tag{A.20}$$

$$\mathcal{R}_\varepsilon(m) - \mathcal{R}(m) \leq 1 - e^{-\frac{(\alpha_2)^2}{2\sigma}} \mathbb{E}_{x\sim\mathcal{D}_{|\mathcal{X}}}\left[e^{-H(m(x))}\right]. \tag{A.21}$$

Figure A.2 illustrates the theoretical lower bound on accuracy under attack (based on the minimum gap between Equations (A.20) and (A.21)) for different standard deviations. The term in entropy has been estimated using a Monte Carlo method with 10^4 simulations. The trade-off between accuracy and robustness appears with respect to the noise intensity. With small noises, the accuracy is high, but the guaranteed accuracy drops fast with respect to the magnitude of the adversarial perturbation. Conversely, with bigger noises, the accuracy is lower but decreases slowly with respect to the magnitude of the adversarial perturbation. Overall, we get strong accuracy guarantees against small adversarial perturbations, but when the perturbation is bigger than 0.5 on CIFAR-10 (resp. 0.3 on CIFAR-100, the guarantees are still not sufficient).

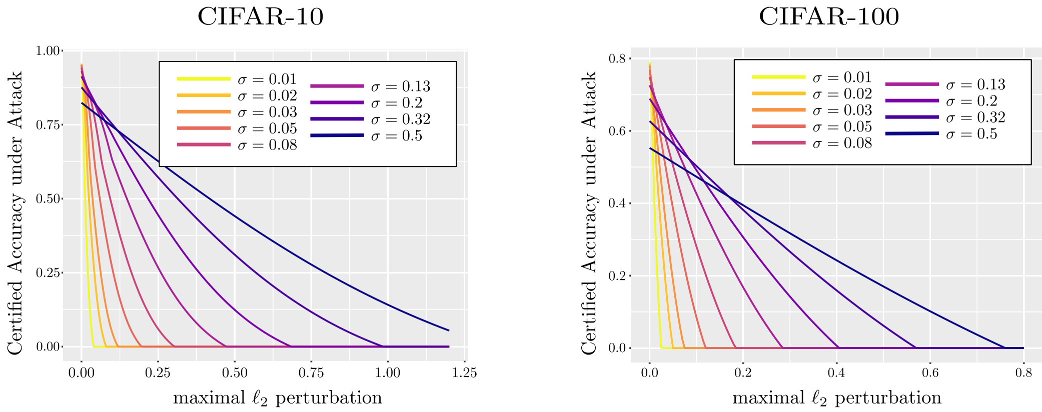


Figure A.2: Guaranteed accuracy of different randomized models with Gaussian noise given the ℓ_2 norm of the adversarial perturbations.

A.9 Lesson learned and future work

This paper brings new contributions to the theory of robustness to adversarial attacks. We provided an in depth analysis of randomized classifier, demonstrating their interest to defend against adversarial attacks. We first defined a notion of robustness for randomized classifiers using probability metrics/divergences, namely the total variation distance and the Renyi divergence. Second, we demonstrated that when a randomized classifier complies with this definition of robustness, we can bound their loss of accuracy under attack. We also studied the generalization properties of this class of functions and gave results indicating that robust randomized classifiers can generalize. Finally, we showed that randomized classifiers have a mode preservation property. This presents a fundamental property of randomized defenses that can be used to explain randomized smoothing from a probabilistic point of view. To support our theoretical findings we presented a simple yet efficient scheme for building robust randomized classifiers. We show that Gaussian noise injection can provide principled robustness against ℓ_2 adversarial attacks. We ran a set of experiments on CIFAR-10 and CIFAR-100 using Gaussian noise injection with advanced neural network architectures to build accurate models with controlled loss of accuracy under attack.

Future work will focus on studying the combination of randomization with more sophisticated defenses and on devising new tight bounds on the adversarial generalization and the adversarial risk gap of randomized classifiers. Based on the connections we established we randomized smoothing in Section A.7, we will also aim at devising bounds on the gap between the standard and adversarial risks for this defense. Another interesting direction would be to show that the classifiers based on randomized smoothing have a generalization gap similar to the classes of randomized classifiers we studied.

A.10 Appendix: Proof of technical Lemmas

A.10.1 Proof of Lemma 6

Proof. Let $\beta > 1$. Let us denote g and g' respectively the probability density functions of ρ and ρ' with respect to the Lebesgue measure. We also set $x' = x + \tau$ for readability. Then we have

$$D_\beta(\rho, \rho') = \frac{1}{\beta - 1} \log \mathbb{E}_{z \sim \rho'} \left[\left(\frac{g(z)}{g'(z)} \right)^\beta \right]$$
$$= \frac{1}{\beta - 1} \log \mathbb{E}_{z \sim \rho'} \left[\exp \left(\frac{\beta}{2} \left((z - x')^\intercal \Sigma^{-1} (z - x') - (z - x)^\intercal \Sigma^{-1} (z - x) \right) \right) \right].$$

By change of variable we get

$$= \frac{1}{\beta - 1} \log \mathbb{E}_{z \sim \mathcal{N}(0, \Sigma)} \left[\exp \left(\frac{\beta}{2} \left(z^\intercal \Sigma^{-1} z - (z + \tau)^\intercal \Sigma^{-1} (z + \tau) \right) \right) \right]$$
$$= \frac{1}{\beta - 1} \log \mathbb{E}_{z \sim \mathcal{N}(0, \Sigma)} \left[\exp \left(\frac{\beta}{2} \left(-2 z^\intercal \Sigma^{-1} \tau - \|\tau\|_{\Sigma^{-1}}^2 \right) \right) \right]$$
$$= \frac{1}{\beta - 1} \log \int_{\mathbb{R}^d} \frac{\exp\left(-\frac{1}{2} z^\intercal \Sigma^{-1} z - \frac{\beta}{2} 2 z^\intercal \Sigma^{-1} \tau - \frac{\beta}{2} \|\tau\|_{\Sigma^{-1}}^2 \right)}{(2\pi)^d \det(\Sigma)^{d/2}} dz\,.$$

Furthermore, for any $z \in \mathbb{R}^d$, we have

$$- \frac{1}{2} z^\intercal \Sigma^{-1} z - \frac{\beta}{2} 2 z^\intercal \Sigma^{-1} \tau - \frac{\beta}{2} \|\tau\|_{\Sigma^{-1}}^2$$
$$= - \frac{1}{2} (z + \beta\tau)^\intercal \Sigma^{-1} (z + \beta\tau) + \frac{\beta^2 - \beta}{2} \|\tau\|_{\Sigma^{-1}}^2\,.$$

Then we can re-write the Renyi divergence as follows

$$D_\beta(\rho, \rho') = \frac{1}{\beta - 1} \log \mathbb{E}_{z \sim \mathcal{N}(-\beta\tau, \Sigma)} \left[\exp \left(\frac{\beta^2 - \beta}{2} \|\tau\|_{\Sigma^{-1}}^2 \right) \right]$$
$$= \frac{1}{\beta - 1} \log \left(\exp \left(\frac{\beta^2 - \beta}{2} \|\tau\|_{\Sigma^{-1}}^2 \right) \right)$$
$$= \frac{\beta}{2} \|\tau\|_{\Sigma^{-1}}^2\,.$$

This concludes the proof. □

A.10.2 Proof of Lemma 7

Proof. Let us denote g and g' respectively the probability density functions of ρ and ρ' with respect to the Lebesgue measure. Furthermore, we denote $x' = x + \tau$. Then by definition of the total variation distance, we have $D_{TV}(\rho, \rho) = \rho(Z) - \rho'(Z)$ with $Z = \{z \mid g(z) \geq g'(z)\}$. In our case $g(z) \geq g'(z)$ is equivalent to

$$(z - x')^\intercal \Sigma^{-1}(z - x') - (z - x)^\intercal \Sigma^{-1}(z - x) \geq 0.$$

Then with the same simplification as above, we have

$$\begin{aligned}
\rho(Z) &= \mathbb{P}_{z\sim\mathcal{N}(x,\Sigma)}\big((z - x')^\intercal \Sigma^{-1}(z - x') - (z - x)^\intercal \Sigma^{-1}(z - x) \geq 0\big) \\
&= \mathbb{P}_{z\sim\mathcal{N}(0,\Sigma)}\big((z - \tau)^\intercal \Sigma^{-1}(z - \tau) - z^\intercal \Sigma^{-1} z \geq 0\big) \\
&= \mathbb{P}_{z\sim\mathcal{N}(0,\Sigma)}\big(-2z^\intercal \Sigma^{-1}\tau + \|\tau\|^2_{\Sigma^{-1}} \geq 0\big) \\
&= \mathbb{P}_{z\sim\mathcal{N}(0,I_d)}\left(z^\intercal \Sigma^{-1/2}\tau \leq \frac{1}{2}\|\tau\|^2_{\Sigma^{-1}}\right).
\end{aligned}$$

Furthermore, if $z \sim \mathcal{N}(0, I_d)$ then $z^\intercal \Sigma^{-1/2}\tau \sim \mathcal{N}(0, \|\tau\|^2_{\Sigma^{-1}})$; hence we also have $\frac{z^\intercal \Sigma^{-1/2}\tau}{\|\tau\|_{\Sigma^{-1}}} \sim \mathcal{N}(0, 1)$. Accordingly we get

$$\rho(Z) = \mathbb{P}_{z\sim\mathcal{N}(0,1)}\left(z \leq \frac{1}{2}\|\tau\|_{\Sigma^{-1}}\right) = \Phi\left(\frac{1}{2}\|\tau\|_{\Sigma^{-1}}\right).$$

By symmetry we get that $\rho'(A) = 1 - \rho(A) = 1 - \Phi\big(\frac{1}{2}\|\tau\|_{\Sigma^{-1}}\big)$. We then get

$$D_{TV}(\mu, \nu) = 2\Phi\left(\frac{\|\tau\|_{\Sigma^{-1}}}{2}\right) - 1$$

which concludes the proof. □

A.11 Discussion on probability metrics

As mentioned earlier in this paper, the choice of the metric/divergence is crucial as it characterizes the notion of adversarial robustness we are examining. We focus on the total variation distance and Renyi divergence, but the question of whether these metrics/divergences are more appropriate than others remains open. It should be noted, however, that our definition of robustness is monotonous depending on the metric/divergence we use.

Proposition 21 (Monotonicity of the robustness)**.** *Let m be a randomized classifier, and let D and D' be two divergences/metrics on $\mathcal{M}_1^+(\mathcal{Y})$. If there exists a non decreasing function $f : \mathbb{R} \mapsto \mathbb{R}$ such that $\forall \rho, \rho' \in \mathcal{M}_1^+(\mathcal{Y})$, $D(\rho, \rho') \leq f(D'(\rho, \rho'))$, then the following assertion holds.*

$$\mathrm{m} \textit{ is } (\varepsilon, \alpha)\text{-robust w.r.t. } D' \implies \mathrm{m} \textit{ is } (\varepsilon, f(\alpha))\textit{-robust w.r.t. } D.$$

The proof straightforwardly comes from the definition of robustness.

Proof. Let us consider m a randomized classifier (ε, α)-robust w.r.t. D'. Then for any $x \sim \mathbb{P}$, and $\tau \mid \|\tau\|_p \leq \varepsilon$, since f is non decreasing, we have

$$D(m(x), m(x+\tau)) \leq f\big(D'(m(x), m(x+\tau))\big) \leq f(\alpha).$$

Then m is $(\varepsilon, f(\alpha))$-robust w.r.t. D which concludes the proof. $\square$

The above result suggests that the different notions of robustness we might conceive are more related than they appear. Here are some of the most classical divergences used in machine learning. Let ρ, ρ', ν three measures in $\mathcal{M}_1^+(\mathcal{Y})$. We denotes g and g' the probability density functions of ρ and ρ' with respect to ν. Then we can define the *Wasserstein distance* as follows

$$D_W(\rho, \rho') := \inf \int_{\mathcal{Y}^2} \operatorname{dist}\big(y, y'\big) d\pi(y, y'), \tag{A.22}$$

where dist is some ground distance on $\mathcal{Y}$, and the infimum is taken over all joint distributions π in $\mathcal{M}_1^+(\mathcal{Y} \times \mathcal{Y})$ with marginals ρ and ρ'.

Remark 12. *In transportation theory, the Wasserstein distance is solution of the Monge-Kantorovich problem with the cost function* $c(y, y') = \operatorname{dist}(y, y')$. *Then, the definitions of total variation and Wasserstein distance match when we use the trivial distance* $\operatorname{dist}(y, y') = \mathbb{1}\{y \neq y'\}$.

We also define respectively the *Hellinger distance* and the *Separation distance* as follows.

$$D_H(\rho, \rho') := \left[\int_{\mathcal{Y}} \left(\sqrt{g} - \sqrt{g'}\right)^2 d\nu\right]^{1/2}. \tag{A.23}$$

$$D_S(\rho, \rho') := \sup_{y \in \mathcal{Y}} \left(1 - \frac{g(y)}{g'(y)}\right). \tag{A.24}$$

If we take any of the above metrics/divergences to instantiate a notion of adversarial robustness we might get very different semantics for them. However, we can show that any of these definitions can be covered – with respect to Proposition 21 – either by the Renyi or the total variation robustness. Figure A.3 summarizes the links we can make between all these different definitions of robustness, and Propositions 22 and 23 present the associated results. We can see that the total variation distance and the Renyi divergence are both central since they can cover any of the other robustness notions. This does not mean that they are more appropriate than the others, but at least they are general enough to cover a wide range of possible definitions.

Proposition 22. *Let m be a randomized classifier. If m is (ε, α)-robust w.r.t. D_{TV} then the following assertions hold.*

- *m is $(\varepsilon, \alpha \times Diam(\mathcal{Y}))$-robust w.r.t. D_W, where $Diam(\mathcal{Y}) := \max_{y, y' \in \mathcal{Y}} \operatorname{dist}(y, y')$.*
- *m is $\left(\varepsilon, \sqrt{2\alpha}\right)$-robust w.r.t. D_H.*

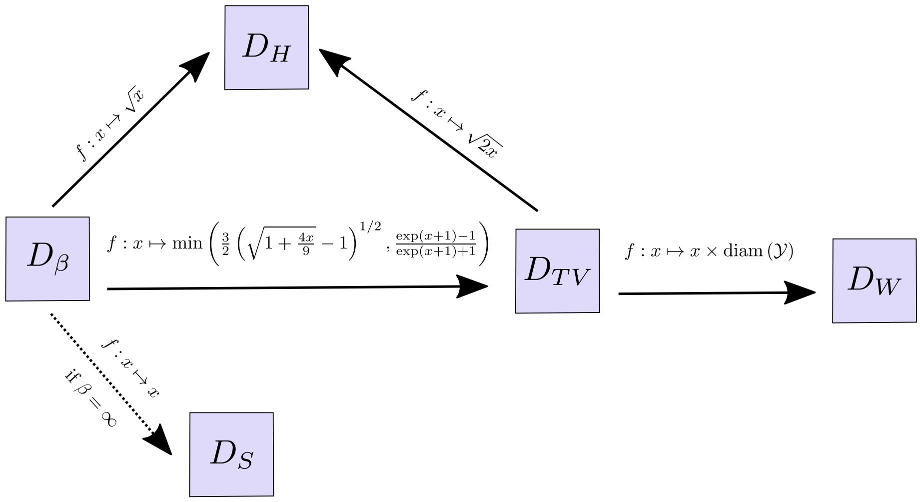


Figure A.3: Summary of the relations between the different robustness notions from Propositions 22 and 23.

Proof. Let us consider ρ and $\rho' \in \mathcal{M}_1^+(\mathcal{Y})$. Thanks to Gibbs and Su [2002] we have

- $D_W(\rho,\rho') \leq Diam(\mathcal{Y})D_{TV}(\rho,\rho')$.
- $D_H(\rho,\rho') \leq \sqrt{2D_{TV}(\rho,\rho')}$.

Hence, by using Proposition 21 respectively with $f : x \mapsto Diam(\mathcal{Y})x$ and $f : x \mapsto \sqrt{2x}$ we get the expected results. $\square$

Proposition 23. *Let m be a randomized classifier. If m is (ε,α)-robust w.r.t. D_β then the following assertions hold.*

- *m is (ε,α')-robust w.r.t. D_{TV} with* $\alpha' = \min\left(\frac{3}{2}\left(\sqrt{1+\frac{4\alpha}{9}}-1\right)^{1/2}, \frac{\exp(\alpha+1)-1}{\exp(\alpha+1)+1}\right)$.
- *m is $(\varepsilon,\sqrt{\alpha})$-robust w.r.t. D_H.*
- *If $\beta=\infty$, then m is (ε,α) robust w.r.t. D_S.*

Proof. 1) First, let us suppose that $\beta \geq 1$. Thanks to Proposition 18 and to [Gibbs and Su, 2002], for any $\rho,\rho' \in \mathcal{M}_1^+(\mathcal{Y})$ we have

- $D_H(\rho,\rho') \leq \sqrt{D_1(\rho,\rho')} \leq \sqrt{D_\beta(\rho,\rho')}$ (see Gibbs and Su [2002]).
- $D_{TV}(\rho,\rho') \leq \min\left(\frac{3}{2}\left(\sqrt{1+\frac{4D_\beta(\rho,\rho')}{9}}-1\right)^{1/2}, \frac{\exp\left(D_\beta(\rho,\rho')+1\right)-1}{\exp\left(D_\beta(\rho,\rho')+1\right)+1}\right)$ (Prop. 18).

Hence, by using Proposition 21, as above, we get the expected results.
2) Now let us suppose that $\beta = \infty$. By definition of the supremum divergence, we have

$$D_\infty(\rho, \rho') = \sup_{B \subset \mathcal{Y}} \left| \ln \frac{\rho(B)}{\rho'(B)} \right|.$$

Furthermore, note that the function $x \mapsto 1 - x - |\ln(x)|$ is negative on $\mathbb{R}$, therefore for any $y \in \mathcal{Y}$ one has

$$1 - \frac{\rho(y)}{\rho'(y)} \leq \left| \ln \frac{\rho(y)}{\rho'(y)} \right|.$$

Since the above inequality is true for any $y \in \mathcal{Y}$, we have

$$D_S(\rho, \rho') = \sup_{y \in \mathcal{Y}} \left(1 - \frac{\rho(y)}{\rho'(y)} \right) \leq \sup_{y \in \mathcal{Y}} \left| \ln \frac{\rho(y)}{\rho'(y)} \right| \leq \sup_{B \subset \mathcal{Y}} \left| \ln \frac{\rho(B)}{\rho'(B)} \right| = D_\infty(\rho, \rho').$$

Finally, by using Proposition 21 with $f : x \mapsto x$ we get the expected results. $\square$

B Black-box adversarial attacks: tiling and evolution strategies

We introduce a new black-box attack achieving state of the art performances. Our approach is based on a new objective function, borrowing ideas from ℓ_∞-white box attacks, and particularly designed to fit derivative-free optimization requirements. It only requires to have access to the logits of the classifier without any other information which is a more realistic scenario. Not only we introduce a new objective function, we extend previous works on black box adversarial attacks to a larger spectrum of evolution strategies and other derivative-free optimization methods. We also highlight a new intriguing property that deep neural networks are not robust to single shot tiled attacks. Our models achieve, with a budget limited to $10,000$ queries, results up to 99.2% of success rate against InceptionV3 classifier with 630 queries to the network on average in the untargeted attacks setting, which is an improvement by 90 queries of the current state of the art. In the targeted setting, we are able to reach, with a limited budget of $100,000$, 100% of success rate with a budget of $6,662$ queries on average, i.e. we need 800 queries less than the current state of the art.

B.1 Introduction

Despite their success, deep learning algorithms have shown vulnerability to adversarial attacks [Biggio et al., 2013, Szegedy et al., 2014], *i.e.* small imperceptible perturbations of the inputs, that lead the networks to misclassify the generated adversarial examples. Since their discovery, adversarial attacks and defenses have become one of the hottest research topics in the machine learning community as serious security issues are raised in many critical fields. They also question our understanding of deep learning behaviors. Although some advances have been made to explain theoretically [Fawzi et al., 2016, Sinha et al., 2017, Cohen et al., 2019, Pinot et al., 2019] and experimentally [Goodfellow et al., 2015b, Xie et al., 2018, Meng and Chen, 2017, Samangouei et al., 2018, Araujo et al., 2019] adversarial attacks, the phenomenon remains misunderstood and there is still a gap to come up with principled guarantees on the robustness of neural networks against maliciously crafted attacks. Designing new and stronger attacks helps building better defenses, hence the motivation of our work.

First attacks were generated in a setting where the attacker knows all the information of the network (architecture and parameters). In this *white box* setting, the main idea is to perturb the input in the direction of the gradient of the loss w.r.t. the input [Goodfellow et al., 2015b, Kurakin et al., 2016, Carlini and Wagner, 2017, Moosavi-Dezfooli et al., 2016]. This case is unrealistic because the attacker has only limited access to the network in practice. For instance, web services that propose commercial recognition systems such as Amazon or Google are backed by pretrained neural networks. A user can *query* this system by sending an image to classify. For such a query, the user only has access to the inference results of the classifier which might be either the label, probabilities or logits. Such a setting is coined in the literature as the *black box* setting. It is more realistic but also more challenging from the attacker's standpoint.

As a consequence, several works proposed black box attacks by just querying the inference results of a given classifier. A natural way consists in exploiting the transferability of an adversarial attack, based on the idea that if an example fools a classifier, it is more likely that it fools another one [Papernot et al., 2016a]. In this case, a white box attack is crafted on a fully known classifier. Papernot et al. [2017a] exploited this property to derive practical black box attacks. Another approach within the black box setting consists in estimating the gradient of the loss by querying the classifier [Chen et al., 2017, Ilyas et al., 2018a,b]. For these attacks, the PGD attack [Kurakin et al., 2016, Madry et al., 2018b] algorithm is used and the gradient is replaced by its estimation.

In this paper, we propose efficient black box adversarial attacks using stochastic derivative free optimization (DFO) methods with only access to the logits of the classifier. By efficient, we mean that our model requires a limited number of queries while outperforming the state of the art in terms of attack success rate. At the very core of our approach is a new objective function particularly designed to suit classical derivative free optimization. We also highlight a new intriguing property that deep neural networks are not robust to single shot tiled attacks. It leverages results and ideas from ℓ_∞-attacks. We also explore a large spectrum of evolution strategies and other derivative-free optimization methods thanks to the Nevergrad framework [Rapin and Teytaud, 2018].

Outline of the paper. We present in Section B.2 the related work on adversarial attacks. Section B.3 presents the core of our approach. We introduce a new generic objective function and discuss two practical instantiations leading to a discrete and a continuous optimization problems. We then give more details on the best performing derivative-free optimization methods, and provide some insights on our models and optimization strategies. Section B.4 is dedicated to a thorough experimental analysis, where we show we reach state of the art performances by comparing our models with the most powerful black-box approaches on both targeted and untargeted attacks. We also assess our models against the most efficient so far defense strategy based on adversarial training. We finally conclude our paper in Section B.5.

B.2 Related work

Adversarial attacks have a long standing history in the machine learning community. Early works appeared in the mid 2000's where the authors were concerned about Spam classification [Biggio et al., 2009]. Szegedy et al. [2014] revives this research topic by highlighting that deep convolutional networks can be easily fooled. Many adversarial attacks against deep neural networks have been proposed since then. One can distinguish two classes of attacks: white box and black box attacks. In the white box setting, the adversary is supposed to have full knowledge of the network (architecture and parameters), while in the black box one, the adversary only has limited access to the network: she does not know the architecture, and can only query the network and gets labels, logits or probabilities from her queries. An attack is said to have *suceeded* (we also talk about Attack Success Rate), if the input was originally well classified and the generated example is classified to the targeted label.

The white box setting attracted more attention even if it is the more unrealistic between the two. The attacks are crafted by by back-propagating the gradient of the loss function w.r.t. the input. The problem writes as a non-convex optimization procedure that either constraints the perturbation or aims at minimizing its norm. Among the most popular ones, one can cite FGSM [Goodfellow et al., 2015b], PGD [Kurakin et al., 2016, Madry et al., 2018b], Deepfool [Moosavi-Dezfooli et al., 2016], JSMA [Papernot et al., 2016b], Carlini&Wagner attack [Carlini and Wagner, 2017] and EAD [Chen et al., 2018a].

The black box setting is more realistic, but also more challenging. Two strategies emerged in the literature to craft attacks within this setting: transferability from a substitute network, and gradient estimation algorithms. Transferability has been pointed out by Papernot et al. [2017a]. It consists in generating a white-box adversarial example on a fully known substitute neural network, i.e. a network trained on the same classification task. This crafted adversarial example can be *transferred* to the targeted unknown network. Leveraging this property, Moosavi-Dezfooli et al. [2017] proposed an algorithm to craft a single adversarial attack that is the same for all examples and all networks. Despite the popularity of these methods, gradient estimation algorithms outperform transferability methods. Chen et al. [2017] proposed a variant of the powerful white-box attack introduced in [Carlini and Wagner, 2017], based on gradient estimation with finite differences. This method achieves good results in practice but requires a high number of queries to the network. To reduce the number of queries, Ilyas et al. [2018a] proposed to rely rather on Natural Evolution Strategies (NES). These derivative-free optimization approaches consist in estimating the parametric distribution of the minima of a given objective function. This amounts for most of NES algorithms to perform a natural gradient descent in the space of distributions [Ollivier et al., 2017]. In [Al-Dujaili and O'Reilly, 2019], the authors propose to rather estimate the sign of the gradient instead of estimating the its magnitude suing zeroth-order optimization techniques. They show further how to reduce the search space from exponential to linear. The achieved results were state of the art at the publication date. In Liu et al. [2019], the authors introduced a zeroth-order version of the signSGD algorithm, studied its convergence properties and showed its efficiency in crafting adversarial black-box attacks. The results are promising but fail to beat the state of the art. In Tu et al. [2019], the authors introduce the AutoZOOM framework combining gradient estimation and an auto-encoder trained offline with unlabeled data. The idea is appealing but requires training an auto-encoder with an available dataset, which an additional effort for the attacker. Besides, this may be unrealistic for several use cases. More recently, Moon et al. [2019] proposed a method based on discrete and combinatorial optimization where the perturbations are pushed towards the corners of the ℓ_∞ ball. This method is to the best of our knowledge the state of the art in the black box setting in terms of queries budget and success rate. We will focus in our experiments on this method and show how our approaches achieve better results.

Several defense strategies have been proposed to diminish the impact of adversarial attacks on networks accuracies. A basic workaround, introduced in [Goodfellow et al., 2015b], is to augment the learning set with adversarial attacks examples. Such an approach is called adversarial training in the literature. It helps recovering some accuracy but fails to fully defend the network, and lacks theoretical guarantees, in particular principled certificates. Defenses based on randomization at inference time were also proposed [Lecuyer et al., 2018, Cohen et al., 2019, Pinot et al., 2019]. These methods are grounded theoretically, but the guarantees cannot ensure full protection against adversarial examples. The question of defenses and attacks is still widely open since our understanding of this phenomenon is still in its infancy. We evaluate our approach against adversarial training, the most powerful defense method so far.

B.3 Methods

B.3.1 General framework

Let us consider a classification task $\mathcal{X} \mapsto [K]$ where $\mathcal{X} \subseteq \mathbb{R}^d$ is the input space and $[K] = \{1, ..., K\}$ is the corresponding label set. Let $f : \mathbb{R}^d \to \mathbb{R}^K$ be a classifier (a feed forward neural network in our paper) from an input space $\mathcal{X}$ returning the logits of each label in $[K]$ such that the predicted label for a given input is $\arg\max_{i\in[K]} f_i(x)$. The aim of $||.||_\infty$-bounded untargeted adversarial attacks is, for some input x with label y, to find a perturbation τ such that $\arg\max_{i\in[K]} f_i(x) \neq y$. Classically, $||.||_\infty$-bounded untargeted adversarial attacks aims at optimizing the following objective:

$$\max_{\tau:||\tau||_\infty \leq \epsilon} L(f(x+\tau), y) \tag{B.1}$$

where L is a loss function (typically the cross entropy) and y the true label. For targeted attacks, the attacker targets a label y_t by maximizing $-L(f(x+\tau), y_t)$. With access to the gradients of the network, gradient descent methods have proved their efficiency [Kurakin et al., 2016, Madry et al., 2018b]. So far, the outline of most black box attacks was to estimate the gradient using either finite differences or natural evolution strategies. Here using evolutionary strategies heuristics, we do not want to take care of the gradient estimation problem.

B.3.2 Two optimization problems

In some DFO approaches, the default search space is $\mathbb{R}^d$. In the ℓ_∞ bounded adversarial attacks setting, the search space is $B_\infty(\epsilon) = \{\tau : ||\tau||_\infty \leq \epsilon\}$. It requires to adapt the problem in Eq B.1. Two variants are proposed in the sequel leading to continuous and discretized versions of the problem.

The continuous problem. As in Carlini and Wagner [2017], we use the hyperbolic tangent transformation to restate our problem since $B_\infty(\epsilon) = \epsilon \tanh(\mathbb{R}^d)$. This leads to a continuous search space on which evolutionary strategies apply. Hence our optimization problem writes:

$$\max_{\tau\in\mathbb{R}^d} L(f(x+\epsilon\tanh(\tau)), y). \tag{B.2}$$

We will call this problem $\text{DFO}_c -$ optimizer where optimizer is the used black box derivative free optimization strategy.

The discretized problem. Moon et al. [2019] pointed out that PGD attacks [Kurakin et al., 2016, Madry et al., 2018b] are mainly located on the corners of the ℓ_∞-ball. They consider optimizing the following

$$\max_{\tau\in\{-\epsilon,+\epsilon\}^d} L(f(x+\tau), y). \tag{B.3}$$

The author in [Moon et al., 2019] proposed a purely discrete combinatorial optimization to solve this problem (Eq. B.3). As in Zoph and Le [2017], we here consider how to automatically convert an algorithm designed for continuous optimization to discrete optimization. To make the problem in Eq. B.3 compliant with our evolutionary strategies setting, we rewrite our problem by considering a stochastic function $f(x+\epsilon\tau)$ where, for all i, $\tau_i \in \{-1,+1\}$ and $\mathbb{P}(\tau_i = 1) = \mathrm{Softmax}(a_i, b_i) = \frac{e^{a_i}}{e^{a_i}+e^{b_i}}$. Hence our problem amounts to find the best parameters a_i and b_i that optimize:

$$\min_{a,b} \mathbb{E}_{\tau\sim\mathbb{P}_{a,b}}(L(f(x+\epsilon\tau), y) \tag{B.4}$$

We then rely on evolutionary strategies to find the parameters a and b. As the optima are deterministic, the optimal values for a and b are at infinity. Some ES algorithms are well suited to such setting as will be discussed in the sequel. We will call this problem $\mathrm{DFO}_d - \mathrm{optimizer}$ where optimizer is the used black box derivative free optimization strategy for a and b. In this case, one could reduce the problem to one variale a_i with $\mathbb{P}(\tau_i = 1) = \frac{1}{1+e^{-a_i}}$, but experimentally the results are comparable, so we concentrate on Problem B.4.

B.3.3 Derivative-free optimization methods

Derivative-free optimization methods are aimed at optimizing an objective function without access to the gradient. There exists a large and wide literature around derivative free optimisation. In this setting, one algorithm aims to minimize some function f on some space $\mathcal{X}$. The only thing that could be done by this algorithm is to query for some points x the value of $f(x)$. As evaluating f can be computationally expensive, the purpose of DFO methods is to get a good approximation of the optima using a moderate number of queries. We tested several evolution strategies [Rechenberg, 1973, Beyer, 2001]: the simple $(1+1)$-algorithm [Matyas, 1965, Schumer and Steiglitz, 1968], Covariance Matrix Adaptation (CMA [Hansen and Ostermeier, 2003]). For these methods, the underlying algorithm is to iteratively update some distribution P_θ defined on $\mathcal{X}$. Roughly speaking, the current distribution $\mathbb{P}_\theta$ represents the current belief of the localization of the optimas of the goal function. The parameters are updated using objective function values at different points. It turns out that this family of algorithms, than can be reinterpreted as natural evolution strategies, perform best. The two best performing methods will be detailed in Section B.3.3; we refer to references above for other tested methods.

Our best performing methods: evolution strategies

The $(1+1)$-ES algorithm. The $(1+1)$-evolution strategy with one-fifth rule [Matyas, 1965, Schumer and Steiglitz, 1968] is a simple but effective derivative-free optimization algorithm (in supplementary material, Alg. 6). Compared to random search, this algorithm moves the center of the Gaussian sampling according to the best candidate and adapts its scale by taking into account their frequency. Yao and Liu [1996] proposed the use of Cauchy distributions instead of classical Gaussian sampling. This favors large steps, and improves the results in case of (possibly partial) separability of the problem, i.e. when it is meaningful to perform large steps in some directions and very moderate ones in the other directions.

CMA-ES algorithm. The Covariance Matrix Adaptation Evolution Strategy [Hansen and Ostermeier, 2003] combines evolution strategies [Beyer, 2001], Cumulative Step-Size Adaptation [Chotard et al., 2012], and a specific method for adaptating the covariance matrix. An outline is provided in supplementary material, Alg. 7. CMA-ES is an effective and robust algorithm, but it becomes catastrophically slow in high dimension due to the expensive computation of the square root of the matrix. As a workaround, Ros and Hansen [2008] propose to approximate the covariance matrix by a diagonal one. This leads to a computational cost linear in the dimension, rather than the original quadratic one.

Link with Natural Evolution Strategy (NES) attacks. Both (1+1)-ES and CMA-ES can be seen as an instantiation of a natural evolution strategy (see for instance Ollivier et al. [2017], Wierstra et al. [2014]). A natural evolution strategy consists in estimating iteratively the distribution of the optima. For most NES approaches, a fortiori CMA-ES, the iterative estimation consists in a second-order gradient descent (also known as natural gradient) in the space of distributions (e.g. Gaussians). (1+1)-ES can also be seen as a NES, where the covariance matrix is restricted to be proportional to the identity. Note however that from an algorithmic perspective, both CME-ES and (1+1)-ES optimize the quantile of the objective function.

Hypotheses for DFO methods in the adversarial attacks context

The state of the art in DFO and intuition suggest the followings. Using $\mathrm{softmax}$ for exploring only points in the corner (Eq. B.3) is better for moderate budget, as corners are known to be good adversarial candidates; however, for high precision attacks (with small τ) a smooth continuous precision (Eq B.2) is more relevant. With or without $\mathrm{softmax}$, the optimum is at infinity [1], which is in favor of methods having fast step-size adaptation or samplings with heavy-tail distributions. With an optimum at infinity, [Chotard et al., 2012] has shown how fast is the adaptation of the step-size when using cumulative step-size adaptation (as in CMA-ES), as opposed to slower rates for most methods. Cauchy sampling [Yao and Liu, 1996] in the $(1+1)$-ES is known for favoring fast changes; this is consistent with the superiority of Cauchy sampling in our setting compared to Gaussian sampling.

Newuoa, Powell, SQP, Bayesian Optimization, Bayesian optimization are present in Nevergrad but they have an expensive (budget consumption linear is linear w.r.t. the dimension) initial sampling stage which is not possible in our high-dimensional / moderate budget context. The targeted case needs more precision and favors algorithms such as Diagonal CMA-ES which adapt a step-size per coordinate whereas the untargeted case is more in favor of fast random exploration such as the $(1+1)$-ES. Compared to Diagonal-CMA, CMA with full covariance might be too slow; given a number of queries (rather than a time budget) it is however optimal for high precision.

B.3.4 The tiling trick

Ilyas et al. [2018b] suggested to tile the attack to lower the number of queries necessary to fool the network. Concretely, they observe that the gradient coordinates are correlated for close pixels in

[1] i.e. the optima of the ball constrained problem B.1, would be close to the boundary or on the boundary of the ℓ_∞ ball. In that case, the optimum of the continuous problem B.2 will be at ∞ or "close" to it. On the discrete case B.4 it is easy to see that the optimum is when a_i or $b_i \to \infty$.

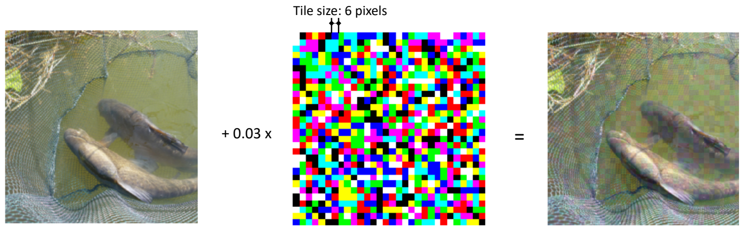


Figure B.1: Illustration of the tiling trick: the same noise is applied on small tile squares.

the images, so they suggested to add the same noise for small square tiles in the image (see Fig. B.1). We exploit the same trick since it reduces the dimensionality of the search space, and makes hence evolutionary strategies suited to the problem at hand. Besides breaking the curse of dimensionality, tiling leads surprisingly to a new property that we discovered during our experiments. At a given tiling scale, convolutional neural networks are not robust to random noise. Section B.4.2 is devoted to this intriguing property. Interestingly enough, initializing our optimization algorithms with a tiled noise at the appropriate scale drastically speeds up the convergence, leading to a reduced number of queries.

B.4 Experiments

B.4.1 General setting and implementation details

We compare our approach to the "bandits" method [Ilyas et al., 2018b] and the parsimonious attack [Moon et al., 2019]. The latter (parsimonious attack) is, to the best of our knowledge, the state of the art in the black-box setting from the literature; bandits method is also considered in our benchmark given its ties to our models. We reproduced the results from [Moon et al., 2019] in our setting for fair comparison. As explained in section B.3.2, our attacks can be interpreted as ℓ_∞ ones. We use the large-scale ImageNet dataset [Deng et al., 2009]. As usually done in most frameworks, we quantify our success in terms of attack success rate, median queries and average queries. Here, the number of queries refers to the number of requests to the output logits of a classifier for a given image. For the success rate, we only consider the images that were correctly classified by our model. We use InceptionV3 [Szegedy et al., 2017] , VGG16 [Simonyan and Zisserman, 2014] with batch normalization (VGG16bn) and ResNet50 [He et al., 2016b] architectures to measure the performance of our algorithm on the ImageNet dataset. These models reach accuracy close to the the state of the art with around $75-80\%$ for the Top-1 accuracy and 95% for the Top-5 accuracy. We use pretrained models from PyTorch [Paszke et al., 2017]. All images are normalized to $[0,1]$. Results on VGG16bn and ResNet50 are deferred in supplementary material B.10. The images to be attacked are selected at random.

We first show that convolutional networks are not robust to tiled random noise, and more surprisingly that there exists an optimal tile size that is the same for all architectures and noise intensities. Then, we evaluate our methods on both targeted and untargeted objectives. We considered the following losses: the cross entropy $L(f(x),y) = -\log(\mathbb{P}(y|x))$ and a loss inspired from

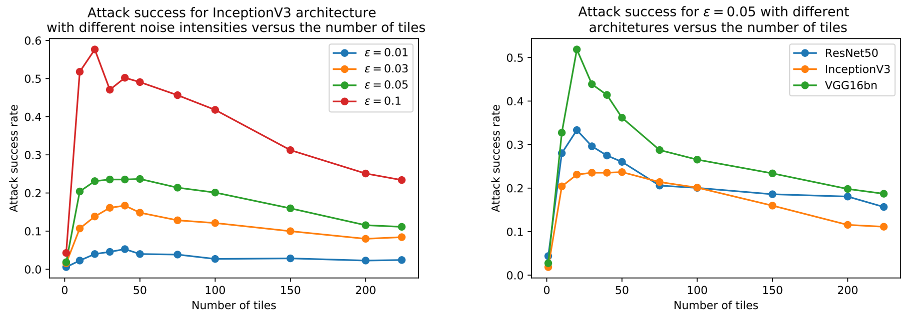


Figure B.2: Success rate of a single shot random attacks on ImageNet vs. the number of tiles used to craft the attack. On the left, attacks are plotted against InceptionV3 classifier with different noise intensities ($\epsilon \in \{0.01, 0.03, 0.05, 0.1\}$). On the right, ϵ is fixed to 0.05 and the single shot attack is evaluated on InceptionV3, ResNet50 and VGG16bn.

the "Carlini&Wagner" attack: $L(f(x), y) = -\mathbb{P}(y|x) + \max_{y' \neq y} \mathbb{P}(y'|x)$ where $\mathbb{P}(y|x) = [\text{Softmax}(f(x))]_y$, the probability for the classifier to classify the input x to label y. The results for the second loss are deferred in supplementary material B.8. For all our attacks, we use the Nevergrad [Rapin and Teytaud, 2018] implementation of evolution strategies. We did not change the default parameters of the optimization strategies.

B.4.2 Convolutional neural networks are not robust to tiled random noise

In this section, we highlight that neural neural networks are not robust to ℓ_∞ tiled random noise. A noise on an image is said to be tiled if the added noise on the image is the same on small squares of pixels (see Figure B.2). In practice, we divide our image in equally sized tiles. For each tile, we add to the image a randomly chosen constant noise: $+\epsilon$ with probability $\frac{1}{2}$ and $-\epsilon$ with probability $\frac{1}{2}$, uniformly on the tile. The tile trick has been introduced inIlyas et al. [2018a] for dimensionality reduction. Here we exhibit a new behavior that we discovered during our experiments. As shown in Fig. B.1 for reasonable noise intensity ($\epsilon = 0.05$), the success rate of a one shot randomly tiled attack is quite high. This fact is observed on many neural network architectures. We compared the number of tiles since the images input size are not the same for all architectures ($299 \times 299 \times 3$ for InceptionV3 and $224 \times 224 \times 3$ for VGG16bn and ResNet50). The optimal number of tiles (in the sense of attack success rate) is, surprisingly, independent from the architecture and the noise intensity. We also note that the InceptionV3 architecture is more robust to random tiled noise than VGG16bn and ResNet50 architectures. InceptionV3 blocks are parallel convolutions with different filter sizes that are concatenated. Using different filter sizes may attenuate the effect of the tiled noise since some convolution sizes might be less sensitive. We test this with a single random attack with various numbers of tiles (cf. Figure B.1, B.2). We plotted additional graphs in supplementary material B.7.

B.4.3 Untargeted adversarial attacks

We first evaluate our attacks in the untargeted setting. The aim is to change the predicted label of the classifier. Following [Moon et al., 2019, Ilyas et al., 2018b], we use $10,000$ images that are

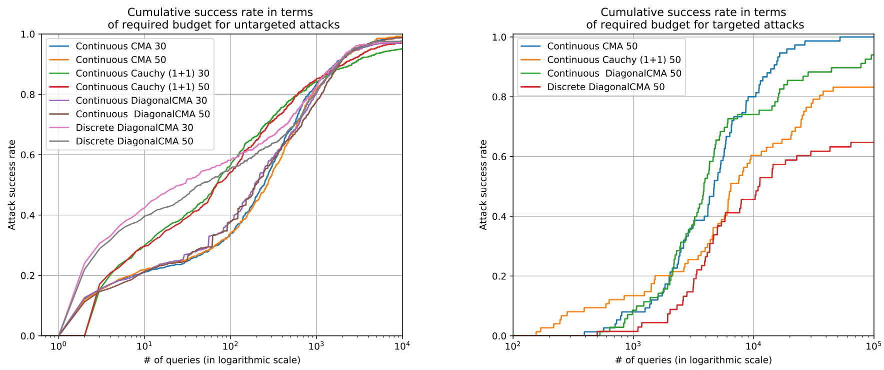


Figure B.3: The cumulative success rate in terms the number of queries for the number of queries required for attacks on ImageNet with $\epsilon = 0.05$ in the untargeted (left) and targeted setting (right). The number of queries (x-axis) is plotted with a logarithmic scale.

initially correctly classified and we limit the budget to $10,000$ queries. We experimented with 30 and 50 tiles on the images. Only the best performing methods are reported in Table B.4. We compare our results with [Moon et al., 2019] and [Ilyas et al., 2018b] on InceptionV3 (cf. Table B.4). We also plotted the cumulative success rate in terms of required budget in Figure B.3. We also evaluated our attacks for smaller noise in supplementary material B.9. We achieve results outperforming or at least equal to the state of the art in all cases. More remarkably, We improve by far the number of necessary queries to fool the classifiers. The tiling trick partially explains why the average and the median number of queries are low. Indeed, the first queries of our evolution strategies is in general close to random search and hence, according to the observation of Figs B.1-B.2, the first steps are more likely to fool the network, which explains why the queries budget remains low. This Discrete strategies reach better median numbers of queries - which is consistent as we directly search on the limits of the ℓ_∞-ball; however, given the restricted search space (only corners of the search space are considered), the success rate is lower and on average the number of queries increases due to hard cases.

B.4.4 Targeted adversarial attacks

We also evaluate our methods in the targeted case on ImageNet dataset. We selected $1,000$ images, correctly classified. Since the targeted task is harder than the untargeted case, we set the maximum budget to $100,000$ queries, and $\epsilon = 0.05$. We uniformly chose the target class among the incorrect ones. We evaluated our attacks in comparison with the bandits methods [Ilyas et al., 2018b] and the parsimonious attack [Moon et al., 2019] on InceptionV3 classifier. We also plotted the cumulative success rate in terms of required budget in Figure B.3. CMA-ES beats the state of the art on all criteria. DiagonalCMA-ES obtains acceptable results but is less powerful that CMA-ES in this specific case. The classical CMA optimizer is more precise, even if the run time is much longer. Cauchy $(1+1)$-ES and discretized optimization reach good results, but when the task is more complicated they do not reach as good results as the state of the art in black box targeted attacks.

Table B.4: Comparison of our method with the parsimonious and bandits attacks in the untargeted setting on ImageNet on InceptionV3 pretrained network for $\epsilon = 0.05$ and $10,000$ as budget limit.

Method	# of tiles	Average queries	Median queries	Success rate
Parsimonious	-	702	222	98.4%
Bandits	30	1007	269	95.3%
Bandits	50	995	249	95.1%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	30	466	60	95.2%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	510	63	97.3%
$\text{DFO}_c - \text{DiagonalCMA}$	30	533	189	97.2%
$\text{DFO}_c - \text{DiagonalCMA}$	50	623	191	98.7%
$\text{DFO}_c - \text{CMA}$	30	589	232	98.9%
$\text{DFO}_c - \text{CMA}$	50	630	259	**99.2%**
$\text{DFO}_d - \text{DiagonalCMA}$	30	**424**	**20**	97.7%
$\text{DFO}_d - \text{DiagonalCMA}$	50	485	38	97.4%

B.4.5 Untargeted attacks against an adversarially trained network

In this section, we experiment our attacks against a defended network by adversarial training [Goodfellow et al., 2015b]. Since adversarial training is computationally expensive, we restricted ourselves to the CIFAR10 dataset [Krizhevsky et al., 2009] for this experiment. Image size is $32 \times 32 \times 3$. We adversarially trained a WideResNet28x10 [Zagoruyko and Komodakis, 2016] with PGD ℓ_∞ attacks [Kurakin et al., 2016, Madry et al., 2018b] of norm $8/256$ and 10 steps of size $2/256$. In this setting, we randomly selected $1,000$ images, and limited the budget to $20,000$ queries. We ran PGD ℓ_∞ attacks [Kurakin et al., 2016, Madry et al., 2018b] of norm $8/256$ and 20 steps of size $1/256$ against our network, and achieved a success rate up to 36%, which is the the state of the art in the white box setting. We also compared our method to the Parsimonious and bandit attacks. Results are reported in Appendix B.11. On this task, the parsimonious attack method is slightly better than our best approach.

B.5 Conclusion

In this paper, we proposed a new framework for crafting black box adversarial attacks based on derivative free optimization. Because of the high dimensionality and the characteristics of the problem (see Section B.3.3), not all optimization strategies give satisfying results. However, combined with the tiling trick, evolutionary strategies such as CMA, DiagonalCMA and Cauchy (1+1)-ES beats the current state of the art in both targeted and untargeted settings. In particular, $\text{DFO}_c - \text{CMA}$ improves the state of the art in terms of success rate in almost all settings. We also validated the robustness of our attack against an adversarially trained network. Future work

Table B.5: Comparison of our method with the parsimonious and bandits attacks in the targeted setting on ImageNet on InceptionV3 pretrained network for $\epsilon = 0.05$ and $100,000$ as budget limit.

Method	**# of tiles**	**Average queries**	**Median queries**	**Success rate**
Parsimonious	-	7184	5116	100%
Bandits	50	25341	18053	92.5%
DFO_c – Cauchy$(1+1)$-ES	50	9789	6049	83.2%
DFO_c – DiagonalCMA	50	6768	**3797**	94.0%
DFO_c – CMA	50	**6662**	4692	**100%**
DFO_d – DiagonalCMA	50	8957	4619	64.2%

will be devoted to better understanding the intriguing property of the effect that a neural network is not robust to a one shot randomly tiled attack.

B.6 Appendix: Algorithms

B.6.1 The (1+1)-ES algorithm

Algorithm 6: The $(1+1)$ Evolution Strategy.

Require: Function $f : \mathbb{R}^d \to \mathbb{R}$ to minimize
  $m \leftarrow 0, C \leftarrow \boldsymbol{I}_d, \sigma \leftarrow 1$
  for $t = 1...n$ **do**
    (Generate candidates)
    Generate $m' \sim m + \sigma X$ where X is sampled from a Cauchy or Gaussian distribution.
    if $f(m') \leq f(m)$ **then**
      $m \leftarrow m', \sigma \leftarrow 2\sigma$
    else
      $\sigma \leftarrow 2^{-\frac{1}{4}}\sigma$
    end if
  end for

B.6.2 CMA-ES algorithm

Algorithm 7: CMA-ES algorithm. The T subscript denotes transposition.

Require: Function $f : \mathbb{R}^d \to \mathbb{R}$ to minimize, parameters $b, c, w_1 > \ldots, w_\mu > 0, p_c$ and others as in e.g. [Hansen and Ostermeier, 2003].
$m \leftarrow 0, C \leftarrow \boldsymbol{I}_d, \sigma \leftarrow 1$
for $t = 1...n$ **do**
Generate $x_1, ..., x_\lambda \sim m + \sigma\mathcal{N}(0, C)$.
Define x'_i the i^{th} best of the x_i.
Update the cumulation for C: $p_c \leftarrow$ cumulation of p_c, overall direction of progress.
Update the covariance matrix:

$$C \leftarrow (1-c) \underbrace{C}_{inertia} + \frac{c}{b} \underbrace{(p_c \times p_c^T)}_{\text{overall direction}} + c(1-\frac{1}{b}) \sum_{i=1}^{\mu} w_i \underbrace{\frac{x'_i - m}{\sigma} \times \frac{(x'_i - m)^T}{\sigma}}_{\text{“covariance” of the } \frac{1}{\sigma}x'_i}$$

Update mean:

$$m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda}$$

Update σ by cumulative step-size adaptation [Chotard et al., 2012].
end for

B.7 Appendix: Additional plots for the tiling trick

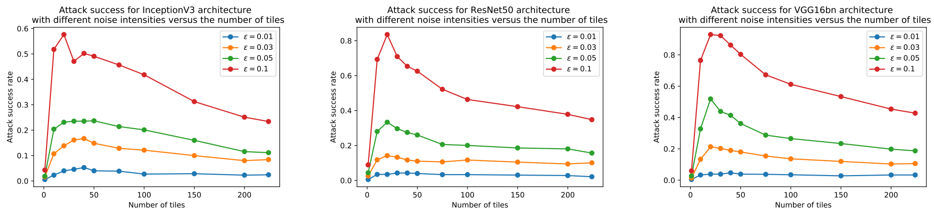


Figure B.6: Random attack success rate against InceptionV3 (left), ResNet50 (center), VGG16bn (right) for different noise intensities. We just randomly draw one tiled attack and check if it is successful.

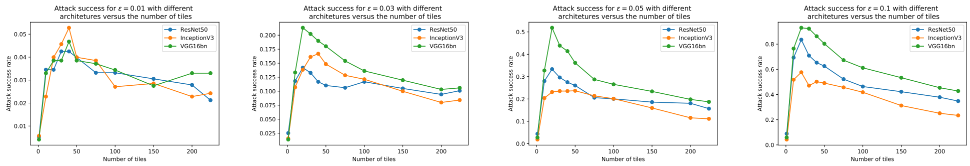


Figure B.7: Random attack success rate for different noise intensities $\epsilon \in \{0.01, 0.03, 0.05, 0.1\}$ (from right to left) against different architectures. We just randomly draw one tiled attack and check if it is successful.

B.8 Appendix: Results with “Carlini&Wagner” loss

In this section, we follow the same experimental setup as in Section B.4.3, but we built our attacks with the “Carlini&Wagner” loss instead of the cross entropy. We remark the results are comparable and similar.

Table B.8: Comparison of our method with“Carlini&Wagner” loss versus the parsimonious and bandits attacks in the untargeted setting on InceptionV3 pretrained network for $\epsilon = 0.05$ and $10,000$ as budget limit.

Method	# of tiles	Average queries	Median queries	Success rate
$\mathrm{DFO}_c - \mathrm{Cauchy}(1+1)$-ES	30	353	57	97.2%
$\mathrm{DFO}_c - \mathrm{Cauchy}(1+1)$-ES	50	**347**	63	98.8%
$\mathrm{DFO}_c - \mathrm{DiagonalCMA}$	30	483	167	98.8%
$\mathrm{DFO}_c - \mathrm{DiagonalCMA}$	50	528	181	99.2%
$\mathrm{DFO}_c - \mathrm{CMA}$	30	475	225	99.2%
$\mathrm{DFO}_c - \mathrm{CMA}$	50	491	246	**99.4%**
$\mathrm{DFO}_d - \mathrm{DiagonalCMA}$	30	482	**27**	98.0%
$\mathrm{DFO}_d - \mathrm{DiagonalCMA}$	50	510	37	98.0%

B.9 Appendix: Untargeted attacks with smaller noise intensities

We evaluated our method on smaller noise intensities ($\epsilon \in \{0.01, 0.03, 0.05\}$) in the untargeted setting on ImageNet dataset. In this framework, we also picked up randomly $10,000$ images and limited our budget to $10,000$ queries. We compared to the bandits method [Ilyas et al., 2018b] and to the parsimonious attack [Moon et al., 2019] on InceptionV3 network. We limited our experiments to a number of tiles of 50. We report our results in Table B.9. We remark our attacks reach state of the art for $\epsilon = 0.03$ and $\epsilon = 0.05$ both in terms of success rate and queries budget. For $\epsilon = 0.01$, we reach results comparable to the state of the art.

Table B.9: Results of our method compared to the parsimonious and bandit attacks in the untargeted setting on InceptionV3 pretrained network for different values of noise intensities $\epsilon \in \{0.01, 0.03, 0.05\}$ and a maximum of $10,000$ queries.

ϵ	**Method**	**# of tiles**	**Avg. queries**	**Med. queries**	**Success rate**
0.05	Parsimonious	-	722	237	98.5%
	Bandits	50	995	249	95.1%
	$\text{DFO}_c - \text{Cauchy}(1+1)$-ES	50	510	63	97.3%
	$\text{DFO}_c - \text{DiagonalCMA}$	50	623	191	98.7%
	$\text{DFO}_c - \text{CMA}$	50	630	259	**99.2%**
	$\text{DFO}_d - \text{DiagonalCMA}$	50	**485**	**38**	97.4%
0.03	Parsimonious	-	1104	392	95.7%
	Bandits	50	1376	466	92.7%
	$\text{DFO}_c - \text{Cauchy}(1+1)$-ES	50	846	**203**	93,2%
	$\text{DFO}_c - \text{DiagonalCMA}$	50	971	429	96,5%
	$\text{DFO}_c - \text{CMA}$	50	911	404	**96.7%**
	$\text{DFO}_d - \text{DiagonalCMA}$	50	**799**	293	94,1%
0.01	Parsimonious	-	2104	1174	80.3%
	Bandits	50	2018	992	72.9%
	$\text{DFO}_c - \text{Cauchy}(1+1)$-ES	50	1668	**751**	72,1%
	$\text{DFO}_c - \text{DiagonalCMA}$	50	1958	1175	79.2%
	$\text{DFO}_c - \text{CMA}$	50	1921	1107	**80.4%**
	$\text{DFO}_d - \text{DiagonalCMA}$	50	**1188**	849	71,3%

B.10 Appendix: Untargeted attacks against other architectures

We also evaluated our method on different neural networks architectures. For each network we randomly selected $10,000$ images that were correctly classified. We limit our budget to $10,000$ queries and set the number of tiles to 50. We achieve a success attack rate up to 100% on every classifier with a budget as low as 8 median queries for the VGG16bn for instance (see Table B.10). One should notice that the performances are lower on InceptionV3 as it is also reported for the bandit methods in [Ilyas et al., 2018b]. This possibly due to the fact that the tiling trick is less relevant on the Inception network than on the other networks (see Fig. B.2).

Table B.10: Comparison of our method on the ImageNet dataset with InceptionV3 (I), ResNet50 (R) and VGG16bn (V) for $\epsilon = 0.05$ and $10,000$ as budget limit.

Method	Tile size	Avg queries			Med. queries			Succ. Rate		
		I	R	V	I	R	V	I	R	V
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	30	466	**163**	86	60	**19**	8	95.2%	99.6%	**100%**
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	50	510	218	**67**	63	32	4	97.3%	99.6%	99.7%
$\text{DFO}_c - \text{DiagonalCMA}$	30	533	263	174	189	95	55	97.2%	99.0%	99.9%
$\text{DFO}_c - \text{DiagonalCMA}$	50	623	373	227	191	121	71	98.7%	99.9%	**100%**
$\text{DFO}_c - \text{CMA}$	30	588	256	176	232	138	72	98.9%	99.9%	99.9%
$\text{DFO}_c - \text{CMA}$	50	630	270	219	259	143	107	**99.2%**	**100%**	99.9%
$\text{DFO}_d - \text{DiagonalCMA}$	50	485	617	345	38	62	6	97.4%	99.2%	99.6%
$\text{DFO}_d - \text{DiagonalCMA}$	30	**424**	417	211	**20**	20	**2**	97.7%	98.8%	99.5%

B.11 Appendix: Table for attacks against adversarially tranined network

Table B.11: Adversarial attacks against an adversarially trained WideResnet28x10 network on CIFAR10 dataset for $\epsilon = 0.03125$ and $20,000$ as budget limit.

Method	# of tiles	Avg. queries	Med. queries	Success rate
PGD (not black-box)	-	20	20	36%
Parsimonious	-	1130	450	**42%**
Bandits	10	1429	530	29.1%
Bandits	20	1802	798	33.8%
Bandits	32	1993	812	34.8%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	10	429	**60**	29.5%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	20	902	93	30.5%
$\text{DFO}_c - \text{Cauchy}(1+1)\text{-ES}$	32	1865	764	31.7%
$\text{DFO}_c - \text{DiagonalCMA}$	10	395	85	30.5%
$\text{DFO}_c - \text{DiagonalCMA}$	20	624	151	31.3%
$\text{DFO}_c - \text{DiagonalCMA}$	32	1379	860	34.7%
$\text{DFO}_c - \text{CMA}$	10	**363**	156	30.4%
$\text{DFO}_c - \text{CMA}$	20	1676	740	40.2%
$\text{DFO}_c - \text{CMA}$	32	2311	1191	40.2%

B.12 Appendix: Failing methods

In this section, we compare our attacks to other optimization strategies. We run our experiments in the same setup as in Section B.4.3. Results are reported in Table B.12. DE and Normal (1+1)-ES performs poorly, probably because these optimization strategies converge slower when the optima are at "infinity". We reformulate this sentence accordingly in the updated version of the paper. Finally, as the initialization of Powell is linear with the dimension and with less variance, it performs poorer than simple random search. Newuoa, SQP and Cobyla algorithms have also been tried on a smaller number images (we did not report the results), but their initialization is also linear in the dimension, so they reach very poor results too.

Table B.12: Comparison with other DFO optimization strategies in the untargeted setting on ImageNet dataset InceptionV3 pretrained network for $\epsilon = 0.05$ and $10,000$ as budget limit.

Method	**# of tiles**	**Avg. queries**	**Med. queries**	**Success rate**
$\mathrm{DFO}_c - \mathrm{Cauchy}(1+1)\text{-ES}$	30	466	60	95.2%
$\mathrm{DFO}_c - \mathrm{Cauchy}(1+1)\text{-ES}$	50	510	63	97.3%
$\mathrm{DFO}_c - \mathrm{DiagonalCMA}$	30	533	189	97.2%
$\mathrm{DFO}_c - \mathrm{DiagonalCMA}$	50	623	191	98.7%
$\mathrm{DFO}_c - \mathrm{CMA}$	30	589	232	98.9%
$\mathrm{DFO}_c - \mathrm{CMA}$	50	630	259	99.2%
$\mathrm{DFO}_c - \mathrm{DE}$	30	756	159	78.8%
$\mathrm{DFO}_c - \mathrm{DE}$	50	699	149	76.0%
$\mathrm{DFO}_c - \mathrm{Normal}(1+1)\text{-ES}$	30	581	45	87.6%
$\mathrm{DFO}_c - \mathrm{Normal}(1+1)\text{-ES}$	50	661	66	92.8%
$\mathrm{DFO}_c - \mathrm{RandomSearch}$	30	568	6	37.9%
$\mathrm{DFO}_c - \mathrm{RandomSearch}$	50	527	5	38.2%
$\mathrm{DFO}_c - \mathrm{Powell}$	30	4889	5332	14.4%
$\mathrm{DFO}_c - \mathrm{Powell}$	50	4578	4076	7.3%

C Advocating for Multiple Defense Strategies against Adversarial Examples

It has been empirically observed that defense mechanisms designed to protect neural networks against ℓ_∞ adversarial examples offer poor performance against ℓ_2 adversarial examples and vice versa. In this paper we conduct a geometrical analysis that validates this observation. Then, we provide a number of empirical insights to illustrate the effect of this phenomenon in practice. Then, we review some of the existing defense mechanism that attempts to defend against multiple attacks by mixing defense strategies. Thanks to our numerical experiments, we discuss the relevance of this method and state open questions for the adversarial examples community.

C.1 Introduction

Deep neural networks achieve state-of-the-art performances in a variety of domains such as natural language processing Radford et al. [2018], image recognition He et al. [2016b] and speech recognition Hinton et al. [2012]. However, it has been shown that such neural networks are vulnerable to *adversarial examples*, *i.e.*, imperceptible variations of the natural examples, crafted to deliberately mislead the models Globerson et al. [2006], Biggio et al. [2013], Szegedy et al. [2014]. Since their discovery, a variety of algorithms have been developed to generate adversarial examples (a.k.a. attacks), for example FGSM [Goodfellow et al., 2015b], PGD [Madry et al., 2018b] and C&W [Carlini and Wagner, 2017], to mention the most popular ones.

Because it is difficult to characterize the space of visually imperceptible variations of a natural image, existing adversarial attacks use surrogates that can differ from one attack to another. For example, Goodfellow et al. [2015b] use the ℓ_∞ norm to measure the distance between the original image and the adversarial image whereas Carlini and Wagner [2017] use the ℓ_2 norm. When the input dimension is low, the choice of the norm is of little importance because the ℓ_∞ and ℓ_2 balls overlap by a large margin, and the adversarial examples lie in the same space. An important insight in this paper is to observe that the overlap between the two balls diminishes exponentially quickly as the dimensionality of the input space increases. For typical image datasets with large dimensionality, the two balls are mostly disjoint. As a consequence, the ℓ_∞ and the ℓ_2 adversarial examples lie in different areas of the space, and it explains why ℓ_∞ defense mechanisms perform poorly against ℓ_2 attacks and vice versa.

Building on this insight, we advocate for designing models that incorporate defense mechanisms against both ℓ_∞ and ℓ_2 attacks and review several ways of mixing existing defense mechanisms. In particular, we evaluate the performance of *Mixed Adversarial Training* (MAT) Goodfellow et al. [2015b] which consists of augmenting training batches using *both* ℓ_∞ and ℓ_2 adversarial examples, and *Randomized Adversarial Training* (RAT) Salman et al. [2019], a solution to benefit from the advantages of both ℓ_∞ adversarial training, and ℓ_2 randomized defense.

Outline. The rest is organized as follows. In Section C.2, we recall the principle of existing attacks and defense mechanisms. In Section C.3, we conduct a theoretical analysis to show why the ℓ_∞ defense mechanisms cannot be robust against ℓ_2 attacks and vice versa. We then corroborate this analysis with empirical results using real adversarial attacks and defense mechanisms. In Section C.4, we discuss various strategies to mix defense mechanisms, conduct comparative experiments, and discuss the performance of each strategy.

C.2 Preliminaries on Adversarial Attacks and Defenses

Let us first consider a standard classification task with an input space $\mathcal{X} = [0,1]^d$ of dimension d, an output space $\mathcal{Y} = [K]$ and a data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. We assume the model f_θ has been trained to minimize the expectation over $\mathcal{D}$ of a loss function $\mathcal{L}$ as follows:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(f_\theta(x), y)]. \tag{C.1}$$

C.2.1 Adversarial attacks

Given an input-output pair $(x, y) \sim \mathcal{D}$, an *adversarial attack* is a procedure that produces a small perturbation $\tau \in \mathcal{X}$ such that $f_\theta(x+\tau) \neq y$. To find the best perturbation τ, existing attacks can adopt one of the two following strategies: (i) maximizing the loss $\mathcal{L}(f_\theta(x+\tau), y)$ under some constraint on $\|\tau\|_p$[1] (a.k.a. loss maximization); or (ii) minimizing $\|\tau\|_p$ under some constraint on the loss $\mathcal{L}(f_\theta(x+\tau), y)$ (a.k.a. perturbation minimization).

(i) Loss maximization. In this scenario, the procedure maximizes the loss objective function, under the constraint that the ℓ_p norm of the perturbation remains bounded by some value ϵ, as follows:

$$\operatorname*{argmax}_{\|\tau\|_p \leq \epsilon} \mathcal{L}(f_\theta(x+\tau), y). \tag{C.2}$$

The typical value of ϵ depends on the norm $\|\cdot\|_p$ considered in the problem setting. In order to compare ℓ_∞ and ℓ_2 attacks of similar strength, we choose values of ϵ_∞ and ϵ_2 (for ℓ_∞ and ℓ_2 norms respectively) which result in ℓ_∞ and ℓ_2 balls of equivalent volumes. For the particular case of CIFAR-10, this would lead us to choose $\epsilon_\infty = 0.03$ and $\epsilon_2 = 0.8$ which correspond to the maximum values chosen empirically to avoid the generation of visually detectable perturba-

[1] with $p \in \{0, \cdots, \infty\}$.

tions. The current state-of-the-art method to solve Problem (C.2) is based on a projected gradient descent (PGD) Madry et al. [2018b] of radius ϵ. Given a budget ϵ, it recursively computes

$$x^{t+1} = \prod_{B_p(x,\epsilon)} \left(x^t + \alpha \underset{\delta \text{ s.t. } ||\delta||_p \leq 1}{\operatorname{argmax}} \left(\Delta^t | \delta \right) \right) \tag{C.3}$$

where $B_p(x, \epsilon) = \{x + \tau \text{ s.t. } \|\tau\|_p \leq \epsilon\}$, $\Delta^t = \nabla_x \mathcal{L}\big(f_\theta\big(x^t\big), y\big)$, α is a gradient step size, and $\prod_S$ is the projection operator on S. Both PGD attacks with $p = 2$, and $p = \infty$ are currently used in the literature as state-of-the-art attacks for the loss maximization problem.

(ii) Perturbation minimization. This type of procedure search for the perturbation that has the minimal ℓ_p norm, under the constraint that $\mathcal{L}(f_\theta(x + \tau), y)$ is bigger than a given bound c:

$$\underset{\mathcal{L}(f_\theta(x+\tau),y) \geq c}{\operatorname{argmin}} \quad \|\tau\|_p. \tag{C.4}$$

The value of c is typically chosen depending on the loss function $\mathcal{L}$[2]. Problem (C.4) has been tackled in Carlini and Wagner [2017], leading to the following method, denoted C&W attack in the rest of this appendix. It aims at solving the following Lagrangian relaxation of Problem (C.4):

$$\underset{\tau}{\operatorname{argmin}} \|\tau\|_p + \lambda \times g(x + \tau) \tag{C.5}$$

where $g(x + \tau) < 0$ if and only if $\mathcal{L}(f_\theta(x + \tau), y) \geq c$. The authors use a change of variable $\tau = \tanh(w) - x$ to ensure that $-1 \leq x + \tau \leq 1$, a binary search to optimize the constant c, and Adam or SGD to compute an approximated solution. The C&W attack is well defined both for $p = 2$, and $p = \infty$, but there is a clear empirical gap of efficiency in favor of the ℓ_2 attack.

In this appendix, we focus on the *Loss Maximization* setting using the PGD attack. However we conduct some of our experiments using *Perturbation Minimization* algorithms such as C&W to capture more detailed information about the location of adversarial examples in the vector space[3].

C.2.2 Defense mechanisms

Adversarial Training (AT). Adversarial Training was introduced in Goodfellow et al. [2015b] and later improved in Madry et al. [2018b] as a first defense mechanism to train robust neural networks. It consists in augmenting training batches with adversarial examples generated during the training procedure. The standard training procedure from Equation (C.1) is thus replaced by the following min max problem, where the classifier tries to minimize the expected loss under maximum perturbation of its input:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\tau\|_p \leq \epsilon} \mathcal{L}(f_\theta(x + \tau), y) \right]. \tag{C.6}$$

[2] For example, if $\mathcal{L}$ is the $0/1$ loss, any $c > 0$ is acceptable.

[3] As it has a more flexible geometry than the *Loss Maximization* attacks.

In the case where $p = \infty$, this technique offers good robustness against ℓ_∞ attacks Athalye et al. [2018a]. AT can also be used with ℓ_2 attacks but as we will discuss in Section C.3, AT with one norm offers poor protection against the other. The main weakness of Adversarial Training is its lack of formal guarantees. Despite some recent work providing great insights Sinha et al. [2017], Zhang et al. [2019a], there is no worst case lower bound yet on the accuracy under attack of this method.

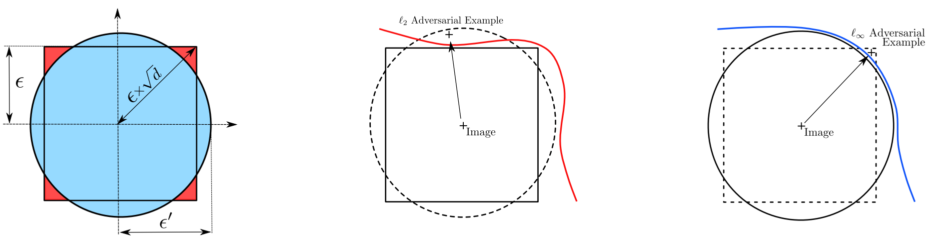
Noise injection mechanisms (NI). Another important technique to defend against adversarial examples is to use Noise Injection. In contrast with Adversarial Training, Noise Injection mechanisms are usually deployed after training. In a nutshell, it works as follows. At inference time, given a unlabeled sample x, the network outputs

$$\tilde{f}_\theta(x) := f_\theta(x + \eta) \quad \text{(instead of } f_\theta(x)) \tag{C.7}$$

where η is a random variable on $\mathbb{R}^d$. Even though, Noise Injection is often less efficient than Adversarial Training in practice (see *e.g.*, Table C.5), it benefits from strong theoretical background. In particular, recent work Lecuyer et al. [2018], followed by Cohen et al. [2019], Pinot et al. [2019] demonstrated that noise injection from a Gaussian distribution can give provable defense against ℓ_2 adversarial attacks. In this work, besides the classical Gaussian noises already investigated in previous works, we evaluate the efficiency of Uniform distributions to defend against ℓ_2 adversarial examples.

C.3 No Free Lunch for Adversarial Defenses

In this Section, we show both theoretically and empirically that defenses mechanisms intending to defend against ℓ_∞ attacks cannot provide suitable defense against ℓ_2 attacks. Our reasoning is perfectly general; hence we can similarly demonstrate the reciprocal statement, but we focus on this side for simplicity.



(a) (b) (c)

Figure C.1: Left: 2D representation of the ℓ_∞ and ℓ_2 balls of respective radius ϵ and ϵ'. Middle: a classifier trained with ℓ_∞ adversarial perturbations (materialized by the red line) remains vulnerable to ℓ_2 attacks. Right: a classifier trained with ℓ_2 adversarial perturbations (materialized by the blue line) remains vulnerable to ℓ_∞ attacks.

C.3.1 Theoretical analysis

Let us consider a classifier f_∞ that is provably robust against adversarial examples with maximum ℓ_∞ norm of value ϵ_∞. It guarantees that for any input-output pair $(x, y) \sim \mathcal{D}$ and for any perturbation τ such that $\|\tau\|_\infty \leq \epsilon_\infty$, f_∞ is not misled by the perturbation, *i.e.*, $f_\infty(x + \tau) = f_\infty(x)$. We now focus our study on the performance of this classifier against adversarial examples bounded with a ℓ_2 norm of value ϵ_2. Using Figure C.1(a), we observe that any ℓ_2 adversarial example that is also in the ℓ_∞ ball, will not fool f_∞. Conversely, if it is outside the ball, we have no guarantee.

To characterize the probability that such an ℓ_2 perturbation fools an ℓ_∞ defense mechanism in the general case (*i.e.*, any dimension d), we measure the ratio between the volume of the intersection of the ℓ_∞ ball of radius ϵ_∞ and the ℓ_2 ball of radius ϵ_2. As Theorem 22 shows, this ratio depends on the dimensionality d of the input vector x, and rapidly converges to zero when d increases. Therefore a defense mechanism that protects against all ℓ_∞ bounded adversarial examples is unlikely to be efficient against ℓ_2 attacks.

Theorem 22 (Probability of the intersection goes to 0)**.** *Let* $B_{2,d}(\epsilon) := \left\{\tau \in \mathbb{R}^d \textit{ s.t } \|\tau\|_2 \leq \epsilon\right\}$ *and* $B_{\infty,d}(\epsilon') := \left\{\tau \in \mathbb{R}^d \textit{ s.t } \|\tau\|_\infty \leq \epsilon'\right\}$*. If for all d, we select* ϵ *and* ϵ' *such that* $\mathrm{Vol}(B_{2,d}(\epsilon)) = \mathrm{Vol}(B_{\infty,d}(\epsilon'))$*, then*

$$\frac{\mathrm{Vol}(B_{2,d}(\epsilon) \bigcap B_{\infty,d}(\epsilon'))}{\mathrm{Vol}(B_{\infty,d}(\epsilon'))} \to 0 \textit{ when } d \to \infty.$$

Proof. Without loss of generality, let us fix $\epsilon = 1$. One can show that for all d,

$$\mathrm{Vol}\left(B_{2,d}\left(\frac{2}{\sqrt{\pi}}\Gamma\left(\frac{d}{2}+1\right)^{1/d}\right)\right) = \mathrm{Vol}(B_{\infty,d}(1)) \tag{C.8}$$

where Γ is the gamma function. Let us denote

$$r_2(d) = \frac{2}{\sqrt{\pi}}\Gamma\left(\frac{d}{2}+1\right)^{1/d}. \tag{C.9}$$

Then, thanks to Stirling's formula

$$r_2(d) \sim \sqrt{\frac{2}{\pi e}} d^{1/2}. \tag{C.10}$$

Finally, if we denote $\mathcal{U}_S$, the uniform distribution on set S, by using Hoeffding inequality between Equation C.14 and C.15, we get:

$$\frac{\mathrm{Vol}(B_{2,d}(r_2(d)) \bigcap B_{\infty,d}(1))}{\mathrm{Vol}(B_{\infty,d}(1))} \tag{C.11}$$

$$= \mathbb{P}_{x \sim \mathcal{U}_{B_{\infty,d}(1)}}[x \in B_{2,d}(r_2(d))] \tag{C.12}$$

$$=\mathbb{P}_{x\sim\mathcal{U}_{B_{\infty,d}(1)}}\left[\sum_{i=1}^{d}|x_i|^2 \le r_2^2(d)\right] \tag{C.13}$$

$$\le \exp\left\{-d^{-1}\left(r_2^2(d) - d\mathbb{E}|x_1|^2\right)^2\right\} \tag{C.14}$$

$$\le \exp\left\{-\left(\frac{2}{\pi e} - \frac{1}{3}\right)^2 d + o(d)\right\}. \tag{C.15}$$

Then the ratio between the volume of the intersection of the ball and the volume of the ball converges towards 0 when d goes to ∞. □

Theorem 22 states that, when d is large enough, ℓ_2 bounded perturbations have a null probability of being also in the ℓ_∞ ball of the same volume. As a consequence, for any value of d that is large enough, a defense mechanism that offers full protection against ℓ_∞ adversarial examples is not guaranteed to offer any protection against ℓ_2 attacks[4].

Table C.2: Bounds of Theorem 22 on the volume of the intersection of ℓ_2 and ℓ_∞ balls at equal volume for typical image classification datasets. When $d = 2$, the bound is $10^{-0.009} \approx 0.98$.

Dataset	**Dim.** (**d**)	**Vol. of the intersection**
–	2	$10^{-0.009}$ ($\approx$ 0.98)
MNIST	784	10^{-144}
CIFAR	3072	10^{-578}
ImageNet	150528	10^{-28946}

Note that this result defeats the 2-dimensional intuition: if we consider a 2 dimensional problem setting, the ℓ_∞ and the ℓ_2 balls have an important overlap (as illustrated in Figure C.1(a)) and the probability of sampling at the intersection of the two balls is bounded by approximately 98%. However, as we increase the dimensionality d, this probability quickly becomes negligible, even for very simple image datasets such as MNIST. An instantiation of the bound for classical image datasets is presented in Table C.2. The probability of sampling at the intersection of the ℓ_∞ and ℓ_2 balls is close to zero for any realistic image setting. In large dimensions, the volume of the corner of the ℓ_∞ ball is much bigger than it appears in Figure C.1(a).

C.3.2 No Free Lunch in Practice

Our theoretical analysis shows that if adversarial examples were uniformly distributed in a high-dimensional space, then any mechanism that perfectly defends against ℓ_∞ adversarial examples has a null probability of protecting against ℓ_2-bounded adversarial attacks. Although existing defense mechanisms do not necessarily assume such a distribution of adversarial examples, we demonstrate that whatever distribution they use, it offers no favorable bias with respect to the result of Theorem 22. As we discussed in Section C.2, there are two distinct attack settings: loss

[4] Th. 22 can easily be extended to any two balls with different norms. For clarity, we restrict to the case of ℓ_∞ and ℓ_2 norms.

maximization (PGD) and perturbation minimization (C&W). Our analysis is mainly focusing on loss maximization attacks. However, these attacks have a very strict geometry[5]. This is why, to present a deeper analysis of the behavior of adversarial attacks and defenses, we also present a set of experiments that use perturbation minimization attacks.

Table C.3: Average norms of PGD-ℓ_2 and PGD-ℓ_∞ adversarial examples with and without ℓ_∞ adversarial training on CIFAR-10 ($d = 3072$).

	Attack PGD-ℓ_2		Attack PGD-ℓ_∞	
	Unprotected	AT-ℓ_∞	Unprotected	AT-ℓ_2
Average ℓ_2 norm	0.830	0.830	1.400	1.640
Average ℓ_∞ norm	0.075	0.200	0.031	0.031

Adversarial training vs. loss maximization attacks To demonstrate that ℓ_∞ adversarial training is not robust against PGD-ℓ_2 attacks we measure the evolution of ℓ_2 norm of adversarial examples generated with PGD-ℓ_∞ between an unprotected model and a model trained with AT-ℓ_∞, *i.e.*, AT where adversarial examples are generated with PGD-ℓ_∞ [6]. Results are presented in Table C.3. [7]

The analysis is unambiguous: the average ℓ_∞ norm of a bounded ℓ_2 perturbation more than double between an unprotected model and a model trained with AT PGD-ℓ_∞. This phenomenon perfectly reflects the illustration of Figure C.1 (c). The attack will generate an adversarial example on the corner of the ℓ_∞ ball thus increasing the ℓ_∞ norm while maintaining the same ℓ_2 norm. We can observe the same phenomenon with AT-ℓ_2 against PGD-ℓ_∞ attack (see Figure C.1 (b) and Table C.3). PGD-ℓ_∞ attack increases the ℓ_2 norm while maintaining the same ℓ_∞ perturbation thus generating the perturbation in the upper area.

As a consequence, we cannot expect adversarial training ℓ_∞ to offer any guaranteed protection against ℓ_2 adversarial examples .

Adversarial training vs. perturbation minimization attacks. To better capture the behavior of ℓ_2 adversarial examples, we now study the performances of an ℓ_2 perturbation minimization attack (C&W) with and without AT-ℓ_∞. It allows us to understand in which area C&W discovers adversarial examples and the impact of AT-ℓ_∞. In high dimensions, the red corners (see Figure C.1 (a)) are very far away from the ℓ_2 ball. Therefore, we hypothesize that a large proportion of the ℓ_2 adversarial examples will remain unprotected. To validate this assumption, we measure the

[5]Due to the projection operator, all PGD attacks saturate the constraint, which makes them all lies in a very small part of the ball.

[6]To do so, we use the same experimental setting as in Section C.4 with ϵ_∞ and ϵ_2 such that the volumes of the two balls are equal.

[7]All experiments in this section are conducted on CIFAR-10, and the experimental setting is fully detailed in Section C.4.1.

proportion of adversarial examples inside of the ℓ_2 ball before and after ℓ_∞ adversarial training. The results are presented in Figure C.4 (left: without adversarial training, right: with adversarial training).

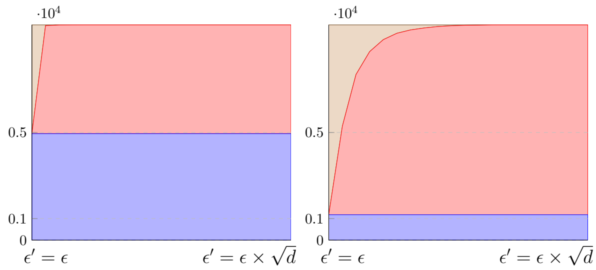


Figure C.4: Comparison of the number of adversarial examples found by C&W, inside the ℓ_∞ ball (lower, blue area), outside the ℓ_∞ ball but inside the ℓ_2 ball (middle, red area) and outside the ℓ_2 ball (upper gray area). ϵ is set to 0.3 and ϵ' varies along the x-axis. Left: without adversarial training, right: with adversarial training. Most adversarial examples have shifted from the ℓ_∞ ball to the cap of the ℓ_2 ball, but remain at the same ℓ_2 distance from the original example.

On both charts, the blue area represents the proportion of adversarial examples that are inside the ℓ_∞ ball. The red area represents the adversarial examples that are outside the ℓ_∞ ball but still inside the ℓ_2 ball (valid ℓ_2 adversarial examples). Finally, the brown-beige area represents the adversarial examples that are beyond the ℓ_2 bound. The radius ϵ' of the ℓ_2 ball varies along the x-axis from ϵ' to $\epsilon'\sqrt{d}$. On the left chart (without adversarial training) most ℓ_2 adversarial examples generated by C&W are inside both balls. On the right chart most of the adversarial examples have been shifted out the ℓ_∞ ball. This is the expected consequence of ℓ_∞ adversarial training. However, these adversarial examples remain in the ℓ_2 ball, *i.e.*, they are in the cap of the ℓ_2 ball. These examples are equally good from the ℓ_2 perspective. This means that even after adversarial training, it is still easy to find good ℓ_2 adversarial examples, making the ℓ_2 robustness of AT-ℓ_∞ almost null.

C.4 Reviewing Defenses Against Multiple Attacks

Adversarial attacks have been an active topic in the machine learning community since their discovery Globerson et al. [2006], Biggio et al. [2013], Szegedy et al. [2014]. Many attacks have been developed. Most of them solve a loss maximization problem with either ℓ_∞ Goodfellow et al. [2015b], Kurakin et al. [2016], Madry et al. [2018b], ℓ_2 Carlini and Wagner [2017], Kurakin et al. [2016], Madry et al. [2018b], ℓ_1 Tramèr and Boneh [2019] or ℓ_0 Papernot et al. [2016] surrogate norms. As we showed, these norms are really different in high dimension. Hence, defending against one norm-based attack is not sufficient to protect against another one. In order to solve this problem, we review several strategies to build defenses against multiple adversarial attacks. These strategies are based on the idea that both types of defense must be used simultaneously in order for the classifier to be protected against multiple attacks. The detailed description of the experimental setting is described in Section C.4.1.

Table C.5: This table shows a comprehensive list of results consisting of the accuracy of several defense mechanisms against ℓ_2 and ℓ_∞ attacks. This table main objective is to compare the overall performance of 'single' norm defense mechanisms (AT and NI presented in the Section C.2.2) against mixed norms defense mechanisms (MAT & RAT mixed defenses presented in Section C.4).

	Baseline	**AT**		**MAT**		**NI**		**RAT-ℓ_∞**		**RAT-ℓ_2**	
	–	ℓ_∞	ℓ_2	Max	Rand	$\mathcal{N}$	$\mathcal{U}$	$\mathcal{N}$	$\mathcal{U}$	$\mathcal{N}$	$\mathcal{U}$
Natural	0.94	0.85	0.85	0.80	0.80	0.79	0.87	0.74	0.80	0.79	0.8
PGD-ℓ_∞	0.00	0.43	0.37	0.37	0.40	0.23	0.22	0.35	0.40	0.23	0.2
PGD-ℓ_2	0.00	0.37	0.52	0.50	0.55	0.34	0.36	0.43	0.39	0.34	0.3

C.4.1 Experimental Setting

To compare the robustness provided by the different defense mechanisms, we use strong adversarial attacks and a conservative setting: the attacker has a total knowledge of the parameters of the model (white-box setting) and we only consider untargeted attacks (a misclassification from one target to any other will be considered as adversarial). To evaluate defenses based on Noise Injection, we use *Expectation Over Transformation* (EOT), the rigorous experimental protocol proposed by Athalye et al. [2018c] and later used by Athalye et al. [2018a], Carlini et al. [2019] to identify flawed defense mechanisms.

To attack the models, we use state-of-the-art algorithms PGD. We run PGD with 20 iterations to generate adversarial examples and with 10 iterations when it is used for adversarial training. The maximum ℓ_∞ bound is fixed to 0.031 and the maximum ℓ_2 bound is fixed to 0.83. As discussed in Section C.2, we chose these values so that the ℓ_∞ and the ℓ_2 balls have similar volumes. Note that 0.83 is slightly above the values typically used in previous publications in the area, meaning the attacks are stronger, and thus more difficult to defend against.

All experiments are conducted on CIFAR-10 with the Wide-Resnet 28-10 architecture. We use the training procedure and the hyper-parameters described in the original paper by Zagoruyko and Komodakis [2016]. Training time varies from 1 day (AT) to 2 days (MAT) on 4 GPUs-V100 servers.

C.4.2 MAT – Mixed Adversarial Training

Earlier results have shown that AT-ℓ_p improves the robustness against corresponding ℓ_p-bounded adversarial examples, and the experiments we present in this section corroborate this observation (See Table C.5, column: AT). Building on this, it is natural to examine the efficiency of *Mixed Adversarial Training* (MAT) against mixed ℓ_∞ and ℓ_2 attacks. MAT is a variation of AT that uses both ℓ_∞-bounded adversarial examples and ℓ_2-bounded adversarial examples as training examples. As discussed in Tramèr and Boneh [2019], there are several possible strategies to mix the adversarial training examples. The first strategy (MAT-Rand) consists in randomly selecting one adversarial example among the two most damaging ℓ_∞ and ℓ_2, and to use it as a training example, as described in Equation (C.16):

MAT-Rand :

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{E}_{p\sim\mathcal{U}(\{2,\infty\})} \max_{\|\tau\|_p\leq\epsilon} \mathcal{L}(f_\theta(x+\tau),y)\right]. \tag{C.16}$$

An alternative strategy is to systematically train the model with the most damaging adversarial example (ℓ_∞ or ℓ_2). As described in Equation (C.17):

MAT-Max :

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{p\in\{2,\infty\}} \max_{\|\tau\|_p\leq\epsilon} \mathcal{L}(f_\theta(x+\tau),y)\right]. \tag{C.17}$$

The accuracy of MAT-Rand and MAT-Max are reported in Table C.5 (Column: MAT). As expected, we observe that MAT-Rand and MAT-Max offer better robustness both against PGD-ℓ_2 and PGD-ℓ_∞ adversarial examples than the original AT does. More generally, we can see that AT is a good strategy against loss maximization attacks, and thus it is not surprising that MAT is a good strategy against mixed loss maximization attacks. However efficient in practice, MAT (for the same reasons as AT) lacks theoretical arguments. In order to get the best of both worlds, Salman et al. [2019] proposed to mix adversarial training with randomization.

C.4.3 RAT – Randomized Adversarial Training

We now examine the performance of Randomized Adversarial Training (RAT) first introduced in Salman et al. [2019]. This technique mixes Adversarial Training with Noise Injection. The corresponding loss function is defined as follows:

$$\min_\theta \mathbb{E}(x,y)\sim\mathcal{D}\left[\max_{\|\tau\|_p\leq\epsilon} \mathcal{L}\Big(\tilde{f}_\theta(x+\tau),y)\Big)\right]. \tag{C.18}$$

where $\tilde{f}_\theta$ is a randomized neural network with noise injection as described in Section C.2.2, and $\|\cdot\|_p$ define which kind of AT is used. For each setting, we consider two noise distributions, Gaussian and Uniform as we did with NI. We also consider two different Adversarial training AT-ℓ_∞ as well as AT-ℓ_2.

The results of RAT are reported in Table C.5 (Columns: RAT-ℓ_∞ and RAT-ℓ_2). We can observe that RAT-ℓ_∞ offers the best extra robustness with both noises, which is consistent with previous experiments, since AT is generally more effective against ℓ_∞ attacks whereas NI is more effective against ℓ_2-attacks. Overall, RAT-ℓ_∞ and a noise from uniform distribution offers the best performances but is still weaker than MAT-Rand. These results are also consistent with the literature, since adversarial training (and its variants) is the best defense against adversarial examples so far.

C.5 Conclusion & Perspective

In this paper, we tackled the problem of protecting neural networks against multiple attacks crafted from different norms. We demonstrated and gave a geometrical interpretation to explain why most defense mechanisms can only protect against one type of attack. Then we reviewed existing strategies that mix defense mechanisms in order to build models that are robust against multiple adversarial attacks. We conduct a rigorous and full comparison of *Randomized Adversarial Training* and *Mixed Adversarial Training* as defenses against multiple attacks.

We could argue that both techniques offer benefits and limitations. We have observed that MAT offers the best empirical robustness against multiples adversarial attacks but this technique is computationally expensive which hinders its use in large-scale applications. Randomized techniques have the important advantage of providing theoretical guarantees of robustness and being computationally cheaper. However, the certificate provided by such defenses is still too small for strong attacks. Furthermore, certain Randomized defenses also suffer from the curse of dimensionality as recently shown by Kumar et al. [2020a].

Although, randomized defenses based on noise injection seem limited in terms of accuracy under attack and scalability, they could be improved either by Learning the best distribution to use or by leveraging different types of randomization such as discrete randomization first proposed in Pinot et al. [2020]. We believe that these certified defenses are the best solution to ensure the robustness of classifiers deployed into real-world applications.

D Adversarial Attacks on Linear Contextual Bandits

Contextual bandit algorithms are applied in a wide range of domains, from advertising to recommender systems, from clinical trials to education. In many of these domains, malicious agents may have incentives to force a bandit algorithm into a desired behavior. For instance, an unscrupulous ad publisher may try to increase their own revenue at the expense of the advertisers; a seller may want to increase the exposure of their products, or thwart a competitor's advertising campaign. In this paper, we study several attack scenarios and show that a malicious agent can force a linear contextual bandit algorithm to pull any desired arm $T - o(T)$ times over a horizon of T steps, while applying adversarial modifications to either rewards or contexts with a cumulative cost that only grow logarithmically as $O(\log T)$. We also investigate the case when a malicious agent is interested in affecting the behavior of the bandit algorithm in a single context (e.g., a specific user). We first provide sufficient conditions for the feasibility of the attack and an efficient algorithm to perform an attack. We empirically validate the proposed approaches in synthetic and real-world datasets.

D.1 Introduction

Recommender systems are at the heart of the business model of many industries like e-commerce or video streaming Davidson et al. [2010], Gomez-Uribe and Hunt [2015]. The two most common approaches for this task are based either on matrix factorization Park et al. [2017] or bandit algorithms Li et al. [2010], which both rely on a unaltered feedback loop between the recommender system and the user. In recent years, a fair amount of work has been dedicated to understanding how targeted perturbations in the feedback loop can fool a recommender system into recommending low quality items.

Following the line of research on adversarial attacks in supervised learning Biggio et al. [2012], Goodfellow et al. [2015b], Jagielski et al. [2018], Li et al. [2016], Liu et al. [2017], attacks on recommender systems have been focused on filtering-based algorithms Christakopoulou and Banerjee [2019], Mehta and Nejdl [2008] and offline contextual bandits Ma et al. [2018]. The question of adversarial attacks for online bandit algorithms has only been studied quite recently Jun et al. [2018], Liu and Shroff [2019], Immorlica et al. [2018], Guan et al. [2020], and solely in the multi-armed stochastic setting. Although the idea of online adversarial bandit algorithms is not new (see Exp3 algorithm in Auer et al. [2002]), the focus is different from what we are considering in this article. Indeed, algorithms like Exp3 or Exp4 Lattimore and Szepesvári [2018] are designed to find optimal actions in hindsight in order to adapt to any rewards stream.

The opposition between adversarial and stochastic bandit settings has sparked interests in studying a middle ground. In Bubeck and Slivkins [2012], the learning algorithm has no knowledge of the type of feedback it receives (either stochastic or adversarial). In Lykouris et al. [2018], Li et al. [2019d], Gupta et al. [2019a], Lykouris et al. [2019], Kapoor et al. [2019], the rewards are assumed to be corrupted by adversarial rewards. The authors focus on building algorithms able to find the optimal actions even in the presence of some non-random perturbations. This setting is different from what is studied in this article because those perturbations are bounded and agnostic to arms pulled by the learning algorithm, i.e., the adversary corrupt the rewards before the algorithm chooses an arm.

In the broader Deep Reinforcement Learning (DRL) literature, the focus is placed on modifying the observations of different states to fool a DRL system at inference time Hussenot et al. [2019], Sun et al. [2020b] or the rewards Ma et al. [2019].

Contribution. In this work, we first follow the research direction opened by Jun et al. [2018] where the attacker has the objective of fooling a learning algorithm into taking a specific action as much as possible. For example in a news recommendation problem, as described in Li et al. [2010], a bandit algorithm chooses between K articles to recommend to a user, based on some information about them, called context. We assume that an attacker sits between the user and the website, they can choose the reward (i.e., click or not) for the recommended article observed by the recommending algorithm. Their goal is to fool the bandit algorithm into recommending some articles to most users. The contributions of our work can be summarized as follows:

- We extend the work of Jun et al. [2018], Liu and Shroff [2019] to the contextual linear bandit setting showing how to perturb rewards for both stochastic and adversarial algorithms, forcing **any** bandit algorithms to pull a specific set of arms, $o(T)$ times for logarithmic cost for the attacker.

- We analyze, for the first time, the setting in which the attacker can only modify the context x associated with the current user (the reward is not altered). The goal of the attacker is to fool the bandit algorithm into pulling arms of a target set for most users (i.e., contexts) while minimizing the total norm of their attacks. We show that the widely known LinUCB algorithm Abbasi-Yadkori et al. [2011], Lattimore and Szepesvári [2018] is vulnerable to this new type of attack.

- We present a harder setting for the attacker, where the latter can only modify the context associated to a specific user. This situation may occur when a malicious agent has infected some computers with a Remote Access Trojan (RAT). The attacker can then modify the history of navigation of a specific user and, as a consequence, the information seen by the online recommender system.We show how the attacker can attack the two very common bandit algorithms LinUCB and Linear Thompson Sampling (LinTS) Agrawal and Goyal [2013], Abeille et al. [2017] and, in certain cases, force them to pull a set of arms most of the time when a specific context (i.e., user) is presented to the algorithm (i.e., visits a website).

D.2 Preliminaries

We consider the standard contextual linear bandit setting with $K \in \mathbb{N}$ arms. At each time t, the agent observes a context $x_t \in \mathbb{R}^d$, selects an action $a_t \in [\![1, K]\!]$ and observes a reward: $r_{t,a_t} = \langle \theta_{a_t}, x_t \rangle + \eta^t_{a_t}$ where for each arm a, $\theta_a \in \mathbb{R}^d$ is a feature vector and $\eta^t_{a_t}$ is a conditionally independent zero-mean, σ^2-subgaussian noise. The contexts are assumed to be sampled *stochastically* except in App. D.11.

Assumption 2. *There exist* $L > 0$ *and* $\mathcal{D} \subset \mathbb{R}^d$*, such that for all* t*,* $x_t \in \mathcal{D}$ *and,* $\forall x \in \mathcal{D}, \forall a \in [\![1, K]\!]$, $\|x\|_2 \leq L$ *and* $\langle \theta_a, x \rangle \in (0, 1]$. *In addition, we assume that there exists* $S > 0$ *such that* $\|\theta_a\|_2 \leq S$ *for all arms* a.

The agent minimizes the cumulative regret after T steps $R_T = \sum_{t=1}^{T} \langle \theta_{a^\star_t}, x_t \rangle - \langle \theta_{a_t}, x_t \rangle$, where $a^\star_t := \mathrm{argmax}_a \langle \theta_a, x_t \rangle$. A bandit learning algorithm $\mathfrak{A}$ is said to be *no-regret* when it satisfies $R_T = o(T)$, i.e., the average expected reward received by $\mathfrak{A}$ converges to the optimal one. Classical bandit algorithms (e.g., LinUCB and LinTS) compute an estimate of the unknown parameters θ_a using past observations. Formally, for each arm $a \in [K]$ we define S^t_a as the set of times up to $t-1$ (included) where the agent played arm a. Then, the estimated parameters are obtained through regularized least-squares regression as $\widehat{\theta}^t_a = (X_{t,a} X_{t,a}^\top + \lambda I)^{-1} X_{t,a} Y_{t,a}$, where $\lambda > 0$, $X_{t,a} = (x_i)_{i \in S^t_a} \in \mathbb{R}^{d \times |S^t_a|}$ and $Y_{t,a} = (r_{i,a_i})_{i \in S^t_a} \in \mathbb{R}^{|S^t_a|}$. Denote by $V_{t,a} = \lambda I + X_{t,a} X_{t,a}^\top$ the design matrix of the regularized least-square problem and by $\|x\|_V = \sqrt{x^\top V x}$ the weighted norm w.r.t. any positive matrix $V \in \mathbb{R}^{d \times d}$. We define the confidence set:

$$\mathcal{C}_{t,a} = \left\{ \theta \in \mathbb{R}^d \, : \, \left\| \theta - \widehat{\theta}_{t,a} \right\|_{V_{t,a}} \leq \beta_{t,a} \right\} \tag{D.1}$$

where $\beta_{t,a} = \sigma \sqrt{d \log \left((1 + L^2 (1 + |S^t_a|)/\lambda)/\delta \right)} + S\sqrt{\lambda}$, which guarantees that $\theta_a \in \mathcal{C}_{t,a}$, for all $t > 0$, w.p. $1 - \delta$. This uncertainty is used to balance the exploration-exploitation trade-off either through optimism (e.g., LinUCB) or through randomization (e.g., LinTS).

D.3 Online Adversarial Attacks on Rewards

The ultimate goal of a malicious agent is to force a bandit algorithm to perform a desired behavior. An attacker may simply want to induce the bandit algorithm to perform poorly—ruining the users' experience—or to force the algorithm to suggest a specific arm. The latter case is particularly interesting in advertising where a seller may want to increase the exposure of its product at the expense of the competitors. Note that the users' experience is also compromised by the latter attack since the suggestions they will receive will not be tailored to their needs. Similarly to Liu and Shroff [2019], Jun et al. [2018], we focus on the latter objective, i.e., to fool the bandit algorithm into pulling arms in $A^\dagger$, a set of target arms, for $T - o(T)$ time steps (*independently of the user*).

A way to obtain this behavior is to dynamically modify the reward in order to make the bandit algorithm believe that $a^\dagger$ is optimal, for some $a^\dagger \in A^\dagger$. Clearly, the attacker has to pay a price in order to modify the perceived bandit problem and fool the algorithm. If there is no restriction on when and how the attacker can alter the reward, the attacker can easily fool the algorithm. However, this setting is not interesting since the attacker may pay a cost higher than the loss suffered by the attacked algorithm. An attack strategy is considered successful when the total cost of the attack is sublinear in T.

In this section, we show that under Assumption 2, there exists an attack algorithm that is successful against any bandit algorithm, stochastic or adverserial.

Setting. We assume that the attacker has the same knowledge as the bandit algorithm $\mathfrak{A}$ about the problem (i.e., knows σ and L). The attacker is assumed to be able to observe the context x_t, the arm a_t pulled by $\mathfrak{A}$, and can modify the reward received by $\mathfrak{A}$. When the attacker modifies the reward r_{t,a_t} into $\widetilde{r}_{t,a_t}$ the *instantaneous cost* of the attack is defined as $c_t := \left| r_{t,a_t} - \widetilde{r}_{t,a_t} \right|$. The goal of the attacker is to fool algorithm $\mathfrak{A}$ such that the arms in $A^\dagger$ are pulled $T - o(T)$ times and $\sum_{t=1}^{T} c_t = o(T)$. We also assume that the action for the arms in the target set is strictly positive for every context $x \in \mathcal{D}$. That is to say that $\Delta := \min_{x\in\mathcal{D}} \left\{ \langle x, \theta_{a_\star^\dagger(x)} \rangle - \max_{a\in A^\dagger, a\neq a_\star^\dagger(x)} \langle x, \theta_a \rangle \right\} > 0$ where $a_\star^\dagger(x) = \arg\max_{a\in A^\dagger} \langle x, \theta_a \rangle$ for every $x \in \mathcal{D}$.

Attack idea. We leverage the idea presented in Liu and Shroff [2019] and Jun et al. [2018] where the attacker lowers the reward of arms $a \notin A^\dagger$ so that algorithm $\mathfrak{A}$ learns that an arm of the target set is optimal for every context. Since $\mathfrak{A}$ is assumed to be no-regret, the attacker only needs to modify the rewards $o(T)$ times to achieve this goal. Lowering the rewards has the effect of shifting the vectors $(\theta_a)_{a\notin A^\dagger}$ to new vectors $(\theta'_a)_{a\notin A^\dagger}$ such that for all arms $a \notin A^\dagger$ and all contexts $x \in \mathcal{D}$, there exists an arm $a^\dagger \in A^\dagger$ such that $\langle \theta'_a, x \rangle \leq \langle \theta_{a^\dagger}, x \rangle$. Since rewards are assumed to be bounded (see Asm. 2), this objective can be achieved by simply forcing the reward of non-target arms $a \notin A^\dagger$ to the minimum value. Contextual ACE (see Fig. D.2) implements a soft version of this idea by leveraging the knowledge of the reward distribution. At each round t, Contextual ACE modifies the reward perceived by $\mathfrak{A}$ as follows:

$$\widetilde{r}^1_{t,a_t} = \eta'_t \mathbb{1}_{\{a_t \notin A^\dagger\}} + r_{t,a_t} \mathbb{1}_{\{a_t \in A^\dagger\}} \tag{D.2}$$

where η'_t is a σ-subgaussian random variable generated by the attacker independently of all other random variables. Contextual ACE transforms the original problem into a *stationary* bandit problem in which there is a targeted arm that is optimal for all contexts and all non targeted arms have expected reward of 0. The following propostion shows that the cumulative cost of the attack is sublinear.

Proposition 24. *For any $\delta \in (0, 1/K]$, when using Contextual ACE algorithm (Fig. D.1) with perturbed rewards $\widetilde{r}^1$, with probability at least $1 - K\delta$, algorithm $\mathfrak{A}$ pulls an arm in $A^\dagger$ for $T - o(T)$ time steps and the total cost of attacks is $o(T)$.*

The proof of this proposition is provided in App. D.8.1. While Prop. 24 holds for any no-regret algorithm $\mathfrak{A}$, we can provide a more precise bound on the total cost by inspecting the algorithm. For example, we can show (see App. D.12), that, with probability at least $1 - K\delta$, the number of times LinUCB Abbasi-Yadkori et al. [2011] pulls arms not in $A^\dagger$ is at most $\sum_{j\notin A^\dagger} N_j(T) \leq \frac{64K\sigma^2\lambda S^2}{\Delta^2} \left(d \log \left(\frac{\lambda + \frac{TL^2}{d}}{\delta^2} \right) \right)^2$. This directly translates into a bound on the total cost.

Comparison with ACE Liu and Shroff [2019]. In the stochastic setting, the ACE algorithm Liu and Shroff [2019] leverages a bound on the expected reward of each arm in order to modify the reward. However, the perturbed reward process seen by algorithm $\mathfrak{A}$ is non-stationary and in general there is no guarantee that an algorithm minimizing the regret in a stationary bandit problem keeps the same performance when the bandit problem is not stationary anymore. Nonetheless, transposing the idea of the ACE algorithm to our setting would give an attack of the following form, where at time t, Alg. $\mathfrak{A}$ pulls arm a_t and receives rewards $\widetilde{r}^2_{t,a_t}$:

$$\widetilde{r}^2_{t,a_t} = (r_{t,a_t} + \max(-1, \min(0, C_{t,a_t})))\mathbb{1}_{\{a_t \notin A^\dagger\}} + r_{t,a_t}\mathbb{1}_{\{a_t \in A^\dagger\}}$$

with $C_{t,a_t} = (1-\gamma)\min_{a^\dagger \in A^\dagger} \min_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle \theta, x_t \rangle - \max_{\theta \in \mathcal{C}_{t,a_t}} \langle \theta, x_t \rangle$. Note that $\mathcal{C}_{t,a}$ is defined as in Eq. D.1 using the *non-perturbed* rewards, i.e., $Y_{t,a} = (r_{i,a_i})_{i \in S^t_a}$.

Bounded Rewards. The bounded reward assumption is necessary in our analysis to prove a formal bound on the total cost of the attacks for *any* no-regret bandit algorithm, otherwise we need more information about the attacked algorithm. In practice, the second attack on the rewards, $\widetilde{r}^2$, can be used in the case of unbounded rewards for any algorithms. The difficulty for unbounded reward is that the attacker has to adapt to the environment reward but in order to do so the reward process observed by the bandit algorithm becomes non-stationary under the attack. Thus, there is no guarantee that an algorithm like LinUCB will pull a target arm as the proof relies on the environment observed by the bandit algorithm being stationary. We observe empirically that the total cost of attack is sublinear when using $\widetilde{r}^2$.

Jun et al. [2018] does not assume that rewards are bounded but focus on attacking algorithms in the stochastic multi-armed setting. That is to say they study attacks only designed for ε-greedy and UCB while we provide an efficient attack for any algorithms in the linear contextual case. We can extend their work, and thus remove the bounded reward assumption, in the linear contextual case by using the following attack, designed only for LinUCB:

$$\widetilde{r}^3_{t,a_t} = \left(r_{t,a_t} + \min_{a^\dagger \in A^\dagger} \min_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle \theta, x_t \rangle - \max_{\theta \in \mathcal{C}_{t,a_t}} \langle \theta, x_t \rangle \right) \mathbb{1}_{\{a_t \notin A^\dagger\}} + r_{t,a_t}\mathbb{1}_{\{a_t \in A^\dagger\}} \qquad \text{(D.3)}$$

with $C_{t,a}$ defined as in Eq. (D.1). Although, the attack $\widetilde{r}^3$ is not stationary, it is possible to prove that the total cost of attack is $\mathcal{O}(\log(T))$ because we know that the attacked bandit algorithm is LinUCB.

Constrained Attack. When the attacker has a constraint on the instantaneous cost of the attack, using the perturbed reward $\widetilde{r}^1$ may not be possible as the cost of the attack at time t is not decreasing over time. Using the perturbed reward $\widetilde{r}^2$ offers a more flexible type of attack with more control on the instantaneous cost thanks to the parameter γ. But it still suffers from a minimal cost of attack from lowering rewards of arms not in $A^\dagger$.

Defense mechanism. The attack based on reward $\widetilde{r}_1$ is hardly detectable without prior knownledge about the problem. In fact, the reward process associated to $\widetilde{r}_1$ is stationary and compatible with the assumption about the true reward (e.g., subgaussian). While having very low rewards is reasonable in advertising, it can make the attack easily detectable in some other problems. On the other hand, the fact that $\widetilde{r}_2$ is a non-stationary process makes this attack easier to detect. When some data are already available on each arm, the learner can monitor the difference between the average rewards per action computed on new and old data.

D.4 Online Adversarial Attacks on Contexts

In this section, we consider the attacker to be able to alter the context x_t perceived by the algorithm rather than the reward. The attacker is now restricted to change the type of users presented to the learning algorithm $\mathfrak{A}$, hence changing its perception of the environment. We show that under the assumption that the attacker knows a lower-bound to the reward of the target set, it is possible to fool LinUCB.

Setting. As in Sec. D.3, we consider the attacker to have the same knowledge about the problem as $\mathfrak{A}$. The main difference with the previous setting is that the attacker attacks before the algorithm. We adopt a *white-box* Goodfellow et al. [2015b] setting attacking LinUCB. The goal of the attacker is unchanged: they aim at forcing the algorithm to pull arms in $A^\dagger$ for $T - o(T)$ time steps while paying a sublinear total cost. We denote by $\widetilde{x}_t$ the context after the attack and by $c_t = \|x_t - \widetilde{x}_t\|_2$ the instantaneous cost.

Difference between attacks on contexts and rewards. Perturbing contexts is fundamentally different from perturbing the rewards. The attacker only modifies the context that is shown to the bandit algorithm. The true context, which is used to compute the reward, remains unchanged. In other words, the attacker cannot modify the reward observed by the bandit algorithm. Instead, the attack algorithm described in this section fools the bandit algorithm by making the rewards appear small relative to the contexts and requires more assumptions on the bandit algorithm than in Sec. D.3.

Attack Idea. The idea of the attack in this setting is similar to the attack of Sec. D.3. The attacker builds a bandit problem where arm an $a^\dagger \in A^\dagger$ is optimal for all contexts by lowering the perceived value of all other arms not in $A^\dagger$. The attacker cannot modify the reward but, thanks to the linear reward assumption, they can scale the contexts to decrease the predicted rewards in the original context.

At time t, the attacker receives the context x_t and computes the attack. Thanks to the white-box setting, it computes the arm a_t that algorithm $\mathfrak{A}$ would pull if presented with context x_t. If $a_t \notin A^\dagger$ then the attacker changes the context to $\widetilde{x}_t = \alpha_{a_t} x_t$ with $\alpha_{a_t} > \max_{x \in \mathcal{D}} \min_{a^\dagger \in A^\dagger} \langle \theta_{a_t}, x \rangle / \langle \theta_{a^\dagger}, x \rangle$.This factor is chosen such that for a ridge regression computed on the dataset $(\alpha x_i, \langle \theta, x_i \rangle)_i$ outputs a parameter close to θ/α therefore the attacker needs to choose α such that for every context $x \in \mathcal{D}$, $\langle x, \theta/\alpha \rangle \leq \max_{a^\dagger \in A^\dagger} \langle x, \theta_{a^\dagger}, x \rangle$. In other words, the attacker performs a dilation of the incoming context every time algorithm $\mathfrak{A}$ does not pull an arm in $A^\dagger$. The fact that the decision rule used by LinUCB is invariant by dilation guarantees that the attacker will not inadvertently lower the perceived rewards for arms in $A^\dagger$. Because the rewards are assumed to be linear, presenting a large context αx and receiving the reward associated with the normal context x will skew the estimated rewards of LinUCB. The attack protocol is summarized in Fig. D.2.

In order to compute the parameter α used in the attack, we make the following assumption concerning the performance of the arms in the target set:

Assumption 3. *For all $x \in \mathcal{D}$, there exists $a^\dagger \in A^\dagger$, such that $0 < \nu \leq \langle x, \theta_{a^\dagger} \rangle$ and ν is known to the attacker.*

Knowing ν. For advertising and recommendation systems, knowing ν is not problematic. Indeed in those cases, the reward is the probability of impression of the ad ($r \in [0, 1]$). The attacker has the freedom to choose one of multiple target arms with strictly positive click probability in

For time $t = 1, 2, ..., T$ **do**

1. Alg. $\mathfrak{A}$ chooses arm a_t based on context x_t
2. Environment generates reward: $r_{t,a_t} = \langle \theta_{a_t}, x_t \rangle + \eta_t$ with $\eta^t_{a_t}$ conditionally σ^2-subgaussian
3. Attacker observes reward r_{t,a_t} and feeds the perturbed reward $\widetilde{r}^1_{t,a_t}$ (or $\widetilde{r}^2_{t,a_t}$) to $\mathfrak{A}$

Figure D.1: Contextual ACE algorithm

Input: attack parameter: α
For time $t = 1, 2, ..., T$ **do**

1. Attacker observes the context x_t, computes potential arm a'_t and sets $\widetilde{x}_t = x_t + (\alpha(x_t) - 1)x_t \mathbb{1}_{\{a'_t \notin A^\dagger\}}$
2. Alg. $\mathfrak{A}$ chooses arm a_t based on context $\widetilde{x}_t$
3. Environment generates reward: $r_{t,a_t} = \langle \theta_{a_t}, x_t \rangle + \eta_t$ with η_t conditionally σ^2-subgaussian
4. Alg. $\mathfrak{A}$ observes reward r_{t,a_t}

Figure D.2: ConicAttack algorithm.

every context. This freedom is an important aspect for the attacker since it allows the attacker to cherry pick the target ad(s). In particular, the attacker can estimate ν based on data from previous campaigns (only for the target ad). For instance, a company could have run many ad campaigns for one of their products and try to get the defender's system to advertise it.

An issue is that the norm of the attacked context can be greater that the upper bound L of Assumption 2. To prevent this issue, we choose a context-dependent multiplicative constant $\alpha(x) = \min\{2/\nu, L/\|x\|_2\}$ which amounts to clip the norm of the attacked context to L. In Sec. D.6, we show that this attack is effective for different size of target arms sets. We also show that in the case of contexts such that $\|x\|_2 \leq \nu L/2$ that the cost of attacks is logarithmic in the horizon T.

Proposition 25. *Using the attack described in Fig. D.2 and assuming that $\|x\|_2 \leq \nu L/2$ for all contexts $x \in \mathcal{D}$, for any $\delta \in (0, 1/K]$, with probability at least $1 - K\delta$, the number of times* LinUCB *does not pull an arm in $A^\dagger$ before time T is at most*

$$\sum_{j \notin A^\dagger} N_j(T) \leq 32K^2 \left(\frac{\lambda}{\alpha^2} + \sigma^2 d \log\left(\frac{\lambda d + TL^2\alpha^2}{d\lambda\delta} \right) \right)^3$$

with $N_j(T)$ the number of times arm j has been pulled during the first T steps, The total cost for the attacker is bounded by: $\sum_{t=1}^T c_t \leq \frac{64K^2}{\nu} \left(\frac{\lambda}{\alpha^2} + \sigma^2 d \log\left(\frac{\lambda d + TL^2\alpha^2}{d\lambda\delta} \right) \right)^3$ with $\alpha = 2/\nu$.

The proof of Proposition 25 (see App. D.8.2) assumes that the attacker can attack at any time step, and that they can know in advance which arm will be pulled by Alg. $\mathfrak{A}$ in a given context. Thus it is not applicable to random exploration algorithms like LinTS Agrawal and Goyal [2013] and ε-greedy. We also observed empirically that thowe two randomized algorithms are more robust to attacks (see Sec. D.6) than LinUCB.

Norm Clipping. Clipping the norm of the attacked contexts is not beneficial for the attacker. Indeed, this means that an attacked context was violating the assumption (used by the bandit algorithm) that contexts are bounded by L. The attack could then be easily detectable and may succeed only because it is breaking an underlying assumption used by the bandit algorithm. Prop. 25 provides a theoretical grounding for the proposed attack when contexts are bounded by $\nu L/2$ and not only L. Although, we can not prove a bound on the cumulative cost of attacks in general, we show in Sec. D.6 that attacks are still successful for multiple datasets where contexts are not bounded by $\nu L/2$.

D.5 Offline attacks on a Single Context

Previous sections focused on the man-in-the-middle (MITM) attack either on reward or context. The MITM attack allows the attacker to arbitrarily change the information observed by the recommender system at each round. This attack may be hardly feasible in practice, since the exchange channels are generally protected by authentication and cryptographic systems. In this section, we consider the scenario where the attacker has control over a single user u. As an example, consider the case where the device of the user is infected by a malware (e.g., Trojan horse), giving full control of the system to the malicious agent. The attacker can thus modify the context of the specific user (e.g., by altering the cookies) that is perceived by the recommender system. We believe that changes to the context (e.g., cookies) are more subtle and less easily detectable than changes to the reward (e.g., click). Moreover, if the reward is a purchase, it cannot be altered easily by taking control of the user's device. Clearly, the impact of the attacker on the overall performance of the recommender system depends on the frequency of the specific user, that is out of the attacker's control. It may be thus difficult to obtain guarantees on the cumulative regret of algorithm $\mathfrak{A}$. For this reason, we mainly focus on the study of the feasibility of the attack.

The attacker targets a specific user (i.e., the infected user) associated to a context $x^\dagger$. Similarly to Sec. D.4, the objective of the attacker is to find the minimal change to the context presented to the recommender system $\mathfrak{A}$ such that $\mathfrak{A}$ selects an arm in $A^\dagger$. $\mathfrak{A}$ observes a modified context $\widetilde{x}$ instead of $x^\dagger$. After selecting an arm a_t, $\mathfrak{A}$ observes the true noisy reward $r_{t,a_t} = \langle \theta_{a_t}, x^\dagger \rangle + \eta_{a_t}^t$. We still study a white-box setting: the attacker can access all the parameters of $\mathfrak{A}$.

In this section, we show under which condition it is possible for an attacker to fool both an optimistic and posterior sampling algorithm.

D.5.1 Optimistic Algorithm: LinUCB

We consider the LinUCB algorithm which chooses the arm to pull by maximizing an upper-confidence bound on the expected reward. For each arm a and context x, the UCB value is given by $\max_{\theta \in \mathcal{C}_{t,a}} \langle x, \theta \rangle = \langle x, \hat{\theta}_a^t \rangle + \beta_{t,a} \|x\|_{\widetilde{V}_{t,a}^{-1}}$ (see Sec. **??**). The objective of the attacker is to force LinUCB to pull an arm in $A^\dagger$ once presented with context $x^\dagger$. This means to find a perturbation of context $x^\dagger$ that makes any arm in $A^\dagger$ the most optimistic arm. Clearly, we would like to keep the perturbation as small as possible to reduce the cost for the attacker and the probability of being detected. Formally, the attacker needs to solve the following *non-convex* optimization problem:

$$\min_{y \in \mathbb{R}^d} \quad \|y\|_2 \qquad \text{s.t} \qquad \max_{a \notin A^\dagger} \max_{\theta \in \widetilde{\mathcal{C}}_{t,a}} \langle x^\dagger + y, \theta \rangle + \xi \leq \max_{a^\dagger \in A^\dagger} \max_{\theta \in \widetilde{\mathcal{C}}_{t,a^\dagger}} \langle x^\dagger + y, \theta \rangle \tag{D.4}$$

where $\xi > 0$ is a parameter of the attacker and $\widetilde{\mathcal{C}}_{t,a} := \left\{ \theta \mid \|\theta - \hat{\theta}_a^t\|_{\widetilde{V}_{t,a}} \leq \beta_{t,a} \right\}$ is the confidence set constructed by LinUCB. We use the notation $\widetilde{\mathcal{C}}, \widetilde{V}$ to stress the fact that LinUCB observes only the modified context. In contrast to Sec. D.3 and D.4, the attacker may not be able to force the algorithm to pull any of the target arms in $A^\dagger$. In other words, Problem D.4 may not be feasible. However, we are able to characterize the feasibility of (D.4).

Theorem 23. *Problem* (D.4) *is feasible at time* t iff.

$$\exists \theta \in \cup_{a^\dagger \in A^\dagger} \widetilde{\mathcal{C}}_{t,a^\dagger},\ \theta \notin Conv\Big(\cup_{a \notin A^\dagger} \widetilde{\mathcal{C}}_{t,a} \Big) \tag{D.5}$$

The condition given by Theorem 23 says that this attack can be done when there exists a vector x for which an arm in $A^\dagger$ is assumed to be optimal according to LinUCB. The condition mainly stems from the fact that optimizing a linear product on a convex compact set will reach its maximum on the edge of this set. In our case this set is the convex hull of the confidence ellipsoids of LinUCB. Although it is possible to use an optimization algorithm for this class of non-convex problems—e.g., DC programming Tuy [1995]—they are still slow compared to convex algorithms. Therefore, we present a simple convex relaxation of the previous problem for a single target arm $a^\dagger \in A^\dagger$ that still enjoys some empirical performance compared to Problem (D.4). The final attack can then be computed as the minimum of the attacks obtained for each $a^\dagger \in A^\dagger$. The relaxed problem is the following for each $a^\dagger \in A^\dagger$:

$$\min_{y \in \mathbb{R}^d} \quad \|y\|_2 \qquad \text{s.t} \qquad \max_{a \neq a^\dagger, a \notin A^\dagger} \max_{\theta \in \mathcal{C}_{t,a}} \langle x^\dagger + y, \theta - \hat{\theta}^t_{a^\dagger} \rangle \leq -\xi \tag{D.6}$$

Since the RHS of the constraint in Problem (D.4) can be written as $\max_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle \theta, x^\dagger + y \rangle$ for any y, the relaxation here consists in using $\langle \theta, x^\dagger + y \rangle$ as a lower-bound to this maximum for any $\theta \in \mathcal{C}_{t,a^\dagger}$.

For the relaxed Problem (D.6), the same type of reasoning as for Problem (D.4) gives that Problem (D.6) is feasible if and only if $\hat{\theta}_{a^\dagger}(t) \notin \mathrm{Conv}\Big(\bigcup_{a \neq a^\dagger, a \notin A^\dagger} \mathcal{C}_{t,a} \Big)$.

If Condition (D.5) is not met, no arm $a^\dagger \in A^\dagger$ can be pulled by LinUCB. Indeed, the proof of Theorem 23 shows that the upper-confidence of every arm in $A^\dagger$ is always dominated by another arm for any context. In other words, if any arm in $A^\dagger$ is optimal for some contexts then the condition is satisfied a linear number of times for LinUCB (for formal proof of this fact see App. D.8.4).

D.5.2 Random Exploration Algorithm: LinTS

The previous subsection focused on LinUCB, however we can obtain similar guarantees for algorithms with random exploration such as LinTS. In this case, it is not possible to guarantee that a specific arm will be pulled for a given context because of the randomness in the arm selection process. The objective is to guarantee that an arm from $A^\dagger$ is pulled with probability at least $1-\delta$. Similarly to the previous subsection, the problem of the attacker can be written as:

$$\min_{y \in \mathbb{R}^d} \quad \|y\| \qquad \text{s.t} \qquad \mathbb{P}\Big(\exists a^\dagger \in A^\dagger,\ \forall a \notin A^\dagger,\ \langle x^\dagger + y, \widetilde{\theta}_a - \widetilde{\theta}_{a^\dagger} \rangle \leq -\xi \Big) \geq 1 - \delta \tag{D.7}$$

where the $\widetilde{\theta}_a$ for different arms a are independently drawn from a normal distribution with mean $\hat{\theta}_a(t)$ and covariance matrix $\upsilon^2 \bar{V}_a^{-1}(t)$ with $\upsilon = \sigma \sqrt{9d \ln(T/\delta)}$. Solving this problem is not easy and in general not possible, even for a single arm. For a given x and arm a, the random variable $\langle x, \widetilde{\theta}_a \rangle$ is normally distributed with mean $\mu_a(x) := \langle \hat{\theta}_a(t), x \rangle$ and variance $\sigma_a^2(x) := \nu^2 ||x||^2_{\bar{V}_a^{-1}(t)}$. We can then write $\langle x, \widetilde{\theta}_a \rangle = \mu_a(x) + \sigma_a(x) Z_a$ with $(Z_a)_a \sim \mathcal{N}(0, I_K)$. For the sake of clarity, we drop the variable x when writing $\mu_a(x)$ and $\sigma_a(x)$.

Let's imagine (just for this paragraph) that $A^\dagger = \{a^\dagger\}$, then the constraint in Problem (D.7) becomes $\left[1 - \mathbb{E}_{Z_{a^\dagger}}\left(\Pi_{a \notin A^\dagger} \Phi\left(\frac{\sigma_{a^\dagger} Z_{a^\dagger} + \mu_{a^\dagger} - \mu_a}{\sigma_a}\right)\right)\right] \leq \delta$ where Φ is the cumulative distribution function of a normally distributed Gaussian random variable. Unfortunately, computing exactly this expectation is an open problem.

In the more general case where $|A^\dagger| \geq 1$, rewriting the constraints of Problem (D.7) is not possible. Following the idea of Liu and Shroff [2019], for every single target arm $a^\dagger \in A^\dagger$, a possible relaxation of the constraint in Problem (D.7) is, to ensure that there exists an arm $a^\dagger \in A^\dagger$ such that for every arm $a \notin A^\dagger$, $1 - \Phi\Big((\mu_{a^\dagger} - \mu_a - \xi)/(\sqrt{\sigma_a^2 + \sigma_{a^\dagger}^2})\Big) \leq \frac{\delta}{K - |A^\dagger|}$, where $|A^\dagger|$ is the cardinal of $A^\dagger$. Thus the relaxed version of the attack on LinTS for a single arm $a^\dagger$ is:

$$\min_{y \in \mathbb{R}^d} \|y\| \quad \text{s.t.} \quad \forall a \notin A^\dagger, \langle x^\dagger + y, \hat{\theta}_{a^\dagger} - \hat{\theta}_a \rangle - \xi \geq \nu \Phi^{-1}\Big(1 - \frac{\delta}{K - |A^\dagger|}\Big) \|x^\dagger + y\|_{\bar{V}_a^{-1} + \bar{V}_{a^\dagger}^{-1}} \tag{D.8}$$

Problem (D.8) is similar to Problem (D.6) as the constraint is also a Second Order Cone Program but with different parameters (see App. D.10). As in section D.5.1, we compute the final attack as the minimum of the attacks computed for each arm in $A^\dagger$.

D.6 Experiments

In this section, we conduct experiments on the attacks on contextual bandit problems with simulated data and two real-word datasets: MovieLens25M Harper and Konstan [2015] and Jester Goldberg et al. [2001]. The synthetic dataset and the data preprocessing step are presented in App. D.9.1.

D.6.1 Attacks on Rewards

We study the impact of the reward attack for 4 contextual algorithms: LinUCB, LinTS, ε-greedy and Exp4. As parameters, we use $L = 1$ for the maximal norm of the contexts, $\delta = 0.01$, $\upsilon = \sigma\sqrt{d \ln(t/\delta))/2}$, $\varepsilon_t = 1/\sqrt{t}$ at each time step t and $\lambda = 0.1$. We choose only a *unique target arm* $a^\dagger$. For Exp4, we use $N = 10$ experts with $N - 2$ experts returning a random arm at each time, one expert choosing arm $a^\dagger$ every time and one expert returning the optimal arm for every context. With this set of experts the regret of bandits with expert advice is the same as in the contextual case. To test the performance of each algorithm, we generate 40 random contextual bandit problems and run each algorithm for $T = 10^6$ steps on each. We report the average cost and regret for each of the 40 problems. Figure D.4 (Top) shows the attacked algorithms using the attacked reward $\widetilde{r}^1$ (reported as "stationary CACE") and the rewards $\widetilde{r}^2$ (reported as CACE).

These experiments show that, even though the reward process is non-stationary, usual stochastic algorithms like LinUCB can still adapt to it and pull the optimal arm for this reward process (which is arm $a^\dagger$). The true regret of the attacked algorithms is linear as $a^\dagger$ is not optimal for all contexts. In the synthetic case, for the algorithms attacked with the rewards $\widetilde{r}^2$, over 1M iterations and $\gamma = 0.22$, the target arm is drawn more than 99.4% of the time on average for every algorithm and more than 97.8% of the time for the stationary attack $\widetilde{r}^1$ (see Table D.7 in App. D.9.2). The dataset-based environments (see Figure D.4 (Left)) exhibit the same behavior: the target arm is pulled more than 94.0% of the time on average for all our attacks on Jester and MovieLens and more than 77.0% of the time in the worst case (for LinTS attacked with the stationary rewards) (see Table D.7).

D.6.2 Attacks on Contexts

We now illustrate the effectiveness of the attack in Alg. D.2. We study the behavior of attacked LinUCB, LinTS, ε-greedy with different size of target arms set ($|A^\dagger|/K \in \{0.3, 0.6, 0.9\}$ with K the total number of arms). We test the performance of LinUCB with the same parameters as in the previous experiments. Yet since the variance is much smaller in this case, we generate a random problem and run 20 simulations for each algorithm. The target arms are chosen randomly and we use the exact lower-bound on the reward of those arms to compute ν.

Table D.3: Percentage of iterations for which the algorithm pulled an arm in the target set $A^\dagger$ (with a target set size of $0.3K$ arms) (**Left**) Online attacks using ContextualConic (CC) algorithm. Percentages are averaged over 20 runs of 1M iterations. (**Right**) Offline attacks with exact (Full) and Relaxed optimization problem. Percentages are averaged over 40 runs of 1M iterations.

	Synthetic	Jester	Movilens
LinUCB	28.91%	26.59%	31.13%
CC LinUCB	98.55%	98.36%	99.61%
ε-greedy	25.7%	25.85%	31.78%
CC ε-greedy	89.71%	99.85%	99.92%
LinTS	27.2%	26.10%	33.24%
CC LinTS	30.93%	97.26%	98.82%

	Synthetic	Jester	MovieLens
LinUCB	0.07%	0.01%	0.39%
LinUCB Relaxed	13.76%	97.81%	4.09%
LinUCB Full	88.30%	99.98%	99.99%
ε-greedy	0.01%	0.00%	0.03%
ε-greedy Full	99.98%	99.95%	99.97%
LinTS	0.02%	0.01%	0.05%
LinTS Relaxed	18.21%	80.48%	5.56%

Table D.3 (Left) shows the percentage of times an arm in $A^\dagger$, for $|A^\dagger| = 0.3K$, has been selected by the attacked algorithm. We see that, as expected, CC LinUCB reaches a ratio of almost 1, meaning the target arms are indeed pulled a linear number of times. A more surprising result (at least not covered by the theory) is that ε-greedy exhibits the same behavior. Similarly to LinTS, ε-greedy exhibits some randomness in the action selection process. It can cause an arm $a^\dagger \in A^\dagger$ to be chosen when the context is attacked and interfere with the principle of the attack. We suspect that is what happens for LinTS. Fig. D.4 (Bottom) shows the total cost of the attacks for the attacked algorithms . Despite the fact that the estimate of $\theta_{a^\dagger}$ can be polluted by attacked samples, it seems that LinTS can still pick up $a^\dagger$ as being optimal for some dataset like MovieLens and Jester but not on the simulated dataset.

D.6.3 Offline attacks on a Single Context

We now move to the setting described in Sec. D.5 and test the same algorithms as in Sec. D.6.2. We run 40 simulations for each algorithm and each attack type. The target context $x^\dagger$ is chosen randomly and the target arm as the arm minimizing the expected reward for $x^\dagger$. The attacker is

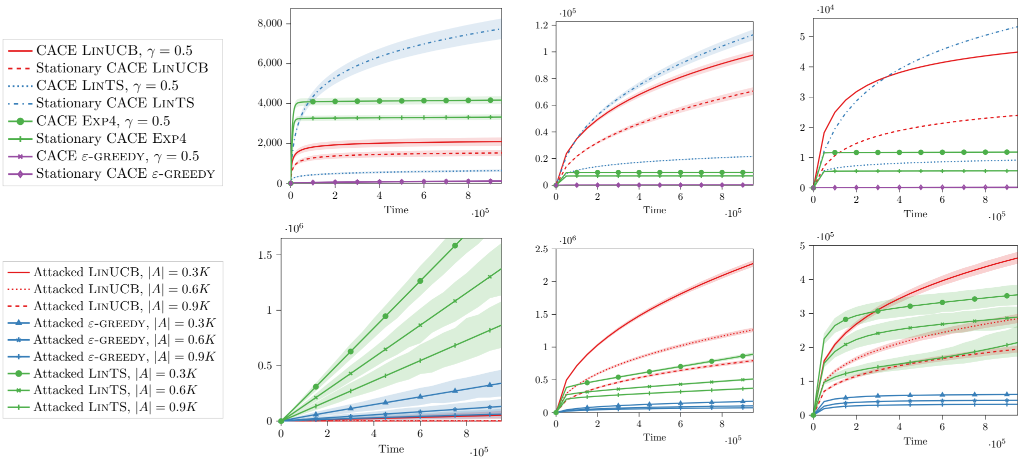


Figure D.4: Total cost of attacks on rewards for the synthetic (Left, $\gamma = 0.22$), Jester (Center, $\gamma = 0.5$) and MovieLens (Right, $\gamma = 0.5$) environments. Bottom, total cost of ContextualConic attacks on the synthetic (Left), Jester (Center) and MovieLens (Right) environments.

only able to modify the incoming context for the target context (which corresponds to the context of one user) and the incoming contexts are sampled uniformly from the set of all possible contexts (of size 100). Table D.3 (Right) shows the percentage of success for each attack. We observe that the non-relaxed attacks on ε-greedy and LinUCB work well across all datasets. However, the relaxed attack for LinUCB and LinTS are not as successful, on the synthetic dataset and MovieLens25M. The Jester dataset seems to be particularly suited to this type of attacks because the true feature vectors are well separated from the convex hull formed by the feature vectors of the other arms: only 5% of Jester's feature vectors are within the convex hull of the others versus 8% for MovieLens and 20% for the synthetic dataset. As expected, the cost of the attacks is linear on all the datasets (see Figure D.8 in App. D.9.4). The cost is also lower for the non-relaxed than for the relaxed version of the attack on LinUCB. Unsurprisingly, the cost of the attacks on LinTS is the highest due to the need to guarantee that $a^\dagger$ will be chosen with high probability (95% in our experiments).

D.7 Conclusion

We presented several settings for online attacks on contextual bandits. We showed that an attacker can force any contextual bandit algorithm to almost always pull an arbitrary target arm $a^\dagger$ with only sublinear modifications of the rewards. When the attacker can only modify the contexts, we prove that LinUCB can still be attacked and made to almost always pull an arm in $A^\dagger$ by adding sublinear perturbations to the contexts. When the attacker can only attack a single context, we derive a feasibility condition for the attacks and we introduce a method to compute some attacks of small instantaneous cost for LinUCB, ε-greedy and LinTS. To the best of our knowledge, this paper is the first to describe effective attacks on the contexts of contextual bandit algorithms. Our numerical experiments, conducted on both synthetic and real-world data, validate our results and show that the attacks on all contexts are actually effective on several algorithms and with more permissible settings.

D.8 Appendix: Proofs

In this appendix, we present the proofs of different theoretical results presented in the paper.

D.8.1 Proof of Proposition 24

Proposition. *For any $\delta \in (0, 1/K]$, when using Contextual ACE algorithm (Alg. D.1) with perturbed rewards $\tilde{r}^1$, with probability at least $1 - K\delta$, algorithm $\mathfrak{A}$ pulls an arm in $A^\dagger$ for $T - o(T)$ time steps and the total cost of attacks is $o(T)$.*

Proof. Let us consider the contextual bandit problem $\mathcal{A}_1$, with K arms with contexts $x \in \mathcal{D}$ such that every arm in $a^\dagger \in A^\dagger$ has mean reward $\langle \theta_{a^\dagger}, x \rangle$ and all other arms has mean 0. Then the regret of algorithm $\mathfrak{A}$ for this bandit problem is upper-bounded with probability at least $1 - \delta$ by a function $f_{\mathfrak{A}}(T)$ such that $f_{\mathfrak{A}}(T) = o(T)$. In addition, the reward process fed to Alg. $\mathfrak{A}$ by the attacker is a stationary reward process with σ^2-subgaussian noise. Therefore, the number of times algorithm $\mathfrak{A}$ pulls an arm not in $A^\dagger$ is upper-bounded by $f_{\mathfrak{A}}(T)/\min_{x \in \mathcal{D}} \Delta(x)$ where for every context $x \in \mathcal{D}$, let $a_\star^\dagger(x) := \arg\max_{a \in A^\dagger} \langle x, \theta_a \rangle$ and $\Delta(x) = \langle x, \theta_{a_\star^\dagger(x)} \rangle - \max_{a \in A^\dagger, a \neq a_\star^\dagger(x)} \langle x, \theta_a \rangle$.

In addition, the total cost of the attack is upper-bounded by $\max_{a \in [\![1,K]\!]} \max_{x \in \mathcal{D}} |\langle x, \theta_a \rangle| (T - N_{A^\dagger}(T))$ where $N_{A^\dagger}(T)$ is the number of times an arm in $A^\dagger$ has been pulled up to time T. Thanks to the previous argument, $T - N_{A^\dagger}(T) \leq f_{\mathfrak{A}}(T)/\min_{x \in \mathcal{D}} \Delta(x)$. □

D.8.2 Proof of Proposition 25

Proposition. *Using the attack described in Alg. D.2, for any $\delta \in (0, 1/K]$, with probability at least $1 - K\delta$, the number of times LinUCB does not pull an arm in $A^\dagger$ is at most:*

$$\sum_{j \notin A^\dagger} N_j(T) \leq 32K^2 \left(\frac{\lambda}{\alpha^2} + \sigma^2 d \log\left(\frac{\lambda d + T L^2 \alpha^2}{d \lambda \delta} \right) \right)^3$$

with $N_j(T)$ the number of times arm j has been pulled after T steps, $||\theta_a|| \leq S$ for all arms a, λ the regularization parameter of LinUCB and for all $x \in \mathcal{D}$, $||x||_2 \leq L$. The total cost for the attacker is bounded by:

$$\sum_{t=1}^{T} c_t \leq \frac{64K^2}{\nu} \left(\frac{\lambda}{\alpha^2} + \sigma^2 d \log\left(\frac{\lambda d + T L^2 \alpha^2}{d \lambda \delta} \right) \right)^3$$

Proof. Let a_t be the arm pulled by LinUCB at time t. For each arms a, let $\tilde{\theta}_a(t)$ be the result of the linear regression with the attacked context and $\hat{\theta}_a(t, \lambda/\alpha^2)$ the one with the unattacked context and a regularization of $\frac{\lambda}{\alpha^2}$. At any time step t, we can write, for all $a \notin A^\dagger$:

$$\begin{aligned}\tilde{\theta}_a(t) &= \left(\lambda I_d + \sum_{l=0, a_l=a}^{t} \alpha^2 x_l x_l^\intercal\right)^{-1} \sum_{k=0, a_k=a}^{t} r_k \alpha x_k \\ &= \frac{1}{\alpha}\left(\frac{\lambda}{\alpha^2} I_d + \sum_{k=0, a_k=a}^{t} x_k x_k^\intercal\right)^{-1} \sum_{k=0, a_k=a}^{t} r_k x_k \\ &= \frac{\hat{\theta}_a(t, \lambda/\alpha^2)}{\alpha}\end{aligned}$$

We also note that, since the contexts are not modified for arms in $a^\dagger \in A^\dagger$: $\tilde{\theta}_{a^\dagger}(t) = \hat{\theta}_{a^\dagger}(t, \lambda)$. In addition, for any context x and arm $a \notin A^\dagger$, the exploration term used by LinUCB becomes:

$$||x||_{\tilde{V}_{a,t}^{-1}} = \frac{1}{\alpha} ||x||_{\hat{V}_{a,t}^{-1}} \tag{D.9}$$

where $\tilde{V}_{a,t} = \lambda I_d + \sum_{l=0, a_l=a}^{t} \alpha^2 x_l x_l^\intercal$ and $\hat{V}_{a,t}^{-1} = \lambda/\alpha^2 I_d + \sum_{k=0, a_k=a}^{t} x_k x_k^\intercal$. For a time t, if presented with context x_t LinUCB pulls arm $a_t \notin A^\dagger$, we have:

$$\alpha\left(\left\langle \hat{\theta}_{a^\dagger}(t), x_t \right\rangle + \beta_{a^\dagger}(t)||x_t||_{V_{a^\dagger,t}^{-1}}\right) \leq \left\langle \hat{\theta}_{a_t}(t, \lambda/\alpha^2), x_t \right\rangle + \beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}}$$

As $\alpha = \frac{2}{\nu} \geq \min_{a^\dagger \in A^\dagger} \frac{2}{\langle \theta_{a^\dagger}, x_t \rangle}$, we deduce that on the event that the confidence sets (Theorem 2 in Abbasi-Yadkori et al. [2011]) hold for arm $a^\star$:

$$2 \leq \left\langle \hat{\theta}_{a_t}(t, \lambda/\alpha^2), x_t \right\rangle + \beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}} \leq \langle \theta_{a_t}, x_t \rangle + 2\beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}}$$

Thus, $1 \leq 2 - \langle \theta_{a_t}, x_t \rangle \leq 2\beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}}$. Therefore,

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{1}_{\{a_t \notin A^\dagger\}} &\leq \sum_{t=1}^{T} \min(2\beta_{a_t}(t)||x_t||_{\hat{V}_{a_t,t}^{-1}}, 1)\mathbb{1}_{\{a_t \notin A^\dagger\}} \\ &\leq \sum_{j \notin A^\dagger} 2\beta_j(T) \sqrt{\sum_{t=1}^{T} \mathbb{1}_{\{a_t = j\}} \sum_{t=1, a_t=j}^{T} \min(1, ||x_t||^2_{\hat{V}_{j,t}^{-1}})}\end{aligned}$$

But using Lemma 11 from Abbasi-Yadkori et al. [2011] and the bound on the $\beta_j(T)$ for all arms j, we have with Jensen inequality:

$$\sum_{t=1}^{T} \mathbb{1}_{\{a_t \notin A^\dagger\}} \leq 4\sqrt{K \sum_{t=1}^{T} \mathbb{1}_{\{a_t \notin A^\dagger\}} d \log\left(1 + \frac{\alpha^2 T L^2}{\lambda d}\right)}$$
$$\times \left(\sqrt{\lambda/\alpha^2} S + \sigma\sqrt{2\log(1/\delta) + d\log\left(1 + \frac{\alpha^2 T L^2}{\lambda d}\right)}\right)$$

□

D.8.3 Proof of Theorem 23

Theorem. *For any $\xi > 0$, Problem (D.4) is feasible if and only if:*

$$\exists \theta \in \bigcup_{a^\dagger \in A^\dagger} \mathcal{C}_{t,a^\dagger}, \qquad \theta \notin Conv\left(\bigcup_{a \notin A^\dagger} \mathcal{C}_{t,a}\right) \tag{D.10}$$

where for every arm a, $\mathcal{C}_{t,a} := \{\theta \mid ||\theta - \hat{\theta}_a(t)||_{\tilde{V}_{a,t}} \leq \beta_a(t)\}$ with $\hat{\theta}_a(t)$ the least squares estimate for arm a built by LinUCB and

$$\tilde{V}_{a,t} = \lambda I_d + \sum_{l=1, x_l \neq x^\dagger}^{t} \mathbb{1}_{\{a_l = a\}} x_l x_l^\intercal + \sum_{l=1, x_l = x^\dagger}^{t} \mathbb{1}_{\{a_l = a\}} \tilde{x}_l \tilde{x}_l^\intercal$$

the design matrix of LinUCB at time t for all arms a (where $\tilde{x}_l$ is the modified context)

Proof. The proof of Theorem 23 is decomposed in two parts.

First, let us assume that Equation (D.10) is satisfied. Then, let us define $a^\dagger \in A^\dagger$ such that $\theta \in \mathcal{C}_{t,a^\dagger} \setminus \mathrm{Conv}(\bigcup_{a \notin A^\dagger} \mathcal{C}_{t,a})$, then by the theorem of separation of convex sets applied to $\mathcal{C}_{t,a^\dagger}$ and $\{\theta\}$. There exists a vector v and $c_1 < c_2$ such that for all $y \in \mathrm{Conv}\left(\bigcup_{a \neq a^\dagger} \mathcal{C}_{t,a}\right)$:

$$\langle y, v \rangle \leq c_1 < c_2 \leq \langle \theta, v \rangle.$$

Hence, for $\xi > 0$ we have that for $\tilde{v} = \frac{\xi}{c_2 - c_1} v$ that:

$$\langle y, \tilde{v} \rangle + \xi \leq \langle \theta, \tilde{v} \rangle$$

So the problem is feasible.

Secondly, let us assume that an attack is feasible. Then there exists a vector y such that:

$$\max_{a^\dagger \in A^\dagger} \max_{\theta \in \mathcal{C}_{t,a^\dagger}} \langle y, \theta \rangle > c_1 := \max_{a \notin A^\dagger} \max_{\theta \in \mathcal{C}_{t,a}} \langle y, \theta \rangle \tag{D.11}$$

Let us reason by contradiction. We assume that $\bigcup_{a\in A^\dagger} \mathcal{C}_{t,a^\dagger} \subset \mathrm{Conv}\big(\bigcup_{a\notin A^\dagger} \mathcal{C}_{t,a}\big)$ and consider

$$\theta^* \in \bigcup_{a\in A^\dagger} \mathcal{C}_{t,a^\dagger} \text{ such that } \langle y, \theta^* \rangle = \max_{a^\dagger\in A^\dagger} \max_{\theta\in\mathcal{C}_{t,a^\dagger}} \langle y, \theta \rangle$$

As we assumed $\bigcup_{a\in A^\dagger} \mathcal{C}_{t,a^\dagger} \subset \mathrm{Conv}\big(\bigcup_{a\notin A^\dagger} \mathcal{C}_{t,a}\big)$, there exists $n \in \mathbb{N}^\star$, $\lambda_1, \cdots, \lambda_n \geq 0$ and $\theta_1, \cdots, \theta_n \in \bigcup_{a\notin A^\dagger} \mathcal{C}_{t,a}$ such that

$$\theta^* = \sum_{i=1}^n \lambda_i \theta_i \text{ and } \sum_{i=1}^n \lambda_i = 1$$

Thus

$$\langle y, \theta^* \rangle = \sum_i \lambda_i \langle y, \theta_i \rangle \leq c_1 \sum_{i=1}^n \lambda_i = c_1 \tag{D.12}$$

We assumed that the problem is feasible, so $c_1 < \langle y, \theta^* \rangle$ according to Eq. D.11. It contradicts Eq. D.12. □

D.8.4 Condition of Sec. D.5

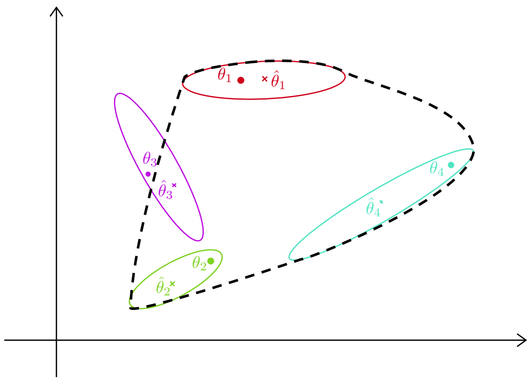


Figure D.5: Illustrative example of condition (D.5). The target arm is arm 3 or 5 and the dashed black line is the convex hull of the other confidence sets. The ellipsoids are the confidence sets $\mathcal{C}_{t,a}$ for each arm a. If we consider only arms $\{1, 2, 4, 5\}$, and we use 5 as the target arm, the condition (D.5) is satisfied as there is a θ outside the convex hull of the other confidence sets. On the other hand, if we consider arms $\{1, 2, 3, 4\}$ and we use 3 as the target arm, the condition is not satisfied anymore.

Let us assume that there is an arm in $a^\dagger \in A^\dagger$ which is optimal for some contexts. More formally, there exists a subspace $V \subset \mathcal{D}$ such that:

$$\forall x \in V, \exists a^\dagger_\star(x) \in A^\dagger, \forall a \in [\![1, K]\!] \setminus \{a^\dagger_\star(x)\} \qquad \langle x, \theta_{a^\dagger_\star(x)} \rangle > \langle x, \theta_a \rangle.$$

We also assume that the distribution of the contexts is such that, for all t, $\mu := \mathbb{P}(x_t \in V) > 0$. Then, the regret is lower-bounded in expectation by:

$$\mathbb{E}(R_T) = \mathbb{E}\left(\sum_{t=1}^{T} \mathbb{1}_{\{x_t \in V\}}\left(\left\langle x_t, \theta_{a_\star^\dagger(x_t)} - \theta_{a_t}\right\rangle\right)\right) \geq \mu m(T) \min_{x \in V} \max_{a \neq a_\star^\dagger(x)} \langle \theta_{a_\star^\dagger(x)} - \theta_a, x \rangle$$

where $m(T)$ is the expected number of times $t \leq T$ such that condition (D.5) is not met. LinUCB guarantees that $\mathbb{E}(R_T) \leq \mathcal{O}(\sqrt{T})$ for every T. Hence,

$$m(T) \leq \mathcal{O}\left(\frac{\sqrt{T}}{\mu \min_{x \in V} \max_{a \neq a_\star^\dagger(x)} \langle \theta_{a_\star^\dagger(x)} - \theta_a, x \rangle}\right)$$

This means that, in an unattacked problem, condition (D.5) is met $T - \mathcal{O}(\sqrt{T})$ times. On the other hand, when the algorithm is attacked the regret of LinUCB is not sub-linear as the confidence bound for the target arm is not valid anymore. Hence we cannot provide the same type of guarantees for the attacked problem.

D.9 Appendix: Experiments

D.9.1 Datasets and preprocessing

We present here the datasets used in the article and how we preprocess them for numerical experiments conducted in Section D.6.

We consider two types of experiments, one on synthetic data with a contextual MAB problems with $K = 10$ arms such that for every arm a, θ_a is drawn from a folded normal distribution in dimension $d = 30$. We also use a finite number of contexts (10), each of them is drawn from a folded normal distribution projected on the unit circle multiplied by a uniform radius variable (i.i.d. across all contexts). Finally, we scale the expected rewards in $(0, 1]$ and the noise is drawn from a centered Gaussian distribution $\mathcal{N}(0, 0.01)$.

The second type of experiments is conducted in the real-world datasets Jester Goldberg et al. [2001] and MovieLens25M Harper and Konstan [2015]. Jester consists of joke ratings on a continuous scale from -10 to 10 for 100 jokes from a total of 73421 users. We use the features extracted via a low-rank matrix factorization ($d = 35$) to represent the actions (i.e., the jokes). We consider a complete subset of 40 jokes and 19181 users . Each user rates all the 40 jokes. At each time, a user is randomly selected from the 19181 users and mean rewards are normalized in $[0, 1]$. The reward noise is $\mathcal{N}(0, 0.01)$. The second dataset we use is MovieLens25M. It contains 25000095 ratings created by 162541 users on 62423 movies. We perform a low-rank matrix factorization to compute users features and movies features. We keep only movies with at least 1000 ratings, which leave us with 162539 users and 3794 movies. At each time step, we present a random user, and the reward is the scalar product between the user feature and the recommend movie feature. All rewards are scaled to lie in $[0, 1]$ and a Gaussian noise $\mathcal{N}(0, 0.01)$ is added to the rewards.

D.9.2 Attacks on Rewards

In this appendix, we present empirical evolution of the total cost and the number of draws for a unique target arm as a function of the attack parameter γ for the Contextual ACE attack with perturbed rewards $\tilde{r}^2$ on generated data.

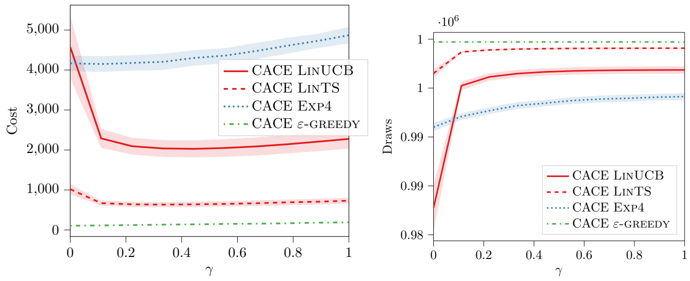


Figure D.6: Total cost of attacks and number of draws of the target arm at $T = 10^6$ as a function of γ on synthetic data

Fig. D.6 (left) shows that the total cost of attacks seems to be quite invariant w.r.t. γ except when $\gamma \to 0$ because the difference between the target arm and the other becomes negligible. This is also depicted by the total number of draws (Fig. D.6, Right) as the number of draws plummets when $\gamma \to 0$.

D.9.3 Attacks on all Contexts

Fig. ?? shows the regret for all the attacks. This figure shows that even though the total cost of attacks is linear for algorithms like LinTS in the synthetic dataset, the regret is linear. More generally, we observe that the regret is linear for all attacked algorithms on all datasets.

D.9.4 Attack on a single context

The attacks are computed by solving the optimization problems D.4 and D.6 (Sec. D.5). We choose the libraries according to their efficiency for each problem we need to solve. For Problem (D.6) and Problem (D.8) we use cvxpy Agrawal et al. [2018] and the ECOS solver. For Problem (D.4) we use the SLSQP method from the Scipy optimize library Virtanen et al. [2019] to solve the full LinUCB problem (Equation D.4) and quadprog to solve the quadratic problem to attack ε-greedy.

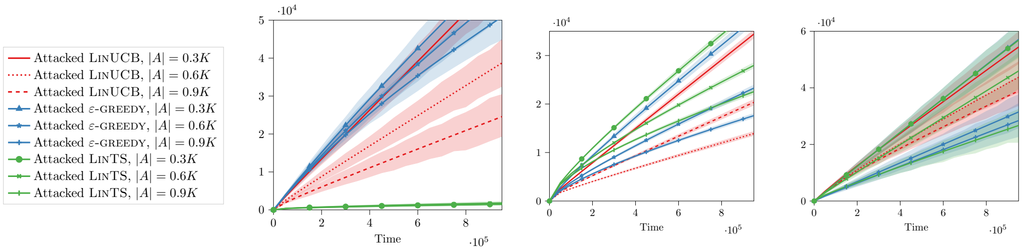
D.10 Appendix: Problem (D.8) as a Second Order Cone (SOC) Program

Problem (D.6) and Problem (D.8) are both SOC programs. We can see the similarities between both problems as follows. Let us define for every arm $a \notin A^\dagger$, the ellipsoid:

$$\mathcal{C}'_{t,a} := \left\{ y \in \mathbb{R}^d \mid ||y - \hat{\theta}_a(t)||_{A_a^{-1}(t)} \leq \upsilon \Phi^{-1}\left(1 - \frac{\delta}{K - |A^\dagger|}\right) \right\}$$

Table D.7: Number of draws of the target arm $a^\dagger$ at $T = 10^6$, for the synthetic data, $\gamma = 0.22$ for the Contextual ACE algorithm and for the Jester and MovieLens datasets $\gamma = 0.5$.

	Synthetic	Jester	Movilens
LinUCB	$86,731.6$	$23,548.16$	$25,017.31$
CACE LinUCB	$996,238.6$	$921,083.69$	$944,721.28$
Stationary CACE LinUCB	$995,578.88$	$862,095.67$	$931,531.6$
ε-greedy	$111,380.44$	$21,911.54$	$3,165.81$
CACE ε-greedy	$999,812.92$	$999,755.72$	$999,776.82$
Stationary CACE ε-greedy	$999,806.32$	$999,615.98$	$999,316.76$
LinTS	$91,664.8$	$23,398.3$	$30,189.84$
CACE LinTS	$998,997.04$	$976,708.9$	$990,250.67$
Stationary CACE LinTS	$977,850.96$	$784,715.62$	$845,512.98$
Exp4	$93,860.4$	$29,147.01$	$17,985.78$
CACE Exp4	$992,793.36$	$989,214.36$	$936,230.4$
Stationary CACE Exp4	$993,673.24$	$988,463.56$	$934,304.23$



with $A_a(t) = \tilde{V}_a^{-1}(t) + \tilde{V}_{a^\dagger}^{-1}(t)$ with $\tilde{V}_a(t)$ and $\tilde{V}_{a^\dagger}(t)$ the design matrix built by LinTS and $\hat{\theta}_a(t)$ the least squares estimate of θ_a at time t. Therefore for an arm a, the constraint in Problem (D.8) can be written for any $y \in \mathbb{R}^d$ and some arm $a^\dagger \in A^\dagger$ as:

$$\left\langle x^\star + y, \hat{\theta}_{a^\dagger}(t) \right\rangle - \xi \geq \max_{z \in \mathcal{C}'_{t,a}} \langle z, x^\star + y \rangle$$

Indeed for any $x \in \mathbb{R}^d$,

$$\begin{aligned}
\max_{y \in \mathcal{C}'_{t,a}} \langle y, x \rangle &= \left\langle x, \hat{\theta}_a(t) \right\rangle + \upsilon \Phi^{-1}\left(1 - \frac{\delta}{K - |A^\dagger|}\right) \times \max_{||A_a^{-1/2}(t)u||_2 \leq 1} \langle u, x \rangle \\
&= \left\langle x, \hat{\theta}_a(t) \right\rangle + \upsilon \Phi^{-1}\left(1 - \frac{\delta}{K - |A^\dagger|}\right) \max_{||z||_2 \leq 1} \left\langle z, A_a^{1/2}(t) x \right\rangle
\end{aligned}$$

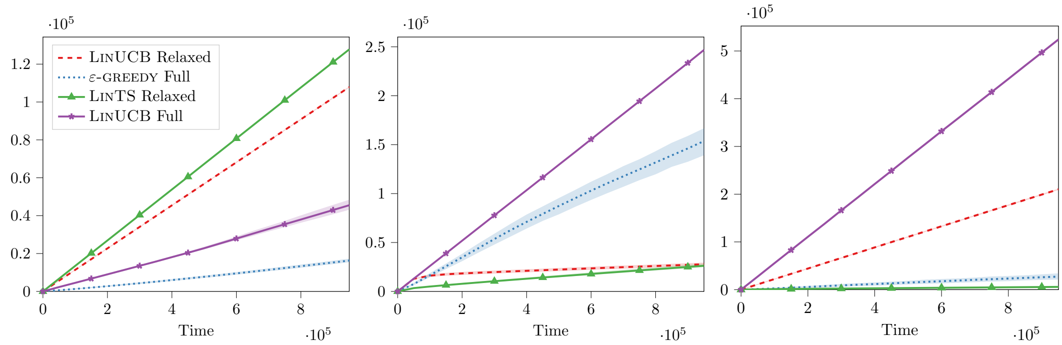


Figure D.8: Total cost of the attacks for the attacks one one context on respectively our synthetic dataset, Jester and MovieLens. As expected, the total cost is linear.

$$= \left\langle x, \hat{\theta}_a(t) \right\rangle + \upsilon \varPhi^{-1}\left(1 - \frac{\delta}{K - |A^\dagger|}\right) \|A_a^{1/2}(t)x\|_2$$

Thus, the constraint is feasible if and only if:

$$\hat{\theta}_{a^\dagger}(t) \notin \mathrm{Conv}\left(\bigcup_{a \notin A^\dagger} \mathcal{C}'_{t,a}\right)$$

D.11 Appendix: Attacks on Adversarial Bandits

In the previous sections, we studied algorithms with sublinear regret R_T, i.e., mainly bandit algorithms designed for stochastic stationary environments. Adversarial algorithms like Exp4 do not provably enjoy a sublinear **stochastic** regret R_T (as defined in the introduction) [1]. In addition, because this type of algorithms are, by design, robust to non-stationary environments, one could expect them to induce a linear cost on the attacker. In this section, we show that this is not the case for most contextual adversarial algorithms. Contextual adversarial algorithms are studied through the reduction to the bandit with expert advice problem. This is a bandit problem with K arms where at every step, N experts suggest a probability distribution over the arms. The goal of the algorithm is to learn which expert gets the best expected reward in hindsight after T steps. The regret in this type of problem is defined as $R_T^{\text{exp}} = \mathbb{E}\left(\max_{m \in [\![1,N]\!]} \sum_{t=1}^{T} \sum_{j=1}^{K} E_{m,j}^{(t)} r_{t,j} - r_{t,a_t}\right)$ where $E_{m,j}^{(t)}$ is the probability of selecting arm j for expert m. In the case of contextual adversarial bandits, the experts first observe the context x_t before recommending an expert m. Assuming the current setting with linear rewards, we can show that if an algorithm $\mathfrak{A}$, like Exp4, enjoys a sublinear regret R_T^{exp}, then, using the Contextual ACE attack with either $\tilde{r}^1$ or $\tilde{r}^2$, the attacker can fool the algorithm into pulling arm $a^\dagger$ a linear number of times under some mild assump-

[1] Exp4 enjoys a sublinear hindsight regret though. Showing a sublinear upper-bound for the stochastic regret of Exp4 is still an open problem (see Section 29.1 in Lattimore and Szepesvári [2018])

tions. However, attacking contexts for this type of algorithm is difficult because, even though the rewards are linear, the experts are not assumed to use a specific model for selecting an action.

Proposition 26. *Suppose an adversarial algorithm $\mathfrak{A}$ satisfies a regret $R_T^{\exp}$ of order $o(T)$ for any bandit problem and that there exists an expert $m^\star$ such that $T - \sum_{t=1}^T \mathbb{E}\left(E^{(t)}_{m^\star,a^\dagger_{t,\star}}\right) = o(T)$ with $a^\dagger_{t,\star}$ the optimal arim in $A^\dagger$ at time t. Then attacking alg. $\mathfrak{A}$ with Contextual ACE leads to pulling arm $a^\dagger$, $T - o(T)$ of times in expectation with a total cost of $o(T)$ for the attacker.*

Proof. Similarly to the proof of Proposition 24, let's define the bandit with expert advice problem, $\mathcal{A}_i$, such that at each time t the reward vector is $(\tilde{r}^i_{t,a})_a$ (with $i \in \{1,2\}$). The regret of this algorithm is: $\tilde{R}_T^{i,\exp} = \mathbb{E}\left(\max_{m\in[\![1,N]\!]} \sum_{t=1}^T E_m^{(t)} \tilde{r}^i_t - \tilde{r}^i_{t,a_t}\right) \in o(T)$. The regret of the learner is: $\mathbb{E}\left(\max_{m\in[\![1,N]\!]} \sum_{t=1}^T E_m^{(t)} r_t - r_{t,a_t}\right)$ where a_t are the actions taken by algorithm $\mathcal{A}_i$ to minimize $\tilde{R}_T^{i,\exp}$. Then we have:

$$\tilde{R}_T^{i,\exp} \geq \mathbb{E}\left(\sum_{t=1}^T \sum_{j=1}^K (E^{(t)}_{m^\star,j} - \mathbb{1}_{\{j=a^\dagger_{t,\star}\}})\tilde{r}^i_{t,j} + \sum_{t=1}^T \tilde{r}^i_{t,a^\dagger_{t,\star}} - \tilde{r}^i_{t,a_t}\right)$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left(\sum_{t=1}^T \tilde{r}^i_{t,a^\dagger_{t,\star}} - \tilde{r}^i_{t,a_t}\right) &\leq \tilde{R}_T^{i,\exp} + \mathbb{E}\left(\sum_{t=1}^T \sum_{j=1}^K (\mathbb{1}_{\{j=a^\dagger_{t,\star}\}} - E^{(t)}_{m^\star,j})\tilde{r}^i_{t,j}\right) \\
&\leq \tilde{R}_T^{i,\exp} + \mathbb{E}\left(\sum_{t=1}^T (1 - E^{(t)}_{m^\star,a^\dagger_{t,\star}})\tilde{r}^i_{t,j}\right) \\
&\leq \tilde{R}_T^{i,\exp} + \mathbb{E}\left(\sum_{t=1}^T (1 - E^{(t)}_{m^\star,a^\dagger_{t,\star}})\right)
\end{aligned}$$

For strategy $i = 1$, we have:

$$\mathbb{E}\left(\sum_{t=1}^T \tilde{r}^1_{t,a^\dagger_{t,\star}} - \tilde{r}^1_{t,a_t}\right) = \sum_{t=1}^T \mathbb{E}\left(r_{t,a^\dagger_{t,\star}} - \mathbb{1}_{\{a_t\in A^\dagger\}}\right) \geq \left(T - \mathbb{E}\left(\sum_{t=1}^T \mathbb{1}_{\{a_t=a^\dagger_{t,\star}\}}\right)\right)\Delta$$

where $\Delta := \min_{x\in\mathcal{D}}\left\{\langle\theta_{a^\dagger(x)}, x\rangle - \max_{a\in A^\dagger, a\neq a^\dagger(x)}\langle\theta_{a'}, x\rangle\right\}$ with $a^\dagger(x) := \arg\max_{a\in A^\dagger}\langle\theta_a, x\rangle$. Then, as $\tilde{R}_T^{1,\exp} \in o(T)$ and $\mathbb{E}\left(\sum_{t=1}^T (1 - E^{(t)}_{m^\star,a^\dagger_{t,\star}})\right) \in o(T)$, we deduce that

$$\mathbb{E}(\sum_t \mathbb{1}_{\{a_t=a^\dagger_{t,\star}\}}) = T - o(T) \quad .$$

For strategy $i = 2$, and $\delta > 0$, let us denote by E_δ the event that all confidence intervals hold with probability $1-\delta$. But on the event E_δ, for a time t where $a_t \neq a^\dagger_{t,\star}$ and such that $-1 \leq C_{t,a_t} \leq 0$:

$$\begin{aligned}\tilde{r}^2_{t,a_t} = r_{t,a_t} + C_{t,a_t} &= (1-\gamma)\min_{a^\dagger\in A^\dagger}\min_{\theta\in\mathcal{C}_{t,a^\dagger}}\langle\theta, x_t\rangle + \eta_{a_t,t} + \langle\theta_a, x_t\rangle - \max_{\theta\in\mathcal{C}_{t,a_t}}\langle\theta, x_t\rangle \\ &\leq (1-\gamma)\langle\theta_{a^\dagger_{t,\star}}, x_t\rangle + \eta_{a_t,t}\end{aligned}$$

when $C_{t,a_t} > 0$ (still on the event E_δ):

$$\tilde{r}^2_{t,a_t} = r_{t,a_t} \leq (1-\gamma)\langle\theta_{a^\dagger_{t,\star}}, x_t\rangle + \eta_{a_t,t}$$

because $C_{t,a_t} > 0$ means that $(1-\gamma)\langle\theta_{a^\dagger_{t,\star}}, x_t\rangle \geq (1-\gamma)\min_{a^\dagger\in A^\dagger}\min_{\theta\in\mathcal{C}_{t,a^\dagger}}\langle\theta, x_t\rangle \geq \max_{\theta\in\mathcal{C}_{t,a_t}}\langle\theta, x_t\rangle \geq \langle\theta_a, x_t\rangle$. But finally, when $C_{t,a_t} \leq -1$, $\tilde{r}^2_{t,a_t} = r_{t,a_t} - 1 \leq \eta_{a_t,t} \leq (1-\gamma)\langle\theta_{a^\dagger_{t,\star}}, x_t\rangle + \eta_{a_t,t}$. But on the complementary event E^c_δ, $\tilde{r}^2_{t,a_t} \leq r_{t,a_t}$. Thus, given that the expected reward is assumed to be bounded in $(0,1]$ (Assumption 2):

$$\begin{aligned}\mathbb{E}\left(\sum_{t=1}^T \tilde{r}^2_{t,a^\dagger_{t,\star}} - \tilde{r}^2_{t,a_t}\right) &= \mathbb{E}\left(\sum_{t=1}^T (r_{t,a^\dagger} - \tilde{r}^2_{t,a_t})\mathbb{1}_{\{a_t\neq a^\dagger_{t,\star}\}}\right) \\ &\geq \mathbb{E}\left(\sum_{t=1}^T \min\{\gamma\min_{x\in\mathcal{D}}\langle x, \theta_{a^\dagger_{t,\star}}\rangle, \Delta\}\mathbb{1}_{\{a_t\neq a^\dagger_{t,\star}\}}\mathbb{1}_{\{E_\delta\}}\right) - T\delta\end{aligned}$$

Finally, putting everything together we have:

$$\begin{aligned}\mathbb{E}\left(\sum_{t=1}^T \gamma\min_{x\in\mathcal{D}}\langle x, \theta_{a^\dagger_{t,\star}}\rangle\mathbb{1}_{\{a_t\neq a^\dagger_{t,\star}\}}\right) \leq \qquad & \tilde{R}^{2,\exp}_T + \mathbb{E}\left(\sum_{t=1}^T (1 - E^{(t)}_{m^\star, a^\dagger_{t,\star}})\right) \\ + \delta T\left(\min\{\gamma\min_{a^\dagger\in A^\dagger}\min_{x\in\mathcal{D}}\langle x, \theta_{a^\dagger}\rangle, \Delta\} + 1\right) &\end{aligned}$$

Hence, because $\tilde{R}^{1,\exp}_T = o(T)$ and $\mathbb{E}\left(\sum_{t=1}^T (1 - E^{(t)}_{m^\star,a^\dagger})\right) = o(T)$ we have that for $\delta \leq 1/T$, the expected number of pulls of the optimal arm in $A^\dagger$ is of order $o(T)$. In addition, the cost for the attacker is bounded by:

$$\mathbb{E}\left(\sum_{t=1}^T c_t\right) = \mathbb{E}\left(\sum_{t=1}^T \mathbb{1}_{\{a_t\neq a^\dagger_{t,\star}\}}\left|\max(-1, \min(C_{t,a_t}, 0))\right|\right) \leq \mathbb{E}\left(\sum_{t=1}^T \mathbb{1}_{\{a_t\neq a^\dagger_{t,\star}\}}\right)$$

□

The proof is similar to the one of Prop. 24. The condition on the expert in Prop. 26 means that there exists an expert which believes an arm $a^\dagger \in A^\dagger$ is optimal most of the time. The adversarial algorithm will then learn that this expert is optimal. Algorithm Exp4 has a regret $R^{\exp}_T$ bounded by $\sqrt{2TK\log(N)}$, thus the total number of pulls of arms not in $A^\dagger$ is bounded by

$\sqrt{2TK\log(M)}/\gamma$. This result also implies that for adversarial algorithms like Exp3 Auer et al. [2002], the same type of attacks could be used to fool $\mathfrak{A}$ into pulling arms in $A^\dagger$ because the MAB problem can be seen as a reduction of the contextual bandit problem with a unique context and one expert for each arm.

D.12 Appendix: Contextual Bandit Algorithms

In this appendix, we present the different bandit algorithms studied in this paper. All algorithms we consider except Exp4 uses disjoint models for building estimate of the arm feature vectors $(\theta_a)_{a\in[\![1,K]\!]}$. Each algorithm (except Exp4) builds least squares estimates of the arm features.

Algorithm 8: Contextual LinUCB

Input: regularization λ, number of arms K, number of rounds T, bound on context norms: L, bound on norms θ_a: D
Initialize for every arm a, $\bar{V}_a^{-1}(t) = \frac{1}{\lambda}I_d$, $\hat{\theta}_a(t) = 0$ and $b_a(t) = 0$
for $t = 1, ..., T$ **do**
Observe context x_t
Compute $\beta_a(t) = \sigma\sqrt{d\log\left(\frac{1+N_a(t)L^2/\lambda}{\delta}\right)}$ with $N_a(t)$ the number of pulls of arm a
Pull arm $a_t = \text{argmax}_a \langle\hat{\theta}_a(t), x_t\rangle + \beta_a(t)||x_t||_{\bar{V}_a^{-1}(t)}$
Observe reward r_t and update parameters $\hat{\theta}_a(t)$ and $\bar{V}_a^{-1}(t)$ such that:

$$\bar{V}_{a_t}(t+1) = \bar{V}_{a_t}(t) + x_t x_t^\intercal, \quad b_{a_t}(t+1) = b_{a_t}(t) + r_t x_t,$$
$$\theta_{a_t}(t+1) = \bar{V}_{a_t}^{-1}(t+1) b_{a_t}(t+1)$$

end for

Algorithm 9: Linear Thompson Sampling with Gaussian prior

Input: regularization λ, number of arms K, number of rounds T, variance υ
Initialize for every arm a, $\bar{V}_a^{-1}(t) = \lambda I_d$ and $\hat{\theta}_a(t) = 0$, $b_a(t) = 0$
for $t = 1, ..., T$ **do**
Observe context x_t
Draw $\tilde{\theta}_a \sim \mathcal{N}(\hat{\theta}_a(t), \upsilon^2\bar{V}_a^{-1}(t))$
Pull arm $a_t = \text{argmax}_{a\in[\![1,K]\!]}\left\langle\tilde{\theta}_a, x_t\right\rangle$
Observe reward r_t and update parameters $\hat{\theta}_a(t)$ and $\bar{V}_a^{-1}(t)$

$$\bar{V}_{a_t}(t+1) = \bar{V}_{a_t}(t) + x_t x_t^\intercal, \quad b_{a_t}(t+1) = b_{a_t}(t) + r_t x_t,$$
$$\theta_{a_t}(t+1) = \bar{V}_{a_t}^{-1}(t+1) b_{a_t}(t+1)$$

end for

Algorithm 10: ε-GREEDY

Input: regularization λ, number of arms K, number of rounds T, exploration parameter $(\varepsilon)_t$
Initialize, for all arms a, $\bar{V}_a^{-1}(t) = \lambda I_d$ and $\hat{\theta}_a(t) = 0$, $\varepsilon_t = 1$, $b_a(t) = 0$
for $t = 1, \ldots, T$ **do**
Observe context x_t
With probability ε_t, pull $a_t \sim \mathcal{U}([\![1, K]\!])$, or pull $a_t = \text{argmax}\langle \theta_a, x_t \rangle$
Observe reward r_t and update parameters $\hat{\theta}_a(t)$ and $\bar{V}_a^{-1}(t)$

$$\bar{V}_{a_t}(t+1) = \bar{V}_{a_t}(t) + x_t x_t^\intercal, \quad b_{a_t}(t+1) = b_{a_t}(t) + r_t x_t,$$
$$\theta_{a_t}(t+1) = \bar{V}_{a_t}^{-1}(t+1) b_{a_t}(t+1)$$

end for

Algorithm 11: EXP4

Input: number of arms K, experts: $(E_m)_{m \in [\![1,N]\!]}$, parameter η
Set $Q_1 = (1/N)_{j \in [\![1,N]\!]}$
for $t = 1, \ldots, T$ **do**
Observe context x_t and probability recommendation $(E_m^{(t)})_{m \in [\![1,N]\!]}$
Pull arm $a_t \sim P_t$ where $P_{t,j} = \sum_{k=1}^N Q_{t,k} E_{j,k}^{(t)}$
Observe reward r_t and define for all arms i $\hat{r}_{t,i} = 1 - \mathbb{1}_{\{a_t = i\}}(1 - r_t)/P_{t,i}$
Define $\tilde{X}_{t,k} = \sum_a E_{k,a}^{(t)} \hat{r}_{t,a}$
Update $Q_{t+1,j} = \exp(\eta Q_{t,i}) / \sum_{j=1}^N \exp(\eta Q_{t,j})$ for all experts i
end for

D.13 Appendix: Semi-Online Attacks

Liu and Shroff [2019] studies what they call the offline setting for adversarial attacks on stochastic bandits. They consider a setting where a bandit algorithm is successively updated with mini-batches of fixed size B. The attacker can tamper with some of the incoming mini-batches. More precisely, they can modify the context, the reward and even the arm that was pulled for any entry of the attacked mini-batches. The main difference between this type of attacks and the online attacks we considered in the main paper is that we do not assume that we can attack from the start of the learning process: the bandit algorithm may have already converged by the time we attack.

We can still study the cumulative cost for the attacker to change the mini-batch in order to fool a bandit algorithm to pull a target arm $a^\dagger$ (here we take $A^\dagger = \{a^\dagger\}$). Contrarily to Liu and Shroff [2019], we call this setting semi-online. We first study the impact of an attacker on LINUCB where we show that, by modifying only $(K-1)d$ entries from the batch $\mathcal{B}$, the attacker can force LINUCB to pull arm $a^\dagger$, $M'B - o(M'B)$ times with M' the number of remaining batches updates. The cost of our attack is $\sqrt{MB}$ with M the total number of batches.

Cost of an attack: If presented with a mini-batch $\mathcal{B}$, with elements (x_t, a_t, r_t) composed of the context x_t presented at time t, the action taken a_t and the reward received r_t, the attacker modifies element i, namely (x_t^i, a_t^i, r_t^i) into $(\tilde{x}_t^i, \tilde{a}_t^i, \tilde{r}_t^i)$. The cost of doing so is $c_t^i = ||x_t^i - \tilde{x}_t^i||_2 + \left|\tilde{r}_t^i - r_t^i\right| + \mathbb{1}_{\{a_t^i \neq \tilde{a}_t^i\}}$ and the total cost for mini-batch $\mathcal{B}$ is defined as $c_{\mathcal{B}} = \sum_{i \in \mathcal{B}} c_t^i$. Finally, we consider the cumulative cost of the attack over M different mini-batches $\mathcal{B}_1, \dots, \mathcal{B}_M$, $\sum_{l=1}^{M} c_{\mathcal{B}_l}$. The interaction between the environment, the attacker and the learning algorithm is summarized in Alg. 12.

Algorithm 12: Semi-Online Attack Setting.

> **Input:** Bandit alg. $\mathfrak{A}$, size of a mini-batch: B
> Set $t = 0$
> **while** True **do**
> $\mathfrak{A}$ observe context x_t
> $\mathfrak{A}$ pulls arm a_t and observes reward r_t
>   Interaction (x_t, a_t, r_t) is saved in mini-batch $\mathcal{B}$
>   **if** $\left|\mathcal{B}\right| = B$ **then**
>   Attacker modifies mini-batch $\mathcal{B}$ into $\tilde{\mathcal{B}}$
>   Update alg. $\mathfrak{A}$ with poisoned mini-batch $\tilde{\mathcal{B}}$
>   **end if**
> **end while**

The attack presented here is based on the Ahlberg–Nilson–Varah bound Varah [1975], which gives a control on the sup norm of a matrix with dominant diagonal elements. More precisely, when presented with a mini-batch $\mathcal{B}$, the attacker needs to modify the contexts and the rewards. We assume that the attacker knows the number of mini-batch updates M and has access to a lower-bound on the reward of the target arm, ν as in Assumption 3.

The attacker changes $(K-1) \times d$ rows of the first mini-batch to rewards of 0 with a context $\delta_a e_i$ for each arm $a \neq a^\dagger$ with (e_i) the canonical basis of $\mathbb{R}^d$. Moreover, δ_a is chosen such that:

$$\delta_a > \max\left(\sqrt{\frac{2MBL^2d}{\nu} + dMB}, \sqrt{\frac{4\beta_{max}^2 L^2 d}{\nu^2} + dMB}\right) \tag{D.13}$$

with $\beta_{max} = \max_{t=0}^{MB} \beta_a(t)$ and M the number of mini-batch updates.

Proposition 27. *After the first attack, with probability* $1-\delta$*,* LinUCB *always pulls arm* $a^\dagger$*,*

Proof. After having poisoned the first mini-batch $\mathcal{B}$, the latter can be partitioned into two subsets, $\mathcal{B}_c$ (with non-perturbed rows) and $\mathcal{B}_{nc}$ (with the poisoned rows). The design matrix of arm $a \neq a^\dagger$ for every time t after the poisoning is:

$$V_{t,a} = \lambda I_d + \sum_{l=1, a_l = a}^{t} x_l x_l^\intercal + \delta_a^2 \sum_{i=1}^{d} e_i e_i^\intercal \tag{D.14}$$

For every time t, non diagonal elements of $V_{t,a} = (v_{i,j})_{i,j}$ are bounded by:

$$\forall i, r_i := \sum_{j \neq i} v_{i,j} \leq \sum_{j \neq i} \sum_{l=1, a_l=a}^{t} \|x_l x_l^\intercal\|_\infty \leq dN_a(kB) \tag{D.15}$$

Whereas for all diagonal elements, $v_{i,i} \geq \delta_a^2 > r_i$. Thus $V_{t,a}$ is strictly diagonal dominant and by the Ahlberg–Nilson–Varah bound Varah [1975]:

$$\|V_{t,a}^{-1}\|_\infty \leq \frac{1}{\min_i(\|v_{i,i}\| - r_i)} \leq \frac{1}{\delta_a^2 - dMB} \tag{D.16}$$

Then, for every arm $a \neq a^\dagger$ and any context $x \in \mathcal{D}$ and any time t after the attack:

$$\begin{aligned} \langle \hat{\theta}_{a,t}, x \rangle + \beta_a(t)||x||_{V_{t,a}^{-1}} &\leq \sum_{l=1, a_l=a}^{t} r_t (V_{t,a}^{-1} x_t)^\intercal x + \beta_a(t)||x||_1 \sqrt{\|V_{t,a}^{-1}\|_\infty} \\ &\leq \|V_{t,a}^{-1}\|_\infty dN_t(a) \sup_{y \in \mathcal{D}} ||y||_2^2 \\ &+ \beta_{\max} \sqrt{d} \sup_{y \in \mathcal{D}} ||y||_2 \sqrt{\|V_{t,a}^{-1}\|_\infty} < \nu \end{aligned}$$

We have shown that for any arm $a \neq a^\dagger$ and any time step t after the attack, the upper confidence bound computed by LinUCB is upper-bounded bu ν the arm $a^\dagger$. Then, with probability $1 - \delta$, the confidence set for arm $a^\dagger$ holds and, for all $x \in \mathcal{D}$, arm $a^\dagger$ is chosen by LinUCB. The total cost of this attack is $d \sum_{a \neq a^\dagger} \delta_a L = O(\sqrt{MB})$ □

E ROPUST: Improving Robustness through Fine-tuning with Photonic Processors and Synthetic Gradients

Robustness to adversarial attacks is typically obtained through expensive adversarial training with Projected Gradient Descent. Here we introduce ROPUST, a remarkably simple and efficient method to leverage robust pre-trained models and further increase their robustness, at no cost in natural accuracy. Our technique relies on the use of an Optical Processing Unit (OPU), a photonic co-processor, and a fine-tuning step performed with Direct Feedback Alignment, a synthetic gradient training scheme. We test our method on nine different models against four attacks in RobustBench, consistently improving over state-of-the-art performance. We perform an ablation study on the single components of our defense, showing that robustness arises from parameter obfuscation and the alternative training method. We also introduce phase retrieval attacks, specifically designed to increase the threat level of attackers against our own defense. We show that even with state-of-the-art phase retrieval techniques, ROPUST remains an effective defense.

E.1 Introduction

Adversarial examples Goodfellow et al. [2015a] threaten the safety and reliability of machine learning models deployed in the wild. Because of the sheer number of attack and defense scenarios, true real-world robustness can be difficult to evaluate Bubeck et al. [2019]. Standardized benchmarks, such as RobustBench Croce et al. [2020a] using AutoAttack Croce and Hein [2020b], have helped better evaluate progress in the field. Furthermore, the development of defense-specific attacks is also crucial Tramèr and Boneh [2019]. To date, one of the most effective defense techniques remains adversarial training with Projected Gradient Descent (PGD) Madry et al. [2018a]. Adversarial training of a model can be resource-consuming, but robust networks pre-trained with PGD are now widely available.

This motivates the use of these pre-trained robust models as a solid foundation for developing simple and widely applicable defenses that further enhance their robustness. To this end, we introduce **ROPUST**, a drop-in replacement for the classifier of already robust models. Our defense is unique in that it leverages a photonic co-processor (the Optical Processing Unit, OPU) for physical *parameter obfuscation* Cappelli et al. [2021a]: because the *fixed* random parameters are optically implemented, they remain unknown at training and inference time. Additionally, a synthetic gradient method, Direct Feedback Alignment (DFA) Nøkland [2016], is used for fine-tuning the ROPUST classifier.

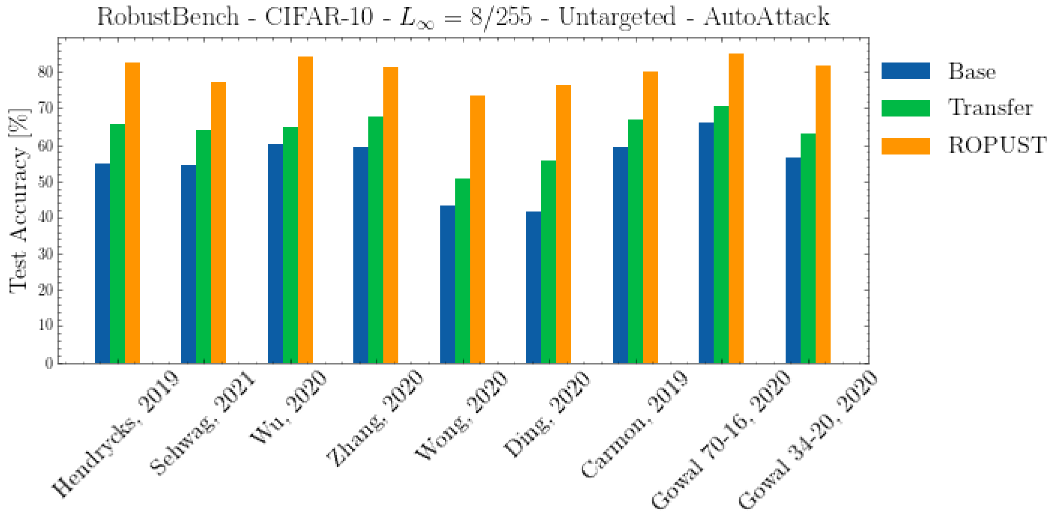


Figure E.1: **ROPUST systematically improves the test accuracy of already robust models**. Transfer refers to the performance when attacks are generated on the base model and transferred to the ROPUST model. Models from the RobustBench model zoo: Hendrycks et al., 2019 Hendrycks et al. [2019], Sehwag et al., 2021 Sehwag et al. [2021], Wu et al., 2020 Wu et al. [2020], Zhang et al., 2020 Zhang et al. [2020], Wong et al., 2020 Wong et al. [2020], Ding et al., 2020 Ding et al. [2020], Carmon et al., 2019 Carmon et al. [2019a], Gowal et al., 2020 Gowal et al. [2020b].

We evaluate extensively our method against AutoAttack on nine different models in RobustBench, and consistently improve robust accuracies over the state-of-the-art (Section E.3 and Fig. E.1). We perform an ablation study, in Section E.4, and find that the robustness of our defense against white-box attacks comes from both *parameter obfuscation* and DFA. Surprisingly, we also discover that simply retraining the classifier of a robust model on natural data increases its robustness to square attacks, a phenomenon that warrants further study. Finally, in Section E.5, we develop a *phase retrieval* attack targeting specifically the parameter obfuscation of our defense, and show that even against state-of-the art phase retrieval techniques, ROPUST achieves fair robustness.

E.1.1 Related work

Attacks. Adversarial attacks have been framed in a variety of settings: white-box, where the attacker is assumed to have unlimited access to the model, including its parameters (e.g. FGSM Goodfellow et al. [2015a], PGD Madry et al. [2018a], Kurakin et al. [2016], Carlini & Wagner Carlini and Wagner [2017]); black-box, assuming only limited access to the network for the attacker, such as the label or logits for a given input, with methods attempting to estimate the gradients Chen et al. [2017], Ilyas et al. [2018a,b], or more recently derived from genetic algorithms Andriushchenko et al. [2019], Meunier et al. [2019] and combinatorial optimization Moon et al. [2019]; transfer attacks, where an attack is crafted on a similar model that is accessible to the attacker, and then applied to the target network Papernot et al. [2016a]. Automated schemes, such as AutoAttack Croce and Hein [2020b], have been proposed to autonomously select which attack to perform against a given network, and to automatically tune its hyperparameters.

Defenses. Adversarial training adds adversarial robustness as an explicit training objective Goodfellow et al. [2015a], Madry et al. [2018a], by incorporating adversarial examples during the training. This has been, and still is, one of the most effective defense against attacks. Repository of pre-trained robust models have been compiled, such as the RobustBench Model Zoo[1]. Conversely, theoretically grounded defenses have been proposed [Lecuyer et al., 2018, Cohen et al., 2019, Alexandre Araujo and Negrevergne, 2020, Pinot et al., 2019, Wong et al., 2018, Wong and Kolter, 2018], but these fail to match the clean accuracy of state-of-the-art networks, making robustness a trade-off with performance. Many empirical defenses have been criticized for providing a false sense of security Athalye et al. [2018a], Tramèr and Boneh [2019], by not evaluating on attacks adapted to the defense.

Obfuscation. Gradient obfuscation, through the use of a non-differentiable activation function, has been proposed as a way to protect against white-box attacks Papernot et al. [2017b]. However, gradient obfuscation can be easily bypassed by Backward Pass Differentiable Approximation (BPDA) Athalye et al. [2018a], where the defense is replaced by an approximated and differentiable version. *Parameter obfuscation* has been proposed with dedicated photonic co-processor Cappelli et al. [2021a], enforced by the physical properties of said co-processor. However, by itself, this kind of defense falls short of adversarial training.

Fine-tuning and analog computing. Previous work introduced *adversarial fine-tuning* Jeddi et al. [2020]: fine-tuning a non-robust model with an adversarial objective. In this work instead we fine-tune a robust model without adversarial training. Additionally, it was shown that robustness improves transfer performance Salman et al. [2020] and that robustness transfers across datasets Shafahi et al. [2020]. The advantage of non-ideal analog computations in terms of robustness has been investigated in the context of NVM crossbars Roy et al. [2020], while we here focus on a photonic technology, readily available to perform computations at scale.

E.1.2 Motivations and contributions

We propose to simplify and extend the applicability of photonic-based parameter obfuscation defenses. Our defense, ROPUST, is a universally and easily applicable drop-in replacement for classifiers of already robust models. In contrast with existing parameter-obfuscation methods, it leverages pre-trained robust models, and achieves state-of-the-art performance.

Beyond silicon and beyond backpropagation. We leverage photonic hardware and alternative training methods to achieve adversarial robustness. The use of dedicated hardware to perform the random projection physically guarantees *parameter obfuscation*. Direct Feedback Alignment enables us to train and/or fine-tune the model despite non-differentiable analog hardware being used in the forward pass. In our ablation study, we find that both these components contribute to adversarial robustness, providing a holistic defense.

[1] Accessible at: `https://github.com/RobustBench/robustbench`.

Simple, universal, and state-of-the-art. ROPUST can be dropped-in to supplement any robust pre-trained model, replacing its classifier. Fine-tuning the ROPUST classifier is fast and does not require additional changes to the model architecture. This enables any existing architecture and adversarial countermeasure to leverage ROPUST to gain additional robustness, at limited cost. We evaluate on RobustBench, across 9 pre-trained models, against AutoAttack sampling from a pool of 4 attacks. We achieve state-of-the-art performance on the leaderboard, and, in light of our results, we suggest the extension of RobustBench to include obfuscation-based methods.

The Square attack mystery. Performing an ablation study on Square attack Andriushchenko et al. [2019], we find that simply retraining from scratch the classifier of a robust model on natural data increases its robustness against it. This phenomenon remains unexplained and occurs even when the original fully connected classification layer is retrained, without using our ROPUST module.

Phase retrieval attacks. Drawing inspiration from the field of phase retrieval, we introduce a new kind of attack against defenses relying on parameter obfuscation, *phase retrieval attacks*. These attacks assume the attacker leverage phase retrieval techniques to retrieve the obfuscated parameters in full, and we show that ROPUST remains robust even against state-of-the-art retrieval methods.

E.2 Methods

E.2.1 Automated adversarial attacks

We evaluate our model against the four attacks implemented in RobustBench: APGD-CE and APGD-T Croce and Hein [2020b], Square attack Andriushchenko et al. [2019], and Fast Adaptive Boundary (FAB) attack Croce and Hein [2020a]. APGD-CE is a standard PGD where the step size is tuned using the loss trend information, squeezing the best performance out of a limited iterations budget. APGD-T, on top of the step size schedule, substitutes the cross-entropy loss with the Difference of Logits Ratio (DLR) loss, reducing the risk of vanishing gradients. Square attack is based on a random search. Random updates δ are sampled from an attack-norm dependent distribution at each iteration: if they improve the objective function they are kept, otherwise they are discarded. FAB attack aims at finding adversarial samples with minimal distortion with respect to the attack point. With respect to PGD, it does not need to be restarted and it achieves fast good quality results. In RobustBench, using AutoAttack, given a batch of samples, these are first attacked with APGD-CE. Then, the samples that were successfully attacked are discarded, and the remaining ones are attacked with APGD-T. This procedure continues with Square and FAB attack.

E.2.2 Our defense

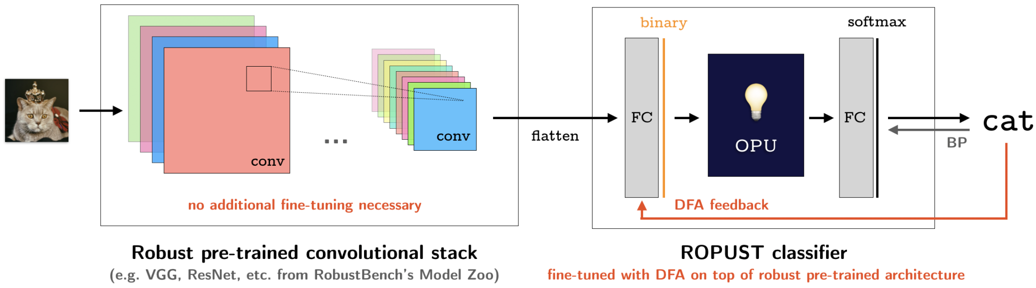


Figure E.2: **ROPUST replaces the classifier of already robust models, enhancing their adversarial robustness.** Only the ROPUST classifier needs fine-tuning; the convolutional stack is frozen. Convolutional features first go through a fully-connected layer, before binarization for use in the Optical Processing Unit (OPU). The OPU performs a non-linear random projection, with *fixed unknown parameters*. A fully-connected layer is then used to obtain a prediction from the output of the OPU. Direct Feedback Alignment is used to train the layer underneath the OPU.

Optical Processing Units. Optical Processing Units (OPU)[2] are photonic co-processors dedicated to efficient large-scale random projections. Assuming an input vector $\mathbf{x}$, the OPU computes the following operation using light scattering through a diffusive medium:

$$\mathbf{y} = |\mathbf{U}\mathbf{x}|^2 \tag{E.1}$$

With $\mathbf{U}$ a *fixed* complex Gaussian random matrix of size up to $10^6 \times 10^6$, which entries are not readily known. In the following, we sometimes refer to $\mathbf{U}$ as the *transmission matrix* (TM). The input $\mathbf{x}$ is binary (1 bit – 0/1) and the output $\mathbf{y}$ is quantized in 8-bit. While it is possible to simulate an OPU and implement ROPUST on GPU, this comes with two significant drawbacks: (1) part of our defense relies on $\mathbf{U}$ being obfuscated to the attacker, which is not possible to guarantee on a GPU; (2) at large sizes, storing $\mathbf{U}$ in GPU memory is costly Ohana et al. [2020].

Because $\mathbf{U}$ is physically implemented through the diffusive medium, the random matrix will remain unknown even if the host system is compromised. Assuming unfettered access to the OPU, an attacker has to perform *phase retrieval* to retrieve the coefficients of $\mathbf{U}$. As only the non-linear intensity $|\mathbf{U}\mathbf{x}|^2$ can be measured and not $\mathbf{U}\mathbf{x}$ directly, this phase retrieval step is computationally costly. This problem is well studied, and state-of-the-art methods have $O(MN \log N)$ time complexity Gupta et al. [2020], and do not result in a perfect retrieval. We develop an attack scenario based on this method in Section E.5.

Direct Feedback Alignment. Because the fixed random parameters implemented by the OPU are unknown, it is impossible to backpropagate through it. We bypass this limitation by training layers upstream of the OPU using Direct Feedback Alignment (DFA) Nøkland [2016]. DFA is an alternative to backpropagation, capable of scaling to modern deep learning tasks and architectures Launay et al. [2020], relying on a random projection of the error as the teaching signal.

[2] Accessible through LightOn Cloud: `https://cloud.lighton.ai`.

In a fully connected network, at layer i out of N, neglecting biases, with $\mathbf{W}_i$ its weight matrix, f_i its non-linearity, and $\mathbf{h}_i$ its activations, the forward pass can be written as $\mathbf{a}_i = \mathbf{W}_i\mathbf{h}_{i-1}, \mathbf{h}_i = f_i(\mathbf{a}_i)$. $\mathbf{h}_0 = X$ is the input data, and $\mathbf{h}_N = f(\mathbf{a}_N) = \hat{\mathbf{y}}$ are the predictions. A task-specific cost function $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$ is computed to quantify the quality of the predictions with respect to the targets $\mathbf{y}$. The weight updates are obtained through the chain-rule of derivatives:

$$\delta\mathbf{W}_i = -\frac{\partial\mathcal{L}}{\partial\mathbf{W}_i} = -[(\mathbf{W}_{i+1}^T\delta\mathbf{a}_{i+1}) \odot f_i'(\mathbf{a}_i)]\mathbf{h}_{i-1}^T, \delta\mathbf{a}_i = \frac{\partial\mathcal{L}}{\partial\mathbf{a}_i} \tag{E.2}$$

where $\odot$ is the Hadamard product. With DFA, the gradient signal $\mathbf{W}_{i+1}^T\delta\mathbf{a}_{i+1}$ of the (i+1)-th layer is replaced with a random projection of the gradient of the loss at the top layer $\delta\mathbf{a}_y$–which is the error $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$ for commonly used losses, such as cross-entropy or mean squared error:

$$\delta\mathbf{W}_i = -[(\mathbf{B}_i\delta\mathbf{a}_y) \odot f_i'(\mathbf{a}_i)]\mathbf{h}_{i-1}^T, \delta\mathbf{a}_y = \frac{\partial\mathcal{L}}{\partial\mathbf{a}_y} \tag{E.3}$$

Learning with DFA is enabled by an alignment process, wherein the forward weights learn a configuration enabling DFA to approximate BP updates Refinetti et al. [2020].

ROPUST To enhance the adversarial robustness of pretrained robust models, we propose to replace their classifier with the ROPUST module (Fig. E.2). We use robust models from the RobustBench model zoo, extracting and freezing their convolutional stack. The robust convolutional features go through a fully connected layer and a binarization step (a sign function), preparing them for the OPU. The OPU then performs a non-linear random projection, with fixed unknown parameters. Lastly, the predictions are obtained through a final fully-connected layer. While the convolutional layers are frozen, we train the ROPUST module on natural data using DFA to bypass the non-differentiable photonic hardware.

Attacking ROPUST. While we could use DFA to attack ROPUST, previous work has shown that methods devoid of weight transport are not effective in generating compelling adversarial examples Akrout [2019]. Therefore, we instead use backward pass differentiable approximation (BPDA) when attacking our defense. For BPDA, we need to find a good differentiable relaxation to non-differentiable layers. For the binarization function, we simply use the derivative of $\tanh$ in the backward pass, while we approximate the transpose of the obfuscated parameters with a different fixed random matrix drawn at initialization of the module. More specifically, if we consider the expression for the forward pass of the ROPUST module:

$$\mathbf{y} = \text{softmax}(\mathbf{W}_3|\mathbf{U}\text{sign}(\mathbf{W}_1\mathbf{x})|^2) \tag{E.4}$$

In the backward we substitute $\mathbf{U}^T$ (that we do not have access to) with a different fixed random matrix $\mathbf{R}$, in a setup similar to Feedback Alignment Lillicrap et al. [2014]. We also relax the sign function derivative to the derivative of $\tanh$.

We present empirical results on RobustBench in the following Section E.3. We then ablate the components of our defense in E.4, demonstrating its holistic nature, and we finally create a phase retrieval attack to challenge parameter obfuscation in Section E.5.

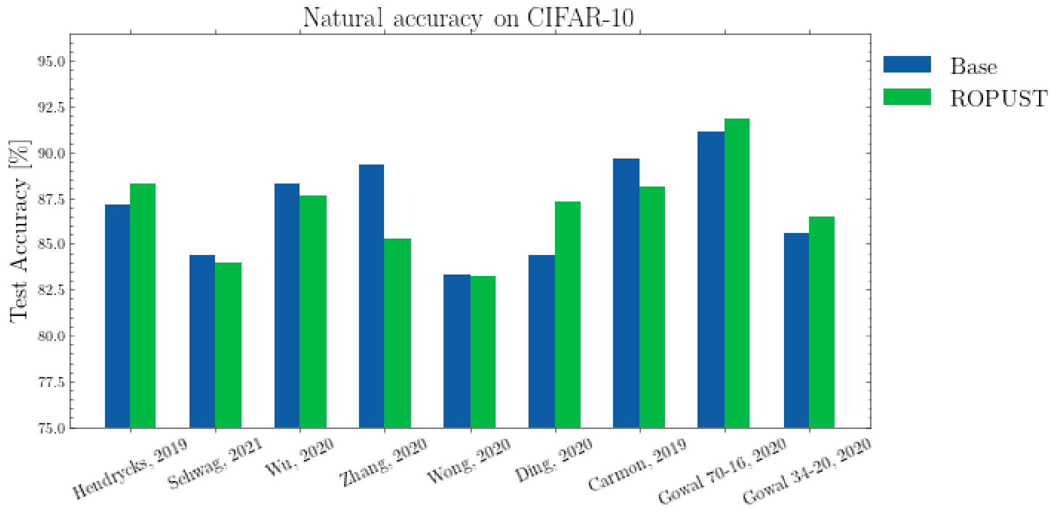


Figure E.3: **Our ROPUST defense comes at no cost in natural accuracy.** In some cases, natural accuracy is even improved. The model from Zhang, 2020 Zhang et al. [2020] is an isolated exception. Models from the RobustBench model zoo: Hendrycks et al., 2019 Hendrycks et al. [2019], Sehwag et al., 2021 Sehwag et al. [2021], Wu et al., 2020 Wu et al. [2020], Zhang et al., 2020 Zhang et al. [2020], Wong et al., 2020 Wong et al. [2020], Ding et al., 2020 Ding et al. [2020], Carmon et al., 2019 Carmon et al. [2019a], Gowal et al., 2020 Gowal et al. [2020b].

E.3 Evaluating ROPUST on RobustBench

All of the attacks are performed on CIFAR-10 Krizhevsky [2009], using a differentiable backward pass approximation Athalye et al. [2018a] as explained in Section E.2.2. For our experiments, we use OPU input size 512 and output size 8000. We use the Adam optimizer Kingma and Ba [2014], with learning rate 0.001, for 10 epochs. The process typically takes as little as 10 minutes on a single NVIDIA V100 GPU.

We show our results on nine different models in RobustBench in Fig. E.1. The performance of the original pretrained models from the RobustBench leaderboard is reported as *Base*. *ROPUST* represents the same models equipped with our defense. Finally, *Transfer* indicates the performance of attacks created on the original model and transferred to fool the ROPUST defense. For all models considered, ROPUST improves the robustness significantly, even under transfer.

For transfer, we also tested crafting the attacks on the *Base* model while using the loss of the ROPUST model for the learning rate schedule of APGD. We also tried to use the predictions of ROPUST, instead of the base model, to *remove* the samples that were successfully attacked from the next stage of the ensemble; however, these modifications did not improve transfer performance.

Finally, we remark that the robustness increase typically comes at no cost in natural accuracy; we show the accuracy on natural data of the *Base* and the *ROPUST* models in Fig. E.3.

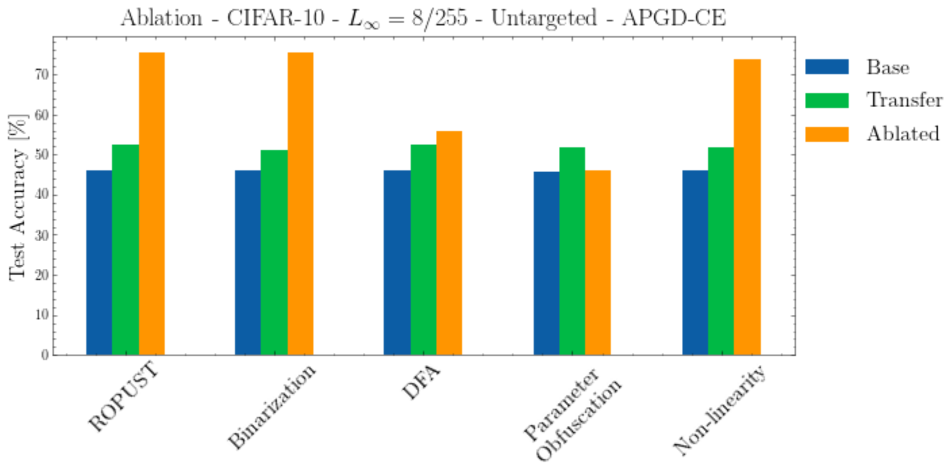


Figure E.4: **Removing either parameter obfuscation or DFA from our defense causes a large drop in accuracy.** This confirms the intuition that robustness is given by the inability to efficiently generate attacks in a white-box settings when the parameters are obfuscated, and that DFA is capable of generating partially robust features. We note that even though the non-linearity $|.|^2$ does not contribute to robustness, it is key to obfuscation, preventing trivial retrieval. Transfer performance does not change much when removing components of the defense. While the `Base` model is not ablated, we leave its performance as a term of comparison.

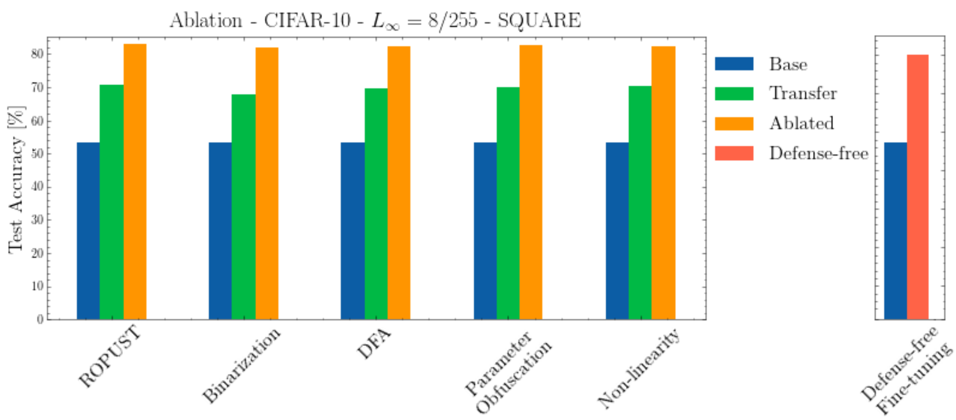


Figure E.5: **Square attack can be evaded by simply retraining on natural data the classifier of a robust model.** We confirm the same result when retraining the standard fully connected classification layer in the pretrained models in place of the ROPUST module (*Defense-free* result in the chart on the right). While the `Base` model is not ablated, we leave its performance as a term of comparison.

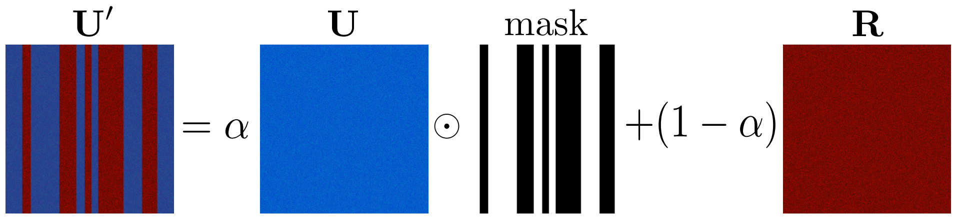


Figure E.6: **Simplified modelling of phase retrieval.** The retrieved matrix $\mathbf{U}'$ is modeled as the linear interpolation between the real transmission matrix $\mathbf{U}$ and a random matrix $\mathbf{R}$, only for some columns selected by a mask. Varying the value of α and the percentage of masked columns allows to modulate the knowledge of the attacker without running resource-hungry phase retrieval algorithms.

E.4 Understanding ROPUST: an ablation study

We use the model from Wong et al. [2020] available in the RobustBench model zoo to perform our ablation studies. It consists in a PreAct ResNet-18 He et al. [2016a], pretrained with a *"revisited"* FGSM of increased effectiveness.

Holistic defense. We conduct an ablation study by removing a single component of our defense at a time in simulation: binarization, DFA, parameter obfuscation, and non-linearity $|.|^2$ of the random projection. To remove DFA, we also remove the binarization step and train the ROPUST module with backpropagation, since we have access to the transpose of the transmission matrix in the simulated setting of the ablation study. We show the results in Fig. E.4: we see that removing the non-linearity $|.|^2$ and the binarization does not have an effect, with the robustness given by *parameter obfuscation* and DFA, as expected. However, note that $|.|^2$ is central to preventing trivial phase retrieval, and is hence a key component of our defense.

Robustness to Square attack While the ablation study on the APGD attack is able to pinpoint the exact sources of robustness for a white-box attack, the same study on the black-box Square attack has surprising results. Indeed, as shown in Fig. E.5, no element of the ROPUST mechanism can be linked to robustness against Square attack. Interestingly, we found an identical behaviour when retraining the standard fully connected classification layer from scratch on natural (non perturbed) data, shown in the same Fig. E.5 under the *Defense-free* label.

E.5 Phase retrieval attack

Our defense leverages parameter obfuscation to achieve robustness. Yet, however demanding, it is still technically possible to recover the parameters through phase retrieval schemes Gupta et al. [2019b, 2020]. To provide a thorough and fair evaluation of our attack, we study in this section *phase retrieval* attacks. We first consider an idealized setting, and then confront this setting with a real-world phase retrieval algorithm from Gupta et al. [2020].

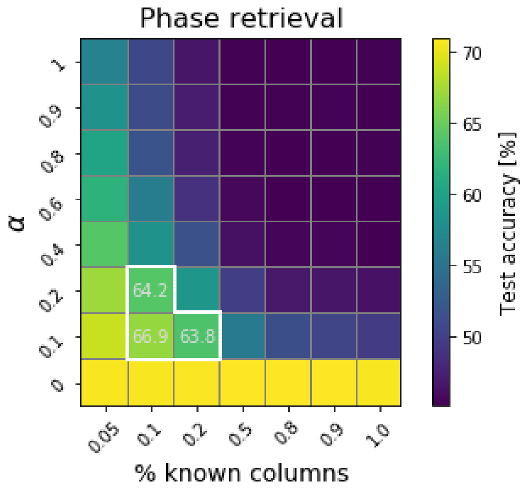


Figure E.7: **Performance of an APGD-CE attack with a retrieved matrix in place of the, otherwise unknown, transpose of the transmission matrix.** As expected, a better knowledge of the transmission matrix, i.e. higher alpha and/or higher percentage of known columns correlates with the success of the attack, with a sharp phase transition. At first glance, it may seem that even a coarse-grained knowledge of the TM can help the attacker. However, optical phase retrieval works on the output correlation only: accordingly, we find that even state-of-the-art phase retrieval methods operates only in the white contoured region, where the robustness is still greater than the *Base* models. We highlighted the accuracies achieved under attack in this region in the heat-map.

Ideal retrieval model. We build an idealized phase retrieval attack, where the attacker knows a certain fraction of columns, up to a certain precision, schematized in Figure E.6. To smoothly vary the precision, we model the retrieved matrix $\mathbf{U}'$ as a linear interpolation of the real transmission matrix $\mathbf{U}$ and a completely different random matrix $\mathbf{R}$:

$$\mathbf{U}' = \alpha\mathbf{U} + (1-\alpha)\mathbf{R} \tag{E.5}$$

In real phase retrieval, this model is valid for a certain fraction of columns of the transmission matrix, and the remaining ones are modeled as independent random vectors. We can model this with a Boolean mask matrix $\mathbf{M}$, so our retrieval model in the end is:

$$\mathbf{U}' = \alpha\mathbf{U} \odot \mathbf{M} + (1-\alpha)\mathbf{R} \tag{E.6}$$

In this setting, we vary the knowledge of the attacker from the minimum to the maximum by varying α and the percentage of retrieved columns, and we show how the performance of our defense changes in Fig. E.7. In this simplified model only a crude knowledge of the parameters seems sufficient, given the sharp phase transition. We now need to chart where state-of-the-art retrieval methods are on this graph to estimate their ability to break our defense.

Real-world retrieval performance. State-of-the-art phase retrieval methods seek to maximize output correlation, i.e. the correlation on $\mathbf{y}$ in Eq. E.1, in place of the correlation with respect to the parameters of the transmission matrix, i.e. $\mathbf{U}$ in Eq. E.1. This leads to a retrieved matrix that may well approximate the OPU outputs, but not the actual transmission matrix it implements. We find this is a significant limitation for attackers. In Fig. E.7, following numerical experiments, we highlight with a white contour the operating region of a state-of-the-art phase retrieval algorithm Gupta et al. [2020], showing that it can manage to only partially reduce the robustness of ROPUST.

E.6 Conclusion

We introduced ROPUST, a drop-in module to enhance the adversarial robustness of pretrained already robust models. Our technique relies on parameter obfuscation guaranteed by a photonic co-processor, and a synthetic gradient method: it is simple, fast and widely applicable.

We thoroughly evaluated our defense on nine different models in the standardized RobustBench benchmark, reaching state-of-the-art performance. In light of these results, we encourage to extend RobustBench to include parameter obfuscation methods.

We performed an ablation study in the white-box setting, confirming our intuition and the results from Cappelli et al. [2021a]: the robustness comes from the parameter obfuscation and from the hybrid synthetic gradient method. The non-linearity $|.|^2$ on the random projection, while not contributing to robustness on its own, is key to prevent trivial deobfuscation by hardening ROPUST against phase retrieval. A similar study in the black-box setting was inconclusive. However it shed light on a phenomenon of increased robustness against Square attack when retraining from scratch the classifier of robust architectures on natural data. This phenomenon appears to be universal, i.e. independent of the structure of the classification module being fine-tuned, warranting further study.

Finally, we developed a new kind of attacks, *phase retrieval attacks*, specifically suited to parameter obfuscation defense such as ours, and we tested their effectiveness. We found that the typical precision regime of even state-of-the-art phase retrieval methods is not enough to completely break ROPUST.

Future work could investigate how the robustness varies with the input and output size of the ROPUST module, and if there are different parameter obfuscation trade-offs when such dimensions change. The combination of ROPUST with other defense techniques, such as adversarial label-smoothing Goibert and Dohmatob [2019], could also be of interest to further increase robustness. By combining beyond silicon hardware and beyond backpropagation training methods, our work highlights the importance of considering solutions outside of the hardware lottery Hooker [2020].

Broader impact. Adversarial attacks have been identified as a significant threat to applications of machine learning in-the-wild. Developing simple and accessible ways to make neural networks more robust is key to mitigating some of the risks and making machine learning applications safer. In particular, more robust models would enable a wider range of business applications, especially in safety-critical sectors.

We do not foresee negative societal impacts from our work, beyond the risk of our defense being broken by future developments of research in adversarial attacks.

A limit of our work is that we prove increased robustness only empirically and not theoretically. However, we note that theoretically grounded defense methods typically fall short of other techniques more used in practice. We also rely on photonic hardware, that is however accessible by anyone similarly to GPUs or TPUs on commercial cloud providers.

We performed all of our experiments on single-GPU nodes with NVIDIA V100, and an OPU, on a cloud provider. We estimate a total of ~ 500 GPU hours was spent.

F Equitable and Optimal Transport with Multiple Agents

We introduce an extension of the Optimal Transport problem when multiple costs are involved. Considering each cost as an agent, we aim to share equally between agents the work of transporting one distribution to another. To do so, we minimize the transportation cost of the agent who works the most. Another point of view is when the goal is to partition equitably goods between agents according to their heterogeneous preferences. Here we aim to maximize the utility of the least advantaged agent. This is a fair division problem. Like Optimal Transport, the problem can be cast as a linear optimization problem. When there is only one agent, we recover the Optimal Transport problem. When two agents are considered, we are able to recover Integral Probability Metrics defined by α-Hölder functions, which include the widely-known Dudley metric. To the best of our knowledge, this is the first time a link is given between the Dudley metric and Optimal Transport. We provide an entropic regularization of that problem which leads to an alternative algorithm faster than the standard linear program.

F.1 Introduction

Optimal Transport (OT) has gained interest last years in machine learning with diverse applications in neuroimaging [Janati et al., 2020], generative models [Arjovsky et al., 2017, Salimans et al., 2018], supervised learning [Courty et al., 2016], word embeddings [Alvarez-Melis et al., 2018], reconstruction cell trajectories [Yang et al., 2020b, Schiebinger et al., 2019] or adversarial examples [Wong et al., 2019]. The key to use OT in these applications lies in the gain of computation efficiency thanks to regularizations that smoothes the OT problem. More specifically, when one uses an entropic penalty, one recovers the so called Sinkhorn distances [Cuturi, 2013]. In this paper, we introduce a new family of variational problems extending the optimal transport problem when multiple costs are involved with various applications in fair division of goods/work and operations research problems.

Fair division [Steinhaus, 1949] has been widely studied by the artificial intelligence [Lattimore et al., 2015] and economics [Moulin, 2004] communities. Fair division consists in partitioning diverse resources among agents according to some fairness criteria. One of the standard problems in fair division is the fair cake-cutting problem [Dubins and Spanier, 1961, Brandt et al., 2016]. The cake is an heterogeneous resource, such as a cake with different toppings, and the agents have heterogeneous preferences over different parts of the cake, i.e., some people prefer the chocolate toppings, some prefer the cherries, others just want a piece as large as possible. Hence, taking into account these preferences, one might share the cake equitably between the agents. A generalization of this problem, for which achieving fairness constraints is more challenging, is when the splitting involves several heterogeneous cakes, and where the agents have linked preferences over the different parts of the cakes. This problem has many variants such as the cake-cutting with two cakes [Cloutier et al., 2010], or the Multi Type Resource Allocation [Mackin and Xia, 2015, Wang et al., 2019a]. In all these models it is assumed that there is only one indivisible unit per type of resource available in each cake, and once an agent choose it, he or she has to take it all. In this setting, the cake can be seen as a set where each element of the set represents a type of resource, for instance each element of the cake represents a topping. A natural relaxation of these problems is when a divisible quantity of each type of resources is available. We introduce EOT (**E**quitable and **O**ptimal **T**ransport), a formulation that solves both the cake-cutting and the cake-cutting with two cakes problems in this setting.

Our problem expresses as an optimal transportation problem. Hence, we prove duality results and provide fast computation based on Sinkhorn algorithm. As interesting properties, some Integral Probability Metrics (IPMs) [Müller, 1997] as Dudley metric [Dudley et al., 1966], or standard Wasserstein metric [Villani, 2003] are particular cases of the EOT problem.

Contributions. In this paper we introduce EOT an extension of Optimal Transport which aims at finding an equitable and optimal transportation strategy between multiple agents. We make the following contributions:

- In Section F.3, we introduce the problem and show that it solves a fair division problem where heterogeneous resources have to be shared among multiple agents. We derive its dual and prove strong duality results. As a by-product, we show that EOT is related to some usual IPMs families and in particular the widely known Dudley metric.

- In Section F.4, we propose an entropic regularized version of the problem, derive its dual formulation, obtain strong duality. We then provide an efficient algorithm to compute EOT. Finally we propose other applications of EOT for Operations Research problems.

F.2 Related Work

Optimal Transport. Optimal transport aims to move a distribution towards another at lowest cost. More formally, if c is a cost function on the ground space $\mathcal{X} \times \mathcal{Y}$, then the relaxed Kantorovich formulation of OT is defined for μ and ν two distributions as

$$\mathrm{W}_c(\mu,\nu) := \inf_{\gamma} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)d\gamma(x,y)$$

where the infimum is taken over all distributions γ with marginals μ and ν. Kantorovich theorem states the following strong duality result under mild assumptions [Villani, 2003]

$$\mathrm{W}_c(\mu,\nu) = \sup_{f,g} \int_{\mathcal{X}} f(x)d\mu(x) + \int_{\mathcal{Y}} g(y)d\nu(y)$$

where the supremum is taken over continuous bounded functions satisfying for all x, y, $f(x) + g(y) \leq c(x,y)$. The question of considering an optimal transport problem when multiple costs are involved has already been raised in recent works. For instance, [Paty and Cuturi, 2019] proposed a robust Wasserstein distance where the distributions are projected on a k-dimensional subspace that maximizes their transport cost. In that sense, they aim to choose the most expensive cost among Mahalanobis square distances with kernels of rank k. In articles [Li et al., 2019c, Sun et al., 2020a], the authors aim to learn a cost given observed matchings by inversing the optimal transport problem [Dupuy et al., 2016]. In [Petrovich et al., 2020] the authors study "feature-robust" optimal transport, which can be also seen as a robust cost selection for optimal transport. In articles [Genevay et al., 2017, Scetbon and Cuturi, 2020], the authors learn an adversarial cost to train a generative adversarial network. Here, we do not aim to consider a worst case scenario among the available costs but rather consider that the costs work together in order to split equitably the transportation problem among them at lowest cost.

Entropic relaxation of OT. Computing exactly the optimal transport cost requires solving a linear program with a supercubic complexity $(n^3 \log n)$ [Tarjan, 1997] that results in an output that is *not* differentiable with respect to the measures' locations or weights [Bertsimas and Tsitsiklis, 1997]. Moreover, OT suffers from the curse of dimensionality [Dudley, 1969, Fournier and Guillin, 2015] and is therefore likely to be meaningless when used on samples from high-dimensional densities. Following the line of work introduced by Cuturi [2013], we propose an approximated computation of our problem by regularizing it with an entropic term. Such regularization in OT accelerates the computation, makes the problem differentiable with regards to the distributions [Feydy et al., 2018] and reduces the curse of dimensionality [Genevay et al., 2018]. Taking the dual of the approximation, we obtain a smooth and convex optimization problem under a simplicial constraint.

Fair Division. Fair division of goods has a long standing history in economics and computational choice. A classical problem is the fair cake-cutting that consists in splitting the cake between N individuals according to their heterogeneous preferences. The cake $\mathcal{X}$, viewed as a set, is divided in $\mathcal{X}_1, \dots, \mathcal{X}_N$ disjoint sets among the N individuals. The utility for a single individual i for a slice S is denoted $V_i(S)$. It is often assumed that $V_i(\mathcal{X}) = 1$ and that V_i is additive for disjoint sets. There exists many criteria to assess fairness for a partition $\mathcal{X}_1, \dots, \mathcal{X}_N$ such as proportionality ($V_i(\mathcal{X}_i) \geq 1/N$), envy-freeness ($V_i(\mathcal{X}_i) \geq V_i(\mathcal{X}_j)$) or equitability ($V_i(\mathcal{X}_i) = V_j(\mathcal{X}_j)$). The cake-cutting problem has applications in many fields such as dividing land estates, advertisement space or broadcast time. An extension of the cake-cutting problem is the cake-cutting with two cakes problem [Cloutier et al., 2010] where two heterogeneous cakes are involved. In this problem, preferences of the agents can be coupled over the two cakes. The slice of one cake that an agent prefers might be influenced by the slice of the other cake that he or she might also obtain. The goal is to find a partition of the cakes that satisfies fairness conditions for the agents sharing the cakes. Cloutier et al. [2010] studied the envy-freeness partitioning. Both the cake-cutting and the cake-cutting with two cakes problems assume that there is only one indivisible unit of supply per element $x \in \mathcal{X}$ of the cake(s). Therefore sharing the cake(s) consists in obtaining a paritition of the set(s). In this paper, we show that EOT is a relaxation of the cutting cake and the cake-cutting with two cakes problems, when there is a divisible amount of each element of the cake(s). In that case, cakes are no more sets but distributions that we aim to divide between the agents according to their coupled preferences.

Integral Probability Metrics. In our work, we make links with some integral probability metrics. IPMs are (semi-)metrics on the space of probability measures. For a set of functions $\mathcal{F}$ and two probability distributions μ and ν, they are defined as

$$\mathrm{IPM}_{\mathcal{F}}(\mu,\nu) = \sup_{f\in\mathcal{F}} \int f d\mu - \int f d\nu.$$

For instance, when $\mathcal{F}$ is chosen to be the set of bounded functions with uniform norm less or equal than 1, we recover the Total Variation distance [Steerneman, 1983] (TV). They recently regained interest in the Machine Learning community thanks to their application to Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] where IPMs are natural metrics for the discriminator [Dziugaite et al., 2015, Arjovsky et al., 2017, Mroueh and Sercu, 2017, Husain et al., 2019]. They also helped to build consistent two-sample tests [Gretton et al., 2012, Scetbon and Varoquaux, 2019a]. However when a closed form of the IPM is not available, exact computation of IPMs between discrete distributions may not be possible or can be costful. For instance, the Dudley metric can be written as a Linear Program [Sriperumbudur et al., 2012] which has at least the same complexity as standard OT. Here, we show that the Dudley metric is in fact a particular case of our problem and obtain a faster approximation thanks to the entropic regularization.

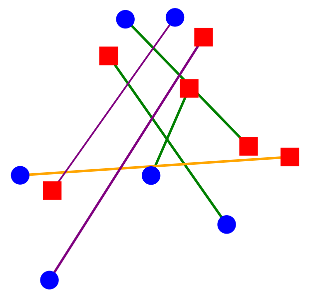


Figure F.1: Equitable and optimal division of the resources between $N = 3$ different negative costs (i.e. utilities) given by EOT. Utilities have been normalized. Blue dots and red squares represent the different elements of resources available in each cake. We consider the case where there is exactly one unit of supply per element in the cakes, which means that we consider uniform distributions. Note that the partition between the agents is equitable (i.e. utilities are equal) and proportional (i.e. utilities are larger than $1/N$).

F.3 Equitable and Optimal Transport

Notations. Let $\mathcal{Z}$ be a Polish space, we denote $\mathcal{M}(\mathcal{Z})$ the set of Radon measures on $\mathcal{Z}$. We call $\mathcal{M}_+(\mathcal{Z})$ the sets of positive Radon measures, and $\mathcal{M}_+^1(\mathcal{Z})$ the set of probability measures. We denote $\mathcal{C}^b(\mathcal{Z})$ the vector space of bounded continuous functions on $\mathcal{Z}$. Let $\mathcal{X}$ and $\mathcal{Y}$ be two Polish spaces. We denote for $\mu \in \mathcal{M}(\mathcal{X})$ and $\nu \in \mathcal{M}(\mathcal{Y})$, $\mu\otimes\nu$ the tensor product of the measures μ and ν, and $\mu \ll \nu$ means that ν dominates μ. We denote $\Pi_1 : (x, y) \in \mathcal{X}\times\mathcal{Y} \mapsto x$

and $\Pi_2 : (x, y) \in \mathcal{X} \times \mathcal{Y} \mapsto y$ respectively the projections on $\mathcal{X}$ and $\mathcal{Y}$, which are continuous applications. For an application g and a measure μ, we denote $g_\sharp\mu$ the pushforward measure of μ by g. For $\mathcal{X}$ and $\mathcal{Y}$ two Polish spaces, we denote $\mathrm{LSC}(\mathcal{X}\times\mathcal{Y})$ the space of lower semi-continuous functions on $\mathcal{X}\times\mathcal{Y}$, $\mathrm{LSC}^+(\mathcal{X}\times\mathcal{Y})$ the space of non-negative lower semi-continuous functions on $\mathcal{X}\times\mathcal{Y}$ and $\mathrm{LSC}^-_*(\mathcal{X}\times\mathcal{Y})$ the set of negative bounded below lower semi-continuous functions on $\mathcal{X}\times\mathcal{Y}$. We also denote $\mathrm{C}^+(\mathcal{X}\times\mathcal{Y})$ the space of non-negative continuous functions on $\mathcal{X}\times\mathcal{Y}$ and $\mathrm{C}^-_*(\mathcal{X}\times\mathcal{Y})$ the set of negative continuous functions on $\mathcal{X}\times\mathcal{Y}$. Let $N \geq 1$ be an integer and denote $\Delta_N^+ := \{\lambda \in \mathbb{R}_+^N \text{ s.t. } \sum_{i=1}^N \lambda_i = 1\}$, the probability simplex of $\mathbb{R}^N$. For two positive measures of same mass $\mu \in \mathcal{M}_+(\mathcal{X})$ and $\nu \in \mathcal{M}_+(\mathcal{Y})$, we define the set of couplings with marginals μ and ν:

$$\Pi_{\mu,\nu} := \{\gamma \text{ s.t. } \Pi_{1\sharp}\gamma = \mu\ ,\ \Pi_{2\sharp}\gamma = \nu\}\ .$$

We introduce the subset of $(\mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y}))^N$ representing marginal decomposition:

$$\begin{aligned}\Upsilon_{\mu,\nu}^N :=\Big\{&(\mu_i,\nu_i)_{i=1}^N \text{ s.t. } \sum_i \mu_i = \mu, \sum_i \nu_i = \nu \\ &\text{and } \forall i,\ \mu_i(\mathcal{X}) = \nu_i(\mathcal{Y})\ \Big\}.\end{aligned}$$

We also define the following subset of $\mathcal{M}_+(\mathcal{X}\times\mathcal{Y})^N$ corresponding to the coupling decomposition:

$$\Gamma_{\mu,\nu}^N := \Big\{(\gamma_i)_{i=1}^N \text{ s.t. } \Pi_{1\sharp}\sum\gamma_i = \mu\ ,\ \Pi_{2\sharp}\sum\gamma_i = \nu\Big\}.$$

F.3.1 Primal Formulation

Consider a fair division problem where several agents aim to share two sets of resources, $\mathcal{X}$ and $\mathcal{Y}$, and assume that there is a divisible amount of each resource $x \in \mathcal{X}$ (resp. $y \in \mathcal{Y}$) that is available. Formally, we consider the case where resources are no more sets but rather distributions on these sets. Denote μ and ν the distribution of resources on respectively $\mathcal{X}$ and $\mathcal{Y}$. For example, one might think about a situation where agents want to share fruit juices and ice creams and there is a certain volume of each type of fruit juices and a certain mass of each type of ice creams available. Moreover each agent defines his or her paired preferences for each couple $(x, y) \in \mathcal{X}\times\mathcal{Y}$. Formally, each person i is associated to an upper semi-continuous mapping $u_i : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}^+$ corresponding to his or her preference for any given pair (x, y). For example, one may prefer to eat chocolate ice cream with apple juice, but may prefer pineapple juice when it comes with vanilla ice cream. The total utility for an individual i and a pairing $\gamma_i \in \mathcal{M}_+(\mathcal{X}\times\mathcal{Y})$ is then given by $V_i(\gamma_i) := \int u_i d\gamma_i$. To partition fairly among individuals, we maximize the minimum of individual utilities.

From a transport point of view, let assume that there are N workers available to transport a distribution μ to another one ν. The cost of a worker i to transport a unit mass from location x to the location y is $c_i(x, y)$. To partition the work among the N workers fairly, we minimize the maximum of individual costs.

These problems are in fact the same where the utility u_i, defined in the fair division problem, might be interpreted as the opposite of the cost c_i defined in the transportation problem, i.e. for all i, $c_i = -u_i$. The two above problem motivate the introduction of EOT defined as follows.

Definition 29 (Equitable and Optimal Transport). *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. Let $\mathbf{c} := (c_i)_{1\leq i\leq N}$ be a family of bounded below lower semi-continuous cost functions on $\mathcal{X} \times \mathcal{Y}$, and $\mu \in \mathcal{M}_+^1(\mathcal{X})$ and $\nu \in \mathcal{M}_+^1(\mathcal{Y})$. We define the equitable and optimal transport primal problem:*

$$EOT_{\mathbf{c}}(\mu,\nu) := \inf_{(\gamma_i)_{i=1}^N \in \Gamma_{\mu,\nu}^N} \max_i \int c_i d\gamma_i \ . \tag{F.1}$$

We prove along with Theorem 24 that the problem is well defined and the infimum is attained. Lower-semi continuity is a standard assumption in OT. In fact, it is the weakest condition to prove Kantorovich duality [Villani, 2003, Chap. 1]. Note that the problem defined here is a linear optimization problem and when $N = 1$ we recover standard optimal transport. Figure F.1 illustrates the equitable and optimal transport problem we consider. Figure F.5 in Appendix F.9 shows an illustration with respect to the transport viewpoint in the exact same setting, i.e. $c_i = -u_i$. As expected, the couplings obtained in the two situations are not the same.

We now show that in fact, EOT optimum satisfies equality constraints in case of constant sign costs, i.e. total utility/cost of each individual are equal in the optimal partition. See Appendix F.6.2 for the proof.

Proposition 28 (EOT solves the problem under equality constraints). *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. Let $\mathbf{c} := (c_i)_{1\leq i\leq N} \in \mathrm{LSC}^+(\mathcal{X}\times\mathcal{Y})^N \cup \mathrm{LSC}_*^-(\mathcal{X}\times\mathcal{Y})^N$, $\mu \in \mathcal{M}_+^1(\mathcal{X})$ and $\nu \in \mathcal{M}_+^1(\mathcal{Y})$. Then the following are equivalent:*

- $(\gamma_i^*)_{i=1}^N \in \Gamma_{\mu,\nu}^N$ *is solution of Eq. (F.1),*
- $(\gamma_i^*)_{i=1}^N \in \underset{(\gamma_i)_{i=1}^N \in \Gamma_{\mu,\nu}^N}{\operatorname{argmin}} \left\{ t \text{ s.t. } \forall i \int c_i d\gamma_i = t \right\}$.

Moreover,

$$EOT_{\mathbf{c}}(\mu,\nu) = \min_{(\gamma_i)_{i=1}^N \in \Gamma_{\mu,\nu}^N} \left\{ t \text{ s.t. } \forall i \int c_i d\gamma_i = t \right\} .$$

This property highly relies on the sign of the costs. For instance if two costs are considered, one always positive and the other always negative, then the constraints cannot be satisfied. When the cost functions are non-negatives, EOT refers to a transportation problem while when the costs are all negatives, costs become utilities and EOT refers to a fair division problem. The two points of view are concordant, but proofs and interpretations rely on the sign of the costs.

F.3.2 An Equitable and Proportional Division

When the cost functions considered c_i are all negatives, EOT become a fair division problem where the utility functions are defined as $u_i := -c_i$. Indeed according to Proposition 28, EOT solves

$$\max_{(\gamma_i)_{i=1}^N \in \Gamma_{\mu,\nu}^N} \left\{ t \text{ s.t. } \forall i, \int u_i d\gamma_i = t \right\} .$$

Recall that in our model, the total utility of the agent i is given by $V_i(\gamma_i) := \int u_i d\gamma_i$. Therefore EOT aims to maximize the total utility of each agent i while ensuring that they are all equal. Let us now analyze which fairness conditions the partition induced by EOT verifies. Assume that the utilities are normalized, i.e., $\forall i$, there exists $\gamma_i \in \mathcal{M}_+^1(\mathcal{X}\times\mathcal{Y})$ such that $V_i(\gamma_i) = 1$. For example one might consider the cases where $\forall i$, $\gamma_i = \mu \otimes \nu$ or $\gamma_i \in \operatorname{argmin}_{\gamma\in\Pi_{\mu,\nu}} \int c_i d\gamma$. Then any solution $(\gamma_i^*)_{i=1}^N \in \Gamma_{\mu,\nu}^N$ of EOT satisfies:

- **Proportionality**: for all i, $V_i(\gamma_i^*) \geq 1/N$,
- **Equitablity**: for all i, j, $V_i(\gamma_i^*) = V_j(\gamma_j^*)$.

Proportionality is a standard fair division criterion for which a resource is divided among N agents, giving each agent at least $1/N$ of the heterogeneous resource by his/her own subjective valuation. Therefore here, this situation corresponds to the case where the normalized utility of each agent is at least $1/N$. Moreover, an equitable division is a division of an heterogeneous resource, in which each partner is equally happy with his/her share. Here this corresponds to the case where the utility of each agent are all equal.

The problem solved by EOT is a fair division problem where heterogeneous resources have to be shared among multiple agents according to their preferences. This problem is a relaxation of the two cake-cutting problem when there are a divisible amount of each item of the cakes. In that case, cakes are distributions and EOT makes a proportional and equitable partition of them. Details are left in Appendix F.6.2.

Fair Cake-cutting. Consider the case where the cake is an heterogeneous resource and there is a certain divisible quantity of each type of resource available. For example chocolate and vanilla are two types of resource present in the cake for which a certain mass is available. In that case, each type of resource in the cake is pondered by the actual quantity present in the cake. Up to a normalization, the cake is no more the set $\mathcal{X}$ but rather a distribution on this set. Note that for the two points of view to coincide, it suffices to assume that there is exactly the same amount of mass for each type of resources available in the cake. In that case, the cake can be represented by the uniform distribution over the set $\mathcal{X}$, or equivalently the set $\mathcal{X}$ itself. When cakes are distributions, the fair cutting cake problem can be interpreted as a particular case of EOT when the utilities of the agents do not depend on the variable $y \in \mathcal{Y}$. In short, we consider that utilities are functions of the form $u_i(x, y) = v_i(x)$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The normalization of utilities can be cast as follows: $\forall i, V_i(\mu) = \int v_i(x)d\mu(x) = 1$. Then Proposition 28 shows that the partition of the cake made by EOT is proportional and equitable. Note that for EOT to coincide with the classical cake-cutting problem, one needs to consider that the uniform masses of the cake associated to each type of resource cannot be splitted. This can be interpreted as a Monge formulation [Villani, 2003] of EOT which is out of the scope of this paper.

F.3.3 Optimality of EOT

We next investigate the coupling obtained by solving EOT. In the next proposition, we show that under the same assumptions of Proposition 28, EOT solutions are optimal transportation plans. See Appendix F.6.3 for the proof.

Proposition 29 (EOT realizes optimal plans). *Under the same conditions of Proposition 28, for any $(\gamma_i^*)_{i=1}^N \in \Gamma_{\mu,\nu}^N$ solution of Eq. (F.1), we have for all $i \in \{1,\dots,N\}$*

$$\begin{aligned} &\gamma_i^* \in \underset{\gamma\in\Pi_{\mu_i^*,\nu_i^*}}{\operatorname{argmin}} \int c_i d\gamma \\ \textit{where} \quad &\mu_i^* := \Pi_{1\sharp}\gamma_i^*\ ,\ \nu_i^* := \Pi_{2\sharp}\gamma_i^*\ , \end{aligned} \tag{F.2}$$

and

$$\begin{aligned} EOT_{\mathbf{c}}(\mu,\nu) = &\min_{(\mu_i,\nu_i)_{i=1}^N\in\Upsilon_{\mu,\nu}^N} t \\ &\text{s.t. } \forall i\ W_{c_i}(\mu_i,\nu_i) = t\ . \end{aligned} \tag{F.3}$$

Given the optimal matchings $(\gamma_i^*)_{i=1}^N \in \Gamma_{\mu,\nu}^N$, one can easily obtain the partition of the agents of each marginals. Indeed for all i, $\mu_i^* := \Pi_{1\sharp}\gamma_i^*$ and $\nu_i^* := \Pi_{2\sharp}\gamma_i^*$ represent respectively the portion of the agent i from distributions μ and ν.

Remark 13 (Utilitarian and Optimal Transport). *To contrast with EOT, an alternative problem is to maximize the sum of the total utilities of agents, or equivalently minimize the sum of the total costs of agents. This problem can be cast as follows:*

$$\inf_{(\gamma_i)_{i=1}^N\in\Gamma_{\mu,\nu}^N} \sum_i \int c_i d\gamma_i \tag{F.4}$$

Here one aims to maximize the total utility of all the agents, while in EOT we aim to maximize the total utility per agent under egalitarian constraint. The solution of (F.4) is not fair among agents and one can show that this problem is actually equal to $W_{\min_i(c_i)}(\mu,\nu)$. Details can be found in Appendix F.8.1.

F.3.4 Dual Formulation

Let us now introduce the dual formulation of the problem and show that strong duality holds under some mild assumptions. See Appendix F.6.4 for the proof.

Theorem 24 (Strong Duality). *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. Let $\mathbf{c} := (c_i)_{i=1}^N$ be bounded below lower semi-continuous costs. Then* strong duality holds, *i.e. for $(\mu,\nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$:*

$$EOT_{\mathbf{c}}(\mu,\nu) = \sup_{\substack{\lambda\in\Delta_N^+ \\ (f,g)\in\mathcal{F}_{\mathbf{c}}^\lambda}} \int f d\mu + \int g d\nu \tag{F.5}$$

where $\mathcal{F}_{\mathbf{c}}^\lambda := \{(f,g) \in \mathcal{C}^b(\mathcal{X}) \times \mathcal{C}^b(\mathcal{Y}) \text{ s.t. } \forall i \in \{1,...,N\},\ f \oplus g \leq \lambda_i c_i\}$.

This theorem holds under the same hypothesis and follows the same reasoning as the one in [Villani, 2003, Theorem 1.3]. While the primal formulation of the problem is easy to understand, we want to analyse situations where the dual variables also play a role. For that purpose we show in the next proposition a simple characterisation of the primal-dual optimality in case of constant sign cost functions. See Appendix F.6.5 for the proof.

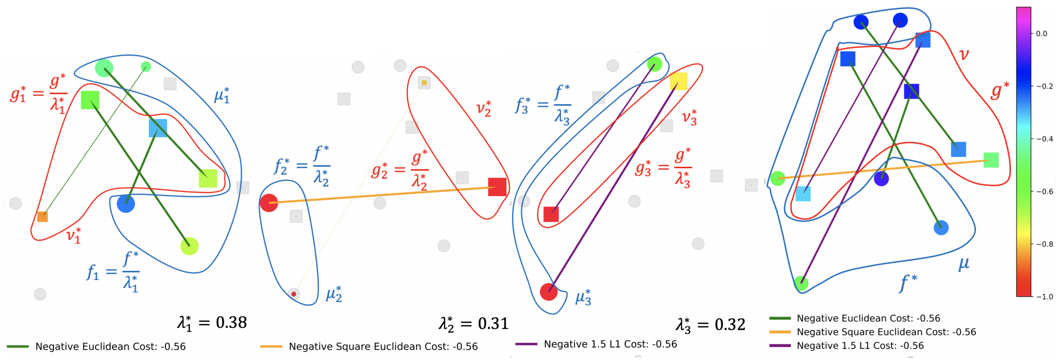


Figure F.2: *Left, middle left, middle right*: the size of dots and squares is proportional to the weight of their representing atom in the distributions μ_k^* and ν_k^* respectively. The utilities f_k^* and g_k^* for each point in respectively μ_k^* and ν_k^* are represented by the color of dots and squares according to the color scale on the right hand side. The gray dots and squares correspond to the points that are ignored by agent k in the sense that there is no mass or almost no mass in distributions μ_k^* or ν_k^*. *Right*: the size of dots and squares are uniform since they correspond to the weights of uniform distributions μ and ν respectively. The values of f^* and g^* are given also by the color at each point. Note that each agent gets exactly the same total utility, corresponding exactly to EOT. This value can be computed using dual formulation (F.5) and for each figure it equals the sum of the values (encoded with colors) multiplied by the weight of each point (encoded with sizes).

Proposition 30. *Let $\mathcal{X}$ and $\mathcal{Y}$ be compact Polish spaces. Let $\mathbf{c} := (c_i)_{1\leq i\leq N} \in \mathrm{C}^+(\mathcal{X}\times\mathcal{Y})^N \cup \mathrm{C}_*^-(\mathcal{X}\times\mathcal{Y})^N$, $\mu \in \mathcal{M}_+^1(\mathcal{X})$ and $\nu \in \mathcal{M}_+^1(\mathcal{Y})$. Let also $(\gamma_k)_{k=1}^N \in \Gamma_{\mu,\nu}^N$ and $(\lambda, f, g) \in \Delta_n^+ \times \mathcal{C}^b(\mathcal{X}) \times \mathcal{C}^b(\mathcal{Y})$. Then Eq. (F.5) admits a solution and the following are equivalent:*

- *$(\gamma_k)_{k=1}^N$ is a solution of Eq. (F.1) and (λ, f, g) is a solution of Eq. (F.5).*
- 1. *$\forall i \in \{1, ..., N\},\ f \oplus g \leq \lambda_i c_i$*
  2. *$\forall i, j \in \{1, ..., N\}\ \int c_i d\gamma_i = \int c_j d\gamma_j$*
  3. *$f \oplus g = \lambda_i c_i\ \ \gamma_i$-a.e.*

Remark 14. *It is worth noting that when we assume that $\mathbf{c} := (c_i)_{1\leq i\leq N} \in \mathrm{C}_*^+(\mathcal{X}\times\mathcal{Y})^N \cup \mathrm{C}_*^-(\mathcal{X}\times\mathcal{Y})^N$, then we can refine the second point of the equivalence presented in Proposition 30 by adding the following condition: $\forall i \in \{1, ..., N\}\ \lambda_i \neq 0$.*

Given two distributions of resources represented by the measures μ and ν, and N utility functions denoted $(u_i)_{i=1}^N$, we want to find an *equitable* and *stable* partition among the agents in case of *transferable utilities*. Let k be an agent. We say that his or her utility is transferable when once $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ get matched, he or she has to decide how to split his or her associated utility $u_k(x, y)$. She or he divides $u_k(x, y)$ into a quantity $f_k(x)$ which can be seen as the utility of having x and $g_k(y)$ for having y. Therefore in that problem we ask for $(\gamma_k, f_k, g_k)_{k=1}^N$ such that

$$u_k(x, y) = f_k(x) + g_k(y)\ \ \gamma_k\text{-a.e.} \tag{F.6}$$

Moreover, for the partition to be *stable* [Sotomayor and Roth, 1990], we want to ensure that, for every agent k, none of the resources $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ that have not been matched together for this agent would increase their utilities, $f_k(x)$ and $g_k(y)$, if there were matched together in the current matching instead. Formally we ask that for $k \in \{1, \ldots, N\}$ and all $(x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$f_k(x) + g_k(y) \geq u_k(x, y) \ . \tag{F.7}$$

Indeed if there exist k, x and y such that $u_k(x, y) > f_k(x) + g_k(y)$, then x and y will not be matched together in the share of the agent k and he can improve his utility for both x and y by matching x with y.

Finally we aim to share equitably the resources among the agents which boils down to ask

$$\forall i, j \in \{1, ..., N\} \int u_i d\gamma_i = \int u_j d\gamma_j \tag{F.8}$$

Thanks to Proposition 30, finding $(\gamma_k, f_k, g_k)_{k=1}^N$ satisfying (F.6), (F.7) and (F.8) can be done by solving Eq. (F.1) and Eq. (F.5). Indeed let $(\gamma_k)_{k=1}^N$ an optimal solution of Eq. (F.1) and (λ, f, g) an optimal solution of Eq. (F.5). Then by denoting for all $k = 1, \ldots, N$, $f_k = \frac{f}{\lambda_k}$ and $g_k = \frac{g}{\lambda_k}$, we obtain that $(\gamma_k, f_k, g_k)_{k=1}^N$ solves the *equitable* and *stable* partition problem in case of *transferable utilities*. Note that again, we end up with equality constraints for the optimal dual variables. Indeed, for all $i, j \in \{1, \ldots, N\}$, at optimality we have $\int f_i + g_i d\gamma_i = \int f_j + g_j d\gamma_j$. Figure F.2 illustrates this formulation of the problem with dual potentials. Figure F.7 in Appendix F.9 shows the dual solutions with respect to the transport viewpoint in the exact same setting, i.e. $c_i = -u_i$. Once again, the obtained solutions differ.

F.3.5 Link with other Probability Metrics

In this section, we provide some topological properties on the object defined by the EOT problem. In particular, we make links with other known probability metrics, such as Dudley and Wasserstein metrics and give a tight upper bound.

When $N = 1$, recall from the definition (F.1) that the problem considered is exactly the standard OT problem. Moreover any EOT problem with $k \leq N$ costs can always be rewritten as a EOT problem with N costs. See Appendix F.8.2 for the proof. From this property, it is interesting to note that, for any $N \geq 1$, EOT generalizes standard Optimal Transport.

Optimal Transport. Given a cost function c, if we consider the problem EOT with N costs such that, for all i, $c_i = N \times c$ then, the problem $\text{EOT}_{\mathbf{c}}$ is exactly W_c. See Appendix F.8.2 for the proof.

Now we have seen that all standard OT problems are sub-cases of the EOT problem, one may ask whether EOT can recover other families of metrics different from standard OT. Indeed we show that the EOT problem recovers an important family of IPMs with supremum taken over the space of α-Hölder functions with $\alpha \in (0, 1]$. See Appendix F.6.6 for the proof.

Proposition 31. *Let $\mathcal{X}$ be a Polish space. Let d be a metric on $\mathcal{X}^2$ and $\alpha \in (0,1]$. Denote $c_1 = 2 \times \mathbf{1}_{x \neq y}$, $c_2 = d^\alpha$ and $\mathbf{c} := (c_1, (N-1) \times c_2, ..., (N-1) \times c_2) \in \mathrm{LSC}(\mathcal{X} \times \mathcal{X})^N$ then for any $(\mu, \nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{X})$*

$$EOT_{\mathbf{c}}(\mu, \nu) = \sup_{f \in B_{d^\alpha}(\mathcal{X})} \int_{\mathcal{X}} f d\mu - \int_{\mathcal{X}} f d\nu \tag{F.9}$$

where $B_{d^\alpha}(\mathcal{X}) := \left\{ f \in C^b(\mathcal{X}) \colon \|f\|_\infty + \|f\|_\alpha \leq 1 \right\}$ and $\|f\|_\alpha := \sup_{x \neq y} \frac{|f(x) - f(y)|}{d^\alpha(x,y)}$.

Dudley Metric. When $\alpha = 1$, then for $(\mu, \nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{X})$, we have

$$\mathrm{EOT}_{\mathbf{c}}(\mu, \nu) = \mathrm{EOT}_{(c_1, d)}(\mu, \nu) = \beta_d(\mu, \nu)$$

where β_d is the *Dudley Metric* [Dudley et al., 1966]. In other words, the Dudley metric can be interpreted as an equitable and optimal transport between the measures with the trivial cost and a metric d. We acknowledge that Chizat et al. [2018] made a link between Unbalanced Optimal Transport and the "flat metric", an IPM close to the Dudley metric, defined on the space $\{f \colon \|f\|_\infty \leq 1, \ \|f\|_1 \leq 1\}$.

Weak Convergence. When d is an unbounded metric on $\mathcal{X}$, it is well known that W_{d^p} with $p \in (0, +\infty)$ metrizes a convergence a bit stronger than weak convergence [Villani, 2003, Chap. 7]. A sufficient condition for Wasserstein distances to metrize weak convergence on the space of distributions is that the metric d is bounded. In contrast, metrics defined by Eq. (F.9) do not require such assumptions and $\mathrm{EOT}_{(\mathbf{1}_{x \neq y}, d^\alpha)}$ metrizes the weak convergence of probability measures [Villani, 2003, Chap. 1-7].

For an arbitrary choice of costs $(c_i)_{1 \leq i \leq N}$, we obtain a tight upper control of EOT and show how it is related to the OT problem associated to each cost involved. See Appendix F.6.7 for the proof.

Proposition 32. *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. Let $\mathbf{c} := (c_i)_{1 \leq i \leq N}$ be a family of nonnegative lower semi-continuous costs. For any $(\mu, \nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$*

$$EOT_{\mathbf{c}}(\mu, \nu) \leq \left(\sum_{i=1}^{N} \frac{1}{W_{c_i}(\mu, \nu)} \right)^{-1} \tag{F.10}$$

Proposition 32 means that the minimal cost to transport all goods under the constraint that all workers contribute equally is lower than the case where agents share equitably and optimally the transport with distributions μ_i and ν_i respectively proportional to μ and ν, which equals the harmonic sum written in Equation (F.10).

Example. *Applying the above result in the case of the Dudley metric recovers the following inequality [Sriperumbudur et al., 2012, Proposition 5.1]*

$$\beta_d(\mu, \nu) \leq \frac{TV(\mu, \nu) W_d(\mu, \nu)}{TV(\mu, \nu) + W_d(\mu, \nu)}.$$

F.4 Entropic Relaxation

In their original form, as proposed by Kantorovich Kantorovich [1942], Optimal Transport distances are not a natural fit for applied problems: they minimize a network flow problem, with a supercubic complexity $(n^3 \log n)$ [Tarjan, 1997]. Following the work of Cuturi [2013], we propose an entropic relaxation of EOT, obtain its dual formulation and derive an efficient algorithm to compute an approximation of EOT.

F.4.1 Primal-Dual Formulation

Let us first extend the notion of Kullback-Leibler divergence for positive Radon measures. Let $\mathcal{Z}$ be a Polish space, for $\mu, \nu \in \mathcal{M}_+(\mathcal{Z})$, we define the generalized Kullback-Leibler divergence as $\mathrm{KL}(\mu||\nu) = \int \log \frac{d\mu}{d\nu} d\mu + \int d\nu - \int d\mu$ if $\mu \ll \nu$, and $+\infty$ otherwise. We introduce the following regularized version of EOT.

Definition 30 (Entropic relaxed primal problem). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two Polish spaces, $\mathbf{c} := (c_i)_{1\leq i\leq N}$ a family of bounded below lower semi-continuous costs lower semi-continuous costs on $\mathcal{X} \times \mathcal{Y}$ and $\boldsymbol{\varepsilon} := (\varepsilon_i)_{1\leq i\leq N}$ be non negative real numbers. For $(\mu, \nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$, we define the* EOT regularized primal problem*:*

$$EOT_{\mathbf{c}}^{\boldsymbol{\varepsilon}}(\mu, \nu) := \inf_{\gamma \in \Gamma_{\mu,\nu}^N} \max_i \int c_i d\gamma_i + \sum_{j=1}^N \varepsilon_j KL(\gamma_j || \mu \otimes \nu)$$

Note that here we sum the generalized Kullback-Leibler divergences since our objective is function of N measures in $\mathcal{M}_+(\mathcal{X} \times \mathcal{Y})$. This problem can be compared with the one from standard regularized OT. In the case where $N = 1$, we recover the standard regularized OT. For $N \geq 1$, the underlying problem is $\sum_{i=1}^N \varepsilon_i-$strongly convex. Moreover, we prove the essential property that as $\boldsymbol{\varepsilon} \to 0$, the regularized problem converges to the standard problem. See Appendix F.8.3 for the full statement and the proof. As a consequence, entropic regularization is a consistent approximation of the original problem we introduced in Section F.3.1. Next theorem shows that strong duality holds for lower semi-continuous costs and compact spaces. This is the basis of the algorithm we will propose in Section F.4.2. See Appendix F.6.8 for the proof.

Theorem 25 (Duality for the regularized problem). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two compact Polish spaces, $\mathbf{c} := (c_i)_{1\leq i\leq N}$ a family of bounded below lower semi-continuous costs on $\mathcal{X} \times \mathcal{Y}$ and $\boldsymbol{\varepsilon} := (\varepsilon_i)_{1\leq i\leq N}$ be non negative numbers. For $(\mu, \nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$, strong duality holds:*

$$\begin{aligned} EOT_{\mathbf{c}}^{\boldsymbol{\varepsilon}}(\mu, \nu) = \sup_{\lambda \in \Delta_N^+} \sup_{\substack{f \in \mathcal{C}_b(\mathcal{X}) \\ g \in \mathcal{C}_b(\mathcal{Y})}} & \int f d\mu + \int g d\nu \\ & - \sum_{i=1}^N \varepsilon_i \left(\int e^{\frac{f(x)+g(y)-\lambda_i c_i(x,y)}{\varepsilon_i}} d\mu(x) d\nu(y) - 1 \right) \end{aligned} \tag{F.11}$$

and the infimum of the primal problem is attained.

As in standard regularized optimal transport there is a link between primal and dual variables at optimum. Let γ^* solving the reguralized primal problem and (f^*, g^*, λ^*) solving the dual one:

$$\forall i,\ \gamma_i^* = \exp\left(\frac{f^* + g^* - \lambda_i^* c_i}{\varepsilon_i}\right) \cdot \mu \otimes \nu$$

.

F.4.2 Proposed Algorithms

Algorithm 13: Projected Alternating Maximization

Input: $\mathbf{C} = (C_i)_{1\leq i\leq N}, a, b, \varepsilon, L_\lambda$ **Init:** $f^0 \leftarrow \mathbf{1}_n;\ g^0 \leftarrow \mathbf{1}_m;$
$\lambda^0 \leftarrow (1/N, ..., 1/N) \in \mathbb{R}^N$ **for** $k = 1, 2, ...$ **do**

$K^k \leftarrow \sum_{i=1}^N K_i^{\lambda_i^{k-1}}, c_k \leftarrow \langle f^{k-1}, K^k g^{k-1}\rangle,\ f^k \leftarrow \frac{c_k a}{K^k g^{k-1}}, d_k \leftarrow \langle f^k, K^k g^{k-1}\rangle,\ g^k \leftarrow \frac{d_k b}{(K^k)^T f^k}, \lambda^k \leftarrow \mathrm{Proj}_{\Delta_N^+}\left(\lambda^{k-1} + \frac{1}{L_\lambda}\nabla_\lambda F_{\mathbf{C}}^\varepsilon(\lambda^{k-1}, f^k, g^k)\right).$

end

Result: λ, f, g

We can now present algorithms obtained from entropic relaxation to approximately compute the solution of EOT. Let $\mu = \sum_{i=1}^n a_i\delta_{x_i}$ and $\nu = \sum_{j=1}^m b_j\delta_{y_j}$ be discrete probability measures where $a \in \Delta_n^+$, $b \in \Delta_m^+$, $\{x_1, ..., x_n\} \subset \mathcal{X}$ and $\{y_1, ..., y_m\} \subset \mathcal{Y}$. Moreover for all $i \in \{1, \dots, N\}$ and $\lambda > 0$, define $\mathbf{C} := (C_i)_{1\leq i\leq N} \in (\mathbb{R}^{n\times m})^N$ with $C_i := (c_i(x_k, y_l))_{k,l}$ the N cost matrices and $K_i^\lambda := \exp(-\lambda C_i/\varepsilon)$. Assume that $\varepsilon_1 = \cdots = \varepsilon_N = \varepsilon$. Compared to the standard regularized OT, the main difference here is that the problem contains an additional variable $\lambda \in \Delta_N^+$. When $N = 1$, one can use Sinkhorn algorithm. However when $N \geq 2$, we do not have a closed form for updating λ when the other variables of the problem are fixed. In order to enjoy from the strong convexity of the primal formulation, we consider instead the dual associated with the equivalent primal problem given when the additional trivial constraint $\mathbf{1}_n^T(\sum_i P_i)\mathbf{1}_m = 1$ is considered. In that the dual obtained is

$$\widehat{\mathrm{EOT}}_{\mathbf{C}}^{\boldsymbol{\varepsilon}}(a,b) = \sup_{\substack{\lambda\in\Delta_N^+\\ f\in\mathbb{R}^n,\, g\in\mathbb{R}^m}} \langle f, a\rangle + \langle g, b\rangle - \varepsilon\left[\log\left(\sum_i \langle e^{\mathbf{f}/\varepsilon}, K_i^{\lambda_i} e^{\mathbf{g}/\varepsilon}\rangle\right) + 1\right]$$

We show that the new objective obtained above is smooth w.r.t (λ, f, g). See Appendix F.8.4 for the proof. One can apply the accelerated projected gradient ascent [Beck and Teboulle, 2009, Tseng, 2008] which enjoys an optimal convergence rate for first order methods of $\mathcal{O}(k^{-2})$ for k iterations.

It is also possible to adapt Sinkhorn algorithm to our problem. See Algorithm 13. We denoted by $\mathrm{Proj}_{\Delta_N^+}$ the orthogonal projection on Δ_N^+ [Shalev-Shwartz and Singer, 2006], whose complexity is in $\mathcal{O}(N\log N)$. The smoothness constant in λ in the algorithm is $L_\lambda = \max_i \|C_i\|_\infty^2/\varepsilon$. In practice Alg. 13 gives better results than the accelerated gradient descent. Note that the pro-

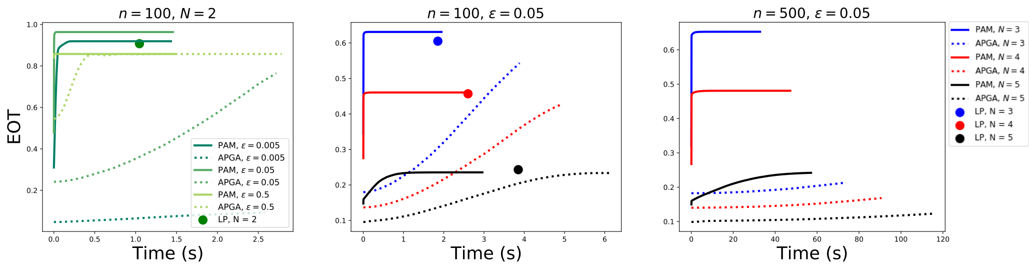


Figure F.3: Comparison of the time-accuracy tradeoffs between the different proposed algorithms. *Left:* we consider the case where the number of days is $N = 2$, the size of support for both measures is $n = m = 100$ and we vary ε from 0.005 to 0.5. *Middle:* we fix $n = m = 100$ and the regularization $\varepsilon = 0.05$ and we vary the number of days N from 3 to 5. *Right:* the setting considered is the same as in the figure in the middle, however we increase the sample size such that $n = m = 500$. Note that in that case, **LP** is too costly to be computed.

posed algorithm differs from the Sinkhorn algorithm in many points and therefore the convergence rates cannot be applied here. Analyzing the rates of a *projected* alternating maximization method is, to the best of our knowledge, an unsolved problem. Further work will be devoted to study the convergence of this algorithm. We illustrate Algorithm 13 by showing the convergence of the regularized version of EOT towards the ground truth when $\varepsilon \to 0$ in the case of the Dudley Metric. See Figure F.8 in Appendix F.9.

F.5 Other applications of EOT

Minimal Transportation Time. Assume there are N internet service providers who propose different debits to transport data across locations, and one needs to transfer data from multiple servers to others, the fastest as possible. We assume that $c_i(x, y) \geq 0$ corresponds to the transportation time needed by provider i to transport one unit of data from a server x to a server y. For instance, the unit of data can be one Megabit. Then $\int c_i d\gamma_i$ corresponds the time taken by provider i to transport $\mu_i = \Pi_{1\sharp}\gamma_i$ to $\nu_i = \Pi_{2\sharp}\gamma_i$. Assuming the transportation can be made in parallel and given a partition of the transportation task $(\gamma_i)_{i=1}^N$, $\max_i \int c_i d\gamma_i$ corresponds to the total time of transport the data $\mu = \Pi_{1\sharp}\sum\gamma_i$ to the locations $\nu = \Pi_{2\sharp}\sum\gamma_i$ according to this partition. Then EOT, which minimizes $\max_i \int c_i d\gamma_i$, is finding the fastest way to transport the data from μ to ν by splitting the task among the N internet service providers. Note that at optimality, all the internet service providers finish their transportation task at the same time (see Proposition 28).

Sequential Optimal Transport. Consider the situation where an agent aims to transport goods from some stocks to some stores in the next N days. The cost to transport one unit of good from a stock located at x to a store located at y may vary across the days. For example the cost of transportation may depend on the price of gas, or the daily weather conditions. Assuming that he or she has a good knowledge of the daily costs of the N coming days, he or she may want a transportation strategy such that his or her daily cost is as low as possible. By denoting c_i the cost of transportation the i-th day, and given a strategy $(\gamma_i)_i^N$, the maximum daily cost is then $\max_i \int c_i d\gamma_i$, and

EOT therefore finds the cheapest strategy to spread the transport task in the next N days such that the maximum daily cost is minimized. Note that at optimality he or she has to spend the exact same amount everyday.

In Figure F.3 we aim to simulate the Sequential OT problem and compare the time-accuracy trade-offs of the proposed algorithms. Let us consider a situation where one wants to transport merchandises from $\mu = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$ to $\nu = \frac{1}{m}\sum_{j=1}^{m}\delta_{y_j}$ in N days. Here we model the locations $\{x_i\}$ and $\{y_j\}$ by drawing them independently from two Gaussian distributions in $\mathbb{R}^2$: $\forall i,\ x_i \sim \mathcal{N}\left(\left(\begin{smallmatrix}3\\3\end{smallmatrix}\right), \left(\begin{smallmatrix}0 & 1\\1 & 0\end{smallmatrix}\right)\right)$ and $\forall j,\ y_j \sim \mathcal{N}\left(\left(\begin{smallmatrix}4\\4\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & -.2\\-.2 & 1\end{smallmatrix}\right)\right)$. We assume that everyday there is wind modeled by a vector $w \sim \mathcal{U}(B(0,1))$ where $B(0,1)$ is the unit ball in $\mathbb{R}^2$ that is perfectly known in advance. We define the cost of transportation on day i as $c_i(x,y) = \|y-x\| - 0.7\langle w_i, y-x\rangle$ to model the effect of the wind on the transportation cost. In the following figures we plot the estimates of EOT obtained from the proposed algorithms in function of the runtime for various sample sizes n, number of days N and regularizations ε. **PAM** denotes Alg. 13, **APGA** denotes Alg. 14 (See Appendix C.4), **LP** denotes the linear program which solves exactly the primal formulation of the EOT problem. Note that when **LP** is computable (i.e. $n \leq 100$), it is therefore the ground truth. We show that in all the settings, **PAM** performs better than **APGA** and provides very high accuracy with order of magnitude faster than LP.

F.6 Appendix: Proofs

F.6.1 Notations

Let $\mathcal{Z}$ be a Polish space, we denote $\mathcal{M}(\mathcal{Z})$ the set of Radon measures on $\mathcal{Z}$ endowed with total variation norm: $\|\mu\|_{\mathrm{TV}} = \mu_+(\mathcal{Z}) + \mu_-(\mathcal{Z})$ with (μ_+, μ_-) is the Dunford decomposition of the signed measure μ. We call $\mathcal{M}_+(\mathcal{Z})$ the sets of positive Radon measures, and $\mathcal{M}_+^1(\mathcal{Z})$ the set of probability measures. We denote $\mathcal{C}^b(\mathcal{Z})$ the vector space of bounded continuous functions on $\mathcal{Z}$ endowed with $\|\cdot\|_\infty$ norm. We recall the *Riesz-Markov theorem*: if $\mathcal{Z}$ is compact, $\mathcal{M}(\mathcal{Z})$ is the topological dual of $\mathcal{C}^b(\mathcal{Z})$. Let $\mathcal{X}$ and $\mathcal{Y}$ be two Polish spaces. It is immediate that $\mathcal{X} \times \mathcal{Y}$ *is a Polish space*. We denote for $\mu \in \mathcal{M}(\mathcal{X})$ and $\nu \in \mathcal{M}(\mathcal{Y})$, $\mu \otimes \nu$ the tensor product of the measures μ and ν, and $\mu \ll \nu$ means that ν dominates μ. We denote $\Pi_1 : (x,y) \in \mathcal{X} \times \mathcal{Y} \mapsto x$ and $\Pi_2 : (x,y) \in \mathcal{X} \times \mathcal{Y} \mapsto y$ respectively the projections on $\mathcal{X}$ and $\mathcal{Y}$, which are continuous applications. For an application g and a measure μ, we denote $g_\sharp\mu$ the pushforward measure of μ by g. For $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{Y} \to \mathbb{R}$, we denote $f \oplus g : (x,y) \in \mathcal{X} \times \mathcal{Y} \mapsto f(x) + g(y)$ the tensor sum of f and g. For $\mathcal{X}$ and $\mathcal{Y}$ two Polish spaces, we denote $\mathrm{LSC}(\mathcal{X} \times \mathcal{Y})$ the space of lower semi-continuous functions on $\mathcal{X} \times \mathcal{Y}$, $\mathrm{LSC}^+(\mathcal{X} \times \mathcal{Y})$ the space of non-negative lower semi-continuous functions on $\mathcal{X} \times \mathcal{Y}$ and $\mathrm{LSC}_*^-(\mathcal{X} \times \mathcal{Y})$ the set of negative bounded below lower semi-continuous functions on $\mathcal{X} \times \mathcal{Y}$. Let $N \geq 1$ be an integer and denote $\Delta_N^+ := \{\lambda \in \mathbb{R}_+^N \text{ s.t. } \sum_{i=1}^N \lambda_i = 1\}$, the probability simplex of $\mathbb{R}^N$. For two positive measures of same mass $\mu \in \mathcal{M}_+(\mathcal{X})$ and $\nu \in \mathcal{M}_+(\mathcal{Y})$, we define the set of couplings with marginals μ and ν:

$$\Pi_{\mu,\nu} := \{\gamma \text{ s.t. } \Pi_{1\sharp}\gamma = \mu\ ,\ \Pi_{2\sharp}\gamma = \nu\}\ .$$

For $\mu \in \mathcal{M}_+^1(\mathcal{X})$ and $\nu \in \mathcal{M}_+^1(\mathcal{Y})$, we introduce the subset of $(\mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y}))^N$ representing marginal decomposition:

$$\Upsilon_{\mu,\nu}^N := \left\{(\mu_i, \nu_i)_{i=1}^N \text{ s.t. } \textstyle\sum_i \mu_i = \mu, \sum_i \nu_i = \nu \text{ and } \forall i,\ \mu_i(\mathcal{X}) = \nu_i(\mathcal{Y})\right\}.$$

We also define the following subset of $\mathcal{M}_+(\mathcal{X} \times \mathcal{Y})^N$ corresponding to the coupling decomposition:

$$\Gamma_{\mu,\nu}^N := \left\{(\gamma_i)_{i=1}^N \text{ s.t. } \Pi_{1\sharp} \sum_i \gamma_i = \mu\ ,\ \Pi_{2\sharp} \sum_i \gamma_i = \nu\right\}.$$

F.6.2 Proof of Proposition 28

Proof. First, it is clear that $\text{EOT}_{\mathbf{c}}(\mu, \nu) \geq \inf_{\gamma \in \Gamma_{\mu,\nu}^N} \{t \text{ s.t. } \forall i,\ t = \int c_i d\gamma_i\}$. Let us now show that in fact it is an equality. Thanks to Theorem 24, the infimum is attained for $\inf_{\gamma \in \Gamma_{\mu,\nu}} \max_i \int c_i d\gamma_i$. Indeed, we recall that $\Gamma_{\mu,\nu}^N$ is compact and that the objective is lower semi-continuous. Let γ^* be such a minimizer. Let I be the set of indices i such that $\int c_i d\gamma_i^* = \text{EOT}_{\mathbf{c}}(\mu, \nu)$. Assume that there exists j such that, $\text{EOT}_{\mathbf{c}}(\mu, \nu) > \int c_j d\gamma_j^*$.

In case of costs of $\text{LSC}^+(\mathcal{X} \times \mathcal{Y})$, for all $i \in I$, there exists $(x_i, y_i) \in \text{Supp}(\gamma_i^*)$ such that $c_i(x_i, y_i) > 0$. Let us denote $A_{(x_i,y_i)}$ measurable sets such that $(x_i, y_i) \in A_{(x_i,y_i)}$ and let us denote $\tilde{\gamma}$ defined as for all $k \notin I \cup \{j\}$, $\tilde{\gamma}_k = \gamma_k^*$, for $i \in I$, $\tilde{\gamma}_i = \gamma_i^* - \epsilon \mathbf{1}_{A_{(x_i,y_i)}} \gamma_i^*$ and $\tilde{\gamma}_j = \gamma_j^* + \sum_{i \in I} \epsilon \mathbf{1}_{A_{(x_i,y_i)}} \gamma_i^*$ for ϵ sufficiently small so that $\tilde{\gamma} \in \Gamma_{\mu,\nu}^N$. Now, $\max_k \int c_k d\gamma_k^* > \max_k \int c_k d\tilde{\gamma}_k$, which contradicts that γ^* is a minimizer. Then for i, j, $\int c_i d\gamma_i^* = \int c_j d\gamma_j^*$. And then: $\text{EOT}_{\mathbf{c}}(\mu, \nu) = \inf_{\gamma \in \Gamma_{\mu,\nu}^N} \max_i \int c_i d\gamma_i$.

In case of costs in $\text{LSC}_*^-(\mathcal{X} \times \mathcal{Y})$, there exists $(x_0, y_0) \in \text{Supp}(\gamma_j^*)$ such that $c_j(x_0, y_0) < 0$. Let us denote $A_{(x_0,y_0)}$ a measurable set such that $(x_0, y_0) \in A_{(x_0,y_0)}$ and let us denote $\tilde{\gamma}$ defined as for all $k \notin I \cup \{j\}$, $\tilde{\gamma}_k = \gamma_k^*$ and for all $i \in I$, $\tilde{\gamma}_i = \gamma_i^* + \frac{\epsilon}{|I|} \mathbf{1}_{A_{(x_0,y_0)}} \gamma_j^*$ and $\tilde{\gamma}_j = \gamma_j^* - \epsilon \mathbf{1}_{A_{(x_0,y_0)}} \gamma_j^*$ for ϵ sufficiently small so that $\tilde{\gamma} \in \Gamma_{\mu,\nu}^N$. Now, $\max_k \int c_k d\gamma_k^* > \max_k \int c_k d\tilde{\gamma}_i$, which contradicts that γ^* is a minimizer. Then for i, j, $\int c_i d\gamma_i^* = \int c_j d\gamma_j^*$. And then: $\text{EOT}_{\mathbf{c}}(\mu, \nu) = \inf_{\gamma \in \Gamma_{\mu,\nu}^N} \max_i \int c_i d\gamma_i$.

It is clear that equity is verified thanks to the previous proof. For proportionality, assume the normalization: $\forall i$, there exists $\gamma_i \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$ such that $V_i(\gamma_i) = 1$. Then for each i, $V_i(\gamma_i/N) = 1/N$ and $(\gamma_i)_i \in \Gamma_{\mu,\nu}^N$. Then at optimum: $\forall i$, $V_i(\gamma_i^*) \geq 1/N$ and proportionality is verified.

□

F.6.3 Proof of Proposition 29

Proof. We prove along with Theorem 24 that the infimum defining $\text{EOT}_{\mathbf{c}}(\mu, \nu)$ is attained. Let γ^* be this infimum. Then at optimum we have shown that for all i, j, $\int c_i d\gamma_i^* = \int c_j d\gamma_j^* = t$. Let denote for all i, $\mu_i = \Pi_{1\sharp} \gamma_i^*$ and $\nu_i = \Pi_{2\sharp} \gamma_i^*$.

Let assume there exists i such that $\int c_i d\gamma_i^* > \mathrm{W}_{c_i}(\mu_i, \nu_i)$. Let γ_i' realizing the infimum of $\mathrm{W}_{c_i}(\mu_i, \nu_i)$. Let $\epsilon > 0$ be sufficiently small, then let define $\tilde{\gamma}$ as follows: for all $j \neq i$, $\tilde{\gamma}_j = (1-\epsilon)\gamma_j^*$ and $\tilde{\gamma}_i = \gamma_i' + \epsilon \sum_{j\neq i} \gamma_j^*$. Then for all $j \neq i$, $\int c_j d\tilde{\gamma}_j = (1-\epsilon)t$ and $\int c_i d\tilde{\gamma}_i = \mathrm{W}_{c_i}(\mu_i, \nu_i) + \epsilon \sum_{j\neq i} \int c_i d\gamma_j^*$. It is clear that $\tilde{\gamma} \in \Gamma^N_{\mu,\nu}$. For $\epsilon > 0$ sufficiently small, $\max_i \int c_i d\tilde{\gamma}_i = (1-\epsilon)t < t$, which contradicts the optimality of γ^*.

A possible reformulation for EOT is:

$$\mathrm{EOT}_{\mathbf{c}}(\mu, \nu) = \min_{\substack{(\mu_i,\nu_i)_{i=1}^N \in \Upsilon^N_{\mu,\nu} \\ \forall i,\ \gamma_i \in \Pi_{\mu,\nu}}} \left\{ t \text{ s.t. } \int c_i d\gamma_i = t \right\}$$

We previously show that at optimum the couplings are optimal transport plans, then:

$$\mathrm{EOT}_{\mathbf{c}}(\mu, \nu) = \min_{(\mu_i,\nu_i)_{i=1}^N \in \Upsilon^N_{\mu,\nu}} \{ t \text{ s.t. } \forall i,\ \mathrm{W}_{c_i}(\mu_i, \nu_i) = t \}$$

which concludes the proof. □

F.6.4 Proof of Theorem 24

To prove this theorem, one need to prove the three following technical lemmas. The first one shows the weak compacity of $\Gamma^N_{\mu,\nu}$.

Lemma 8. *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces, and μ and ν two probability measures respectively on $\mathcal{X}$ and $\mathcal{Y}$. Then $\Gamma^N_{\mu,\nu}$ is sequentially compact for the weak topology induced by $\|\gamma\| = \max_{i=1,..,N} \|\gamma_i\|_{TV}$.*

Proof. Let $(\gamma^n)_{n\geq 0}$ a sequence in $\Gamma^N_{\mu,\nu}$, and let us denote for all $n \geq 0$, $\gamma^n = (\gamma_i^n)_{i=1}^N$. We first remark that for all $i \in \{1, ..., N\}$ and $n \geq 0$, $\|\gamma_i^n\|_{\mathrm{TV}} \leq 1$ therefore for all $i \in \{1, ..., N\}$, $(\gamma_i^n)_{n\geq 0}$ is uniformly bounded. Moreover as $\{\mu\}$ and $\{\nu\}$ are tight, for any $\delta > 0$, there exist $K \subset \mathcal{X}$ and $L \subset \mathcal{Y}$ compact sets such that

$$\mu(K^c) \leq \frac{\delta}{2} \quad \text{and} \quad \nu(L^c) \leq \frac{\delta}{2}. \tag{F.12}$$

Therefore, we obtain that for any for all $i \in \{1, ..., N\}$,

$$\begin{aligned} \gamma_i^n(K^c \times L^c) &\leq \sum_{k=1}^N \gamma_k^n(K^c \times L^c) && \text{(F.13)} \\ &\leq \sum_{k=1}^N \gamma_k^n(K^c \times \mathcal{Y}) + \gamma_k^n(\mathcal{X} \times L^c) && \text{(F.14)} \\ &\leq \mu(K^c) + \nu(L^c) = \delta. && \text{(F.15)} \end{aligned}$$

Therefore, for all $i \in \{1, ..., N\}$, $(\gamma_i^n)_{n\geq 0}$ is tight and uniformly bounded and Prokhorov's theorem [Dupuis and Ellis, 2011, Theorem A.3.15] guarantees for all $i \in \{1, ..., N\}$, $(\gamma_i^n)_{n\geq 0}$

admits a weakly convergent subsequence. By extracting a common convergent subsequence, we obtain that $(\gamma^n)_{n\geq 0}$ admits a weakly convergent subsequence. By continuity of the projection, the limit also lives in $\Gamma^N_{\mu,\nu}$ and the result follows. □

Next lemma generalizes Rockafellar-Fenchel duality to our case.

Lemma 9. *Let V be a normed vector space and V^* its topological dual. Let $V_1, ..., V_N$ be convex functions and lower semi-continuous on V and E a convex function on V. Let $V_1^*, ...V_N^*, E^*$ be the Fenchel-Legendre transforms of $V_1, ...V_N, E$. Assume there exists $z_0 \in V$ such that for all i, $V_i(z_0) < \infty$, $E(z_0) < \infty$, and for all i, V_i is continuous at z_0. Then:*

$$\inf_{u\in V} \sum_i V_i(u) + E(u) = \sup_{\substack{\gamma_1...,\gamma_N,\gamma\in V^* \\ \sum_i \gamma_i=\gamma}} -\sum_i V_i^*(-\gamma_i) - E^*(\gamma)$$

Proof. This Lemma is an immediate application of Rockafellar-Fenchel duality theorem [Brezis, 2010, Theorem 1.12] and of Fenchel-Moreau theorem [Brezis, 2010, Theorem 1.11]. Indeed, $V = \sum\limits_{i=1}^N V_i(u)$ is a convex function, lower semi-continuous and its Legendre-Fenchel transform is given by:

$$V^*(\gamma^*) = \inf_{\sum\limits_{i=1}^N \gamma_i^*=\gamma^*} \sum_{i=1}^N V_i^*(\gamma_i^*). \tag{F.16}$$

□

Last lemma is an application of Sion's Theorem to this problem.

Lemma 10. *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. Let $\mathbf{c} = (c_i)_{1\leq i\leq N}$ be a family of bounded below lower semi-continuous costs on $\mathcal{X}\times\mathcal{Y}$, then for $\mu \in \mathcal{M}_+^1(\mathcal{X})$ and $\nu \in \mathcal{M}_+^1(\mathcal{Y})$, we have*

$$EOT_{\mathbf{c}}(\mu,\nu) = \sup_{\lambda\in\Delta_N^+} \inf_{\gamma\in\Gamma^N_{\mu,\nu}} \sum_{i=1}^N \lambda_i \int_{\mathcal{X}\times\mathcal{Y}} c_i(x,y)d\gamma_i(x,y) \tag{F.17}$$

and the infimum is attained.

Proof. Taking for granted that a minmax principle can be invoked, we have

$$\begin{aligned}\sup_{\lambda\in\Delta_N^+} \inf_{\gamma\in\Gamma^N_{\mu,\nu}} \sum_{i=1}^N \lambda_i \int_{\mathcal{X}\times\mathcal{Y}} c_i(x,y)d\gamma_i(x,y) &= \inf_{\gamma\in\Gamma^N_{\mu,\nu}} \sup_{\lambda\in\Delta_N^+} \sum_{i=1}^N \lambda_i \int_{\mathcal{X}\times\mathcal{Y}} c_i(x,y)d\gamma_i(x,y) \\ &= \mathrm{EOT}_{\mathbf{c}}(\mu,\nu)\end{aligned}$$

But thanks to Lemma 8, we have that $\Gamma^N_{\mu,\nu}$ is compact for the weak topology. And Δ^+_N is convex. The objective function $f : (\lambda, \gamma) \in \Delta^+_N \times \Gamma^N_{\mu,\nu} \mapsto \sum_{i=1}^N \lambda_i \int_{\mathcal{X}\times\mathcal{Y}} c_i^n d\gamma_i$ is bilinear, hence convex and concave in its variables, and continuous with respect to λ. Moreover, let $(c_i^n)_n$ be non-decreasing sequences of bounded cost functions such that $c_i = \sup_n c_i^n$. By monotone convergence, we get $f(\lambda, \gamma) = \sup_n \sum_i \lambda_i \int c_i^n d\gamma_i$, $f(\lambda, .)$. So f the supremum of continuous functions, then f is lower semi-continuous with respect to γ, therefore Sion's minimax theorem [Sion, 1958] holds.

□

We are now able to prove Theorem 24.

Proof. Let $\mathcal{X}$ and $\mathcal{Y}$ be two Polish spaces. For all $i \in \{1, .., N\}$, we define $c_i : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ a bounded below lower-semi cost function. The proof follows the exact same steps as those in the proof of [Villani, 2003, Theorem 1.3]. First we suppose that $\mathcal{X}$ and $\mathcal{Y}$ are compact and that for all i, c_i is continuous, then we show that it can be extended to X and Y non compact and finally to c_i only lower semi continuous.

First, let assume $\mathcal{X}$ and $\mathcal{Y}$ are compact and that for all i, c_i is continuous. Let fix $\lambda \in \Delta^+_N$. We recall the topological dual of the space of bounded continuous functions $\mathcal{C}^b(\mathcal{X} \times \mathcal{Y})$ endowed with $\|.\|_\infty$ norm, is the space of Radon measures $\mathcal{M}(\mathcal{X} \times \mathcal{Y})$ endowed with total variation norm. We define, for $u \in \mathcal{C}^b(\mathcal{X} \times \mathcal{Y})$:

$$V_i^\lambda(u) = \begin{cases} 0 & \text{if} \quad u \geq -\lambda_i c_i \\ +\infty & \text{else} \end{cases}$$

and:

$$E(u) = \begin{cases} \int f d\mu + \int g d\nu & \text{if} \quad \exists (f, g) \in \mathcal{C}^b(\mathcal{X}) \times \mathcal{C}^b(\mathcal{Y}),\ u = f + g \\ +\infty & \text{else} \end{cases}$$

One can show that for all i, V_i^λ is convex and lower semi-continuous (as the sublevel sets are closed) and E^λ is convex. More over for all i, these functions continuous in $u_0 \equiv 1$ the hypothesis of Lemma 9 are satisfied.

Let now compute the Fenchel-Legendre transform of these function. Let $\gamma \in \mathcal{M}(\mathcal{X}\times\mathcal{Y})$:

$$\begin{aligned} V_i^{\lambda *}(-\gamma) &= \sup_{u \in \mathcal{C}^b(\mathcal{X}\times\mathcal{Y})} \left\{ - \int u d\gamma; \quad u \geq -\lambda_i c_i \right\} \\ &= \begin{cases} \int \lambda_i c_i d\gamma & \text{if} \quad \gamma \in \mathcal{M}_+(\mathcal{X} \times \mathcal{Y}) \\ +\infty & \text{otherwise} \end{cases} \end{aligned}$$

On the other hand:

$$E^{\lambda *}(\gamma) = \begin{cases} 0 & \text{if} \quad \forall (f,g) \in \mathcal{C}^b(\mathcal{X}) \times \mathcal{C}^b(\mathcal{Y}),\ \int f d\mu + \int g d\nu = \int (f+g) d\gamma \\ +\infty & \text{else} \end{cases}$$

This dual function is finite and equals 0 if and only if that the marginals of the dual variable γ are μ and ν.

Applying Lemma 9, we get:

$$\inf_{u \in \mathcal{C}^b(\mathcal{X}\times\mathcal{Y})} \sum_i V_i^\lambda(u) + E(u) = \sup_{\substack{\gamma_1,\dots,\gamma_N,\gamma \in \mathcal{M}(\mathcal{X}\times\mathcal{Y}) \\ \sum \gamma_i = \gamma}} \sum -V_i^{\lambda *}(\gamma_i) - E^{\lambda *}(-\gamma)$$

Hence, we have shown that, when $\mathcal{X}$ and $\mathcal{Y}$ are compact sets, and the costs $(c_i)_i$ are continuous:

$$\sup_{(f,g)\in\mathcal{F}^\lambda_{\mathbf{c}}} \int f d\mu + \int g d\nu = \inf_{\gamma \in \Gamma^N_{\mu,\nu}} \sum_i \lambda_i \int c_i d\gamma_i$$

Let now prove the result holds when the spaces $\mathcal{X}$ and $\mathcal{Y}$ are not compact. We still suppose that for all i, c_i is uniformly continuous and bounded. We denote $\|\mathbf{c}\|_\infty := \sup_i \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} |c_i(x,y)|$. Let define $I^\lambda(\gamma) := \sum_i \lambda_i \int_{\mathcal{X}\times\mathcal{Y}} c_i d\gamma_i$

Let $\gamma^* \in \Gamma^N_{\mu,\nu}$ such that $I^\lambda(\gamma^*) = \min_{\gamma\in\Gamma^N_{\mu,\nu}} I^\lambda(\gamma)$. The existence of the minimum comes from the lower-semi continuity of I^λ and the compacity of $\Gamma^N_{\mu,\nu}$ for weak topology.

Let fix $\delta \in (0,1)$. $\mathcal{X}$ and $\mathcal{Y}$ are Polish spaces then $\exists \mathcal{X}_0 \subset \mathcal{X}, \mathcal{Y}_0 \subset \mathcal{Y}$ compacts such that $\mu(\mathcal{X}_0^c) \leq \delta$ and $\mu(\mathcal{Y}_0^c) \leq \delta$. It follows that $\forall i$, $\gamma_i^*((\mathcal{X}_0\times\mathcal{Y}_0)^c) \leq 2\delta$. Let define γ^{*0} such that for all i, $\gamma_i^{*0} = \frac{\mathbf{1}_{\mathcal{X}_0\times\mathcal{Y}_0}}{\sum_i \gamma_i^*(\mathcal{X}_0\times\mathcal{Y}_0)}\gamma_i^*$. We define $\mu_0 = \Pi_{1\sharp}\sum_i \gamma_i^{*0}$ and $\nu_0 = \Pi_{2\sharp}\sum_i \gamma_i^{*0}$. We then naturally define $\Gamma^N_{0,\mu_0,\nu_0} := \{(\gamma_i)_{1\leq i\leq N} \in \mathcal{M}_+(\mathcal{X}_0\times\mathcal{Y}_0)^N \text{ s.t. } \Pi_{1\sharp}\sum_i \gamma_i = \mu_0 \text{ and } \Pi_{2\sharp}\sum_i \gamma_i = \nu_0\}$ and $I_0^\lambda(\gamma_0) := \sum_i \lambda_i \int_{\mathcal{X}_0\times\mathcal{Y}_0} c_i d\gamma_{0,i}$ for $\gamma_0 \in \Gamma^N_{0,\mu_0,\nu_0}$.

Let $\tilde{\gamma}_0$ verifying $I_0^\lambda(\tilde{\gamma}_0) = \min_{\gamma_0\in\Pi^N_{0,\mu_0,\nu_0}} I_0^\lambda(\gamma_0)$. Let $\tilde{\gamma} = (\sum_i \gamma_i^*(\mathcal{X}_0\times\mathcal{Y}_0))\tilde{\gamma}_0 + \mathbf{1}_{(\mathcal{X}_0\times\mathcal{Y}_0)^c}\gamma^* \in \Gamma^N_{\mu,\nu}$. Then we get

$$I^\lambda(\tilde{\gamma}) \leq \min_{\gamma_0\in\Gamma^N_{0,\mu_0,\nu_0}} I_0^\lambda(\gamma_0) + 2\sum |\lambda_i| \|\mathbf{c}\|_\infty \delta$$

We have already proved that:

$$\sup_{(f,g)\in\mathcal{F}^\lambda_{0,\mathbf{c}}} J_0^\lambda(f,g) = \inf_{\gamma_0\in\Gamma^N_{0,\mu_0,\nu_0}} I_0^\lambda(\gamma_0)$$

with $J_0^\lambda(f,g) = \int f d\mu_0 + \int g d\nu_0$ and $\mathcal{F}^\lambda_{0,\mathbf{c}}$ is the set of $(f,g) \in \mathcal{C}^b(\mathcal{X}_0) \times \mathcal{C}^b(\mathcal{Y}_0)$ satisfying, for every i, $f \oplus g \leq \min_i \lambda_i c_i$. Let $(\tilde{f}_0, \tilde{g}_0) \in \mathcal{F}^\lambda_{\mathbf{0,c}}$ such that :

$$J_0^\lambda(\tilde{f}_0, \tilde{g}_0) \geq \sup_{(f,g)\in\mathcal{F}^\lambda_{0,\mathbf{c}}} J_0^\lambda(f,g) - \delta$$

Since $J_0^\lambda(0,0) = 0$, we get $\sup J_0^\lambda \geq 0$ and then, $J_0^\lambda(\tilde{f}_0, \tilde{g}_0) \geq \delta \geq -1$. For every $\gamma_0 \in \Gamma^N_{0,\mu_0,\nu_0}$:

$$J_0^\lambda(\tilde{f}_0, \tilde{g}_0) = \int (\tilde{f}_0(x) + \tilde{g}_0(y)) d\gamma_0(x,y)$$

then we have the existence of $(x_0, y_0) \in \mathcal{X}_0 \times \mathcal{Y}_0$ such that : $\tilde{f}_0(x_0) + \tilde{g}_0(y_0) \geq -1$. If we replace $(\tilde{f}_0, \tilde{g}_0)$ by $(\tilde{f}_0 - s, \tilde{g}_0 + s)$ for an accurate s, we get that: $\tilde{f}_0(x_0) \geq \frac{1}{2}$ and $\tilde{g}_0(y_0) \geq \frac{1}{2}$, and then $\forall (x,y) \in \mathcal{X}_0 \times \mathcal{Y}_0$:

$$\tilde{f}_0(x) \leq c'(x, y_0) - \tilde{g}_0(y_0) \leq c'(x, y_0) + \frac{1}{2}$$
$$\tilde{g}_0(y) \leq c'(x_0, y) - \tilde{f}_0(x_0) \leq c'(x_0, y) + \frac{1}{2}$$

where $c' := \min_i \lambda_i c_i$. Let define $\bar{f}_0(x) = \inf_{y\in\mathcal{Y}_0} c'(x,y) - \tilde{g}_0(y)$ for $x \in \mathcal{X}$. Then $\tilde{f}_0 \leq \bar{f}_0$ on $\mathcal{X}_0$. We then get $J_0^\lambda(\bar{f}_0, \tilde{g}_0) \geq J_0^\lambda(\tilde{f}_0, \tilde{g}_0)$ and $\bar{f}_0 \leq c'(., y_0) + \frac{1}{2}$ on $\mathcal{X}$. Let define $\bar{g}_0(y) = \inf_{x\in\mathcal{X}} c'(x,y) - \bar{f}_0(y)$. By construction $(f_0, g_0) \in \mathcal{F}^\lambda_{\mathbf{c}}$ since the costs are uniformly continuous and bounded and $J_0^\lambda(\bar{f}_0, \bar{g}_0) \geq J_0^\lambda(\bar{f}_0, \tilde{g}_0) \geq J_0^\lambda(\tilde{f}_0, \tilde{g}_0)$. We also have $\bar{g}_0 \geq c'(x_0, .) + \frac{1}{2}$ on $\mathcal{Y}$. Then we have in particular: $\bar{g}_0 \geq -\|\mathbf{c}\|_\infty - \frac{1}{2}$ on $\mathcal{X}$ and $\bar{f}_0 \geq -\|\mathbf{c}\|_\infty - \frac{1}{2}$ on $\mathcal{Y}$. Finally:

$$\begin{aligned}
J^\lambda(\bar{f}_0, \bar{g}_0) := &\int_{\mathcal{X}_0} \bar{f} d\mu_0 + \int_{\mathcal{Y}_0} \bar{g}_0 d\nu \\
&= \sum_i \gamma_i^*(\mathcal{X}_0 \times \mathcal{Y}_0) \int_{\mathcal{X}_0\times\mathcal{Y}_0} (\bar{f}_0(x) + \bar{g}_0(y)) d\left(\sum_i \gamma_i^{*0}(x,y)\right) \\
&+ \int_{(\mathcal{X}_0\times\mathcal{Y}_0)^c} \bar{f}_0(x) + \bar{g}_0(y) d\left(\sum_i \gamma_i^*(x,y)\right) \\
&\geq (1-2\delta)\left(\int_{\mathcal{X}_0} \bar{f}_0 d\mu_0 + \int_{\mathcal{Y}_0} \bar{g}_0 d\nu_0\right) - (2\|\mathbf{c}\|_\infty + 1) \sum_i \gamma^*((\mathcal{X}_0 \times \mathcal{Y}_0)^c) \\
&\geq (1-2\delta) J_0^\lambda(\bar{f}_0, \bar{g}_0) - 2\sum |\lambda_i| (2\|\mathbf{c}\|_\infty + 1)\delta \\
&\geq (1-2\delta) J_0^\lambda(\tilde{f}_0, \tilde{g}_0) - 2\sum |\lambda_i| (2\|\mathbf{c}\|_\infty + 1)\delta \\
&\geq (1-2\delta)(\inf I_0^\lambda - \delta) - 2\sum |\lambda_i| (2\|\mathbf{c}\|_\infty + 1)\delta
\end{aligned}$$

$$\geq (1-2\delta)(\inf I^\lambda - (2\sum|\lambda_i|\|\mathbf{c}\|_\infty + 1)\delta) - 2\sum|\lambda_i|(2\|\mathbf{c}\|_\infty + 1)\delta$$

This being true for arbitrary small δ, we get $\sup J^\lambda \geq \inf I^\lambda$. The other sens is always true then:

$$\sup_{(f,g)\in\mathcal{F}_{\mathbf{c}}^\lambda} \int f d\mu + \int g d\nu = \inf_{\gamma\in\Gamma_{\mu,\nu}^N} \sum_i \lambda_i \int c_i d\gamma_i$$

for c_i uniformly continuous and $\mathcal{X}$ and $\mathcal{Y}$ non necessarily compact.

Let now prove that the result holds for lower semi-continuous costs. Let $\mathbf{c} := (c_i)_i$ be a collection of lower semi-continuous costs. Let $(c_i^n)_n$ be non-decreasing sequences of bounded below cost functions such that $c_i = \sup_n c_i^n$. Let fix $\lambda \in \Delta_N^+$. From last step, we have shown that for all n:

$$\inf_{\gamma\in\Gamma_{\mu,\nu}^N} I_n^\lambda(\gamma) = \sup_{(f,g)\in\mathcal{F}_{\mathbf{c}^n}^\lambda} \int f d\mu + \int g d\nu \tag{F.18}$$

where $I_n^\lambda(\gamma) = \sum_i \lambda_i \int c_i^n d\gamma_i$. First it is clear that:

$$\sup_{(f,g)\in\mathcal{F}_{\mathbf{c}}^\lambda} \int f d\mu + \int g d\nu \leq \sup_{(f,g)\in\mathcal{F}_{\mathbf{c}^n}^\lambda} \int f d\mu + \int g d\nu \tag{F.19}$$

Let show that:

$$\inf_{\gamma\in\Gamma_{\mu,\nu}^N} I^\lambda(\gamma) = \sup_n \inf_{\gamma\in\Gamma_{\mu,\nu}^N} I_n^\lambda(\gamma) = \lim_n \inf_{\gamma\in\Gamma_{\mu,\nu}^N} I_n^\lambda(\gamma)$$

where $I^\lambda(\gamma) = \sum_i \lambda_i \int c_i d\gamma_i$.

Let $(\gamma^{n,k})_k$ a minimizing sequence of $\Gamma_{\mu,\nu}^N$ for the problem $\inf_{\gamma\in\Gamma_{\mu,\nu}^N} \sum_i \lambda_i \int c_i^n d\gamma_i$. By Lemma 8, up to an extraction, there exists $\gamma^n \in \Gamma_{\mu,\nu}^N$ such that $(\gamma^{n,k})_k$ converges weakly to γ^n. Then:

$$\inf_{\gamma\in\Gamma_{\mu,\nu}^N} I_n^\lambda(\gamma) = I_n^\lambda(\gamma^n)$$

Up to an extraction, there also exists $\gamma^* \in \Gamma_{\mu,\nu}^N$ such that γ^n converges weakly to γ^*. For $n \geq m$, $I_n^\lambda(\gamma^n) \geq I_m^\lambda(\gamma^n) \geq I_m^\lambda(\gamma^m)$, so by continuity of I_m^λ:

$$\lim_n I_n^\lambda(\gamma^n) \geq \limsup_n I_m^\lambda(\gamma^n) \geq I_m^\lambda(\gamma^*)$$

By monotone convergence, $I_m^\lambda(\gamma^*) \to I^\lambda(\gamma^*)$ and $\lim_n I_n^\lambda(\gamma_n) \geq I^\lambda(\gamma^*) \geq \inf_{\gamma\in\Gamma_{\mu,\nu}^N} I^\lambda(\gamma)$.

Along with Eqs. F.18 and F.19, we get that:

$$\inf_{\gamma\in\Gamma^N_{\mu,\nu}} I^\lambda(\gamma) \leq \sup_{(f,g)\in\mathcal{F}^\lambda_{\mathbf{c}}} \int f d\mu + \int g d\nu$$

The other sens being always true, we have then shown that, in the general case we still have:

$$\inf_{\gamma\in\Gamma^N_{\mu,\nu}} I^\lambda(\gamma) = \sup_{(f,g)\in\mathcal{F}^\lambda_{\mathbf{c}}} \int f d\mu + \int g d\nu$$

To conclude, we apply Lemma 10, and we get:

$$\begin{aligned}\sup_{\lambda\in\Delta^+_N} \sup_{(f,g)\in\mathcal{F}^\lambda_{\mathbf{c}}} \int f d\mu + \int g d\nu &= \sup_{\lambda\in\Delta^+_N} \inf_{\gamma\in\Gamma^N_{\mu,\nu}} I^\lambda(\gamma)\\ &= \mathrm{EOT}_{\mathbf{c}}(\mu,\nu)\end{aligned}$$

□

F.6.5 Proof of Proposition 30

Proof. Let recall that, from standard optimal transport results:

$$\mathrm{EOT}_{\mathbf{c}}(\mu,\nu) = \sup_{u\in\Phi_{\mathbf{c}}} \int u d\mu d\nu$$

with $\Phi_{\mathbf{c}} := \left\{u \in \mathcal{C}^b(\mathcal{X}\times\mathcal{Y}) \text{ s.t. } \exists\lambda\in\Delta^+_N,\ \exists\phi\in\mathcal{C}^b(\mathcal{X}),\ u = \phi^{cc}\oplus\phi^c \text{ with } c = \min_i \lambda_i c_i\right\}$ where ϕ^c is the c-transform of ϕ, i.e. for $y\in\mathcal{Y}$, $\phi^c(y) = \inf_{x\in\mathcal{X}} c(x,y) - \phi(x)$.

Let denote $\omega_1,\dots,\omega_N$ the continuity modulii of c_1, ..., c_N. The existence of continuity modulii is ensured by the uniform continuity of $c_1,\dots,c_N$ on the compact sets $\mathcal{X}\times\mathcal{Y}$ (Heine's theorem). Then a modulus of continuity for $\min_i \lambda_i c_i$ is $\sum_i \lambda_i\omega_i$. As ϕ^c and ϕ^{cc} share the same modulus of continuity than $c = \min_i \lambda_i c_i$, for u is $\Phi_{\mathbf{c}}$, a common modulus of continuity is $2\times\sum_i \omega_i$. More over, it is clear that for all x, y, $\{u(x,y) \text{ s.t. } u\in\Phi_c\}$ is compact. Then, applying Ascoli's theorem, we get, that $\Phi_{\mathbf{c}}$ is compact for $\|.\|_\infty$ norm. By continuity of $u\to\int u d\mu d\nu$, the supremum is attained, and we get the existence of the optimum u^*. The existence of optima (λ^*, f^*, g^*) immediately follows.

Let first assume that $(\gamma_k)_{k=1}^N$ is a solution of Eq. (F.1) and (λ, f, g) is a solution of Eq. (F.5). Then it is clear that for all i, j, $f\oplus g\leq \lambda_i c_i$, $(\gamma_k)_{k=1}^N\in\Gamma^N_{\mu,\nu}$ and $\int c_j d\gamma_j = \int c_i d\gamma_i$ (by Proposition 28). Let $k\in\{1,\dots,N\}$. Moreover, by Theorem 24:

$$0 = \int f d\mu + \int g d\nu - \int c_i d\gamma_i$$

$$= \sum \int (f(x) + g(y))d\gamma_i(x,y) - \sum_i \lambda_i \int c_i(x,y)d\gamma_i(x,y)$$
$$= \sum \int (f(x) + g(y) - \lambda_i c_i(x,y))d\gamma_i(x,y)$$

Since $f \oplus g \leq \lambda_i c_i$ and γ_i are positive measures then $f \oplus g = \lambda_i c_i$, γ_i-almost everywhere.

Reciprocally, let assume that there exist $(\gamma_k)_{k=1}^N \in \Gamma_{\mu,\nu}^N$ and $(\lambda, f, g) \in \Delta_n^+ \times \mathcal{C}^b(\mathcal{X}) \times \mathcal{C}^b(\mathcal{Y})$ such that $\forall i \in \{1, ..., N\}$, $f \oplus g \leq \lambda_i c_i, \forall i, j \in \{1, ..., N\}$ $\int c_i d\gamma_i = \int c_j d\gamma_j$ and $f \oplus g = \lambda_i c_i$ γ_i-a.e.. Then, for any k:

$$\begin{aligned}
\int c_k d\gamma_k &= \sum_i \lambda_i \int c_i d\gamma_i \\
&= \sum_i \int (f(x) + g(y))d\gamma_i(x,y) \\
&= \int f(x)d\mu(x) + \int g(y)d\nu(y) \\
&\leq \mathrm{EOT}_{\mathbf{c}}(\mu,\nu) \text{ by Theorem 24}
\end{aligned}$$

then γ_k is solution of the primal problem. We also have for any k:

$$\begin{aligned}
\int f d\mu + \int g d\nu &= \sum_i \int (f(x) + g(y))d\gamma_i(x,y) \\
&= \sum_i \int \lambda_i c_i d\gamma_i \\
&= \int c_k d\gamma_k \\
&\geq \mathrm{EOT}_{\mathbf{c}}(\mu,\nu)
\end{aligned}$$

□

then, thanks to Theorem 24, (λ, f, g) is solution of the dual problem.

Let now proof the result stated in Remark 14. Let assume the costs are strictly positive or strictly negative. If there exist i such that $\lambda_i = 0$, thanks to the condition $f \oplus g \leq \lambda_i c_i$, we get $f \oplus g \leq 0$ and then $f \oplus g = 0$ which contradicts the conditions $f \oplus g = \lambda_k c_k$ for all k.

F.6.6 Proof of Proposition 31

Before proving the result let us first introduce the following lemma.

Lemma 11. *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. Let $\mathbf{c} := (c_i)_{1\leq i\leq N}$ a family of bounded below continuous costs. For $(x,y) \in \mathcal{X}\times\mathcal{Y}$ and $\lambda \in \Delta_N^+$, we define*

$$c_\lambda(x,y) := \min_{i=1,\dots,N}(\lambda_i c_i(x,y))$$

then for any $(\mu,\nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$

$$EOT_{\mathbf{c}}(\mu,\nu) = \sup_{\lambda\in\Delta_N^+} W_{c_\lambda}(\mu,\nu) \tag{F.20}$$

Proof. Let $(\mu,\nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$ and $\mathbf{c} := (c_i)_{1\leq i\leq N}$ cost functions on $\mathcal{X}\times\mathcal{Y}$. Let $\lambda \in \Delta_N^+$, then by Proposition 24:

$$\mathrm{EOT}_{\mathbf{c}}(\mu,\nu) = \sup_{\lambda\in\Delta_N^+}\sup_{(f,g)\in\mathcal{F}_{\mathbf{c}}^\lambda}\int_{\mathcal{X}} f(x)d\mu(x) + \int_{\mathcal{Y}} g(y)d\nu(y)$$

Therefore by denoting $c_\lambda := \min_i(\lambda_i c_i)$ which is a continuous. The dual form of the classical Optimal Transport problem gives that:

$$\sup_{(f,g)\in\mathcal{F}_{\mathbf{c}}^\lambda}\int_{\mathcal{X}} f(x)d\mu(x) + \int_{\mathcal{Y}} g(y)d\nu(y) = \mathrm{W}_{c_\lambda}(\mu,\nu)$$

and the result follows. □

Let us now prove the result of Proposition 31.

Proof. Let μ and ν be two probability measures. Let $\alpha \in (0,1]$. Note that if d is a metric then d^α too. Therefore in the following we consider d a general metric on $\mathcal{X}\times\mathcal{X}$. Let $c_1 : (x,y) \to 2\times \mathbf{1}_{x\neq y}$ and $c_2 = d^\alpha$. For all $\lambda \in [0,1)$:

$$c_\lambda(x,y) := \min(\lambda c_1(x,y), (1-\lambda)c_2(x,y)) = \min(2\lambda, (1-\lambda)d(x,y))$$

defines a distance on $\mathcal{X}\times\mathcal{X}$. Then according to [Villani, 2003, Theorem 1.14]:

$$\mathrm{W}_{c_\lambda}(\mu,\nu) = \sup_{f \text{ s.t. } f\ 1-c_\lambda \text{ Lipschitz}} \int f d\mu - \int f d\nu$$

Then thanks to Lemma 11 we have

$$\mathrm{EOT}_{(c_1,c_2)}(\mu,\nu) = \sup_{\lambda\in[0,1], f \text{ s.t. } f\ 1-c_\lambda \text{ Lipschitz}} \int f d\mu - \int f d\nu$$

Let now prove that in this case: $\mathrm{EOT}_{(c_1,c_2)}(\mu,\nu) = \beta_d(\mu,\nu)$. Let $\lambda\in[0,1)$ and f a c_λ Lipschitz function. f is lower bounded: let $m = \inf f$ and $(u_n)_n$ a sequence satisfying $f(u_n)\to m$. Then for all x,y, $f(x)-f(y) \leq 2\lambda$ and $f(x)-f(y)\leq(1-\lambda)d(x,y)$.

Let define $g = f - m - \lambda$. For x fixed and for all n, $f(x) - f(u_n) \leq 2\lambda$, so taking the limit in n we get $f(x) - m \leq 2\lambda$. So we get that for all x, y, $g(x) \in [-\lambda, +\lambda]$ and $g(x) - g(y) \in [-(1-\lambda)d(x, zy), (1-\lambda)d(x, y)]$. Then $||g||_\infty \leq \lambda$ and $||g||_d \leq 1 - \lambda$. By construction, we also have $\int f d\mu - \int f d\nu = \int g d\mu - \int g d\nu$.Then $||g||_\infty + ||g||_d \leq 1$. So we get that $\text{EOT}_{(c_1,c_2)}(\mu, \nu) \leq \beta_d(\mu, \nu)$.
Reciprocally, let g be a function satisfying $||g||_\infty + ||g||_d \leq 1$. Let define $f = g + ||g||_\infty$ and $\lambda = ||g||_\infty$. Then, for all x, y, $f(x) \in [0, 2\lambda]$ and so $f(x) - f(y) \leq 2\lambda$. It is immediate that $f(x) - f(y) \in [-(1-\lambda)d(x, y), (1-\lambda)d(x, y)]$. Then we get $f(x) - f(y) \leq \min(\lambda, (1-\lambda)d(x, y))$. And by construction, we still have $\int f d\mu - \int f d\nu = \int g d\mu - \int g d\nu$. So $\text{EOT}_{(c_1,c_2)}(\mu, \nu) \geq \beta_d(\mu, \nu)$.

Finally we get $\text{EOT}_{(c_1,c_2)}(\mu, \nu) = \beta_d(\mu, \nu)$ when $c_1 : (x, y) \to 2 \times \mathbf{1}_{x \neq y}$ and $c_2 = d$ a distance on $\mathcal{X} \times \mathcal{X}$. ☐

F.6.7 Proof of Proposition 32

Lemma 12. *Let* $x_1, \ldots, x_N \geq 0$*, then:*

$$\sup_{\lambda \in \Delta_N^+} \min_i \lambda_i x_i = \frac{1}{\sum_i \frac{1}{x_i}}$$

Proof. First if there exists i such that $x_i = 0$, we immediately have $\sup_{\lambda \in \Delta_N^+} \min_i \lambda_i x_i = 0$. $g : \lambda \mapsto \min_i \lambda_i x_i$ is a continuous function on the compact set $\lambda \in \Delta_N^+$. Let denote λ^* the maximum of g.
Let show that for all i, j, $\lambda_i^* x_i = \lambda_j^* x_j$. Let denote $i_0, \ldots, i_k$ the indices such that $\lambda_{i_l}^* x_{i_l} = \min_i \lambda_i^* x_i$. Let assume there exists j_0 such that: $\lambda_{j_0}^* x_{j_0} > \min_i \lambda_i^* x_i$, and that all other indices i have a larger $\lambda_i^* x_i \geq \lambda_{j_0}^* x_{j_0}$. Then for $\epsilon > 0$ sufficiently small, let $\tilde{\lambda}$ defined as: $\tilde{\lambda}_{j_0} = \lambda_{j_0}^* - \epsilon$, $\tilde{\lambda}_{i_l} = \lambda_{i_l}^* + \epsilon/k$ for all $l \in \{1, \ldots, k\}$ and $\tilde{\lambda}_i = \lambda_i^*$ for all other indices. Then $\tilde{\lambda} \in \Delta_N^+$ and $g(\lambda^*) < g(\tilde{\lambda})$, which contradicts that λ^* is the maximum.
Then at the optimum for all i, j, $\lambda_i^* x_i = \lambda_j^* x_j$. So $\lambda_i^* x_i = C$ for a certain constant C. Moreover $\sum_i \lambda_i^* = 1$. Then $1/C = \sum_i 1/x_i$. Finally, for all i,

$$\lambda_i^* = \frac{1/x_i}{\sum_i 1/x_i}$$

and then:

$$\sup_{\lambda \in \Delta_N^+} \min_i \lambda_i x_i = \frac{1}{\sum_i \frac{1}{x_i}}.$$

☐

Proof. Let μ and ν be two probability measures respectively on $\mathcal{X}$ and $\mathcal{Y}$. Let $\mathbf{c} := (c_i)_i$ be a family of cost functions. Let define for $\lambda \in \Delta_N^+$, $c_\lambda(x,y) := \min_i(\lambda_i c_i(x,y))$. We have, by linearity $\mathrm{W}_{c_\lambda}(\mu,\nu) \leq \min_i(\lambda_i \mathrm{W}_{c_i}(\mu,\nu))$. So we deduce by Lemma 11:

$$\begin{aligned}\mathrm{EOT}_{\mathbf{c}}(\mu,\nu) &= \sup_{\lambda\in\Delta_N^+} \mathrm{W}_{c_\lambda}(\mu,\nu)\\ &\leq \sup_{\lambda\in\Delta_N^+} \min_i \lambda_i \mathrm{W}_{c_i}(\mu,\nu)\\ &= \frac{1}{\sum_i \frac{1}{\mathrm{W}_{c_i}(\mu,\nu)}} \text{ by Lemma 12}\end{aligned}$$

which concludes the proof. □

F.6.8 Proof of Theorem 25

Proof. To show the strong duality of the regularized problem, we use the same sketch of proof as for the strong duality of the original problem. Let first assume that, for all i, c_i is continuous on the compact set $\mathcal{X}\times\mathcal{Y}$. Let fix $\lambda \in \Delta_N^+$. We define, for all $u \in \mathcal{C}^b(\mathcal{X}\times\mathcal{Y})$:

$$V_i^\lambda(u) = \varepsilon_i\left(\int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \exp\left\{\frac{-u(x,y)-\lambda_i c_i(x,y)}{\varepsilon_i}\right\} d\mu(x)d\nu(y) - 1\right)$$

and:

$$E(u) = \begin{cases} \int f d\mu + \int g d\nu & \text{if} \quad \exists (f,g)\in\mathcal{C}^b(\mathcal{X})\times\mathcal{C}^b(\mathcal{Y}),\ u = f+g \\ +\infty & \text{else}\end{cases}$$

Let compute the Fenchel-Legendre transform of these functions. Let $\gamma \in \mathcal{M}(\mathcal{X}\times\mathcal{Y})$:

$$V_i^{\lambda *}(-\gamma) = \sup_{u\in\mathcal{C}^b(\mathcal{X}\times\mathcal{Y})} -\int u d\gamma - \varepsilon_i\left(\int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \exp\left\{\frac{-u(x,y)-\lambda_i c_i(x,y)}{\varepsilon_i}\right\} d\mu(x)d\nu(y) - 1\right)$$

However, by density of $\mathcal{C}^b(\mathcal{X}\times\mathcal{Y})$ in $L^1_{d\mu\otimes\nu}(\mathcal{X}\times\mathcal{Y})$, the set of integrable functions for $\mu\otimes\nu$ measure, we deduce that

$$V_i^{\lambda *}(-\gamma) = \sup_{u\in L^1_{d\mu\otimes\nu}(\mathcal{X}\times\mathcal{Y})} -\int u d\gamma - \varepsilon_i\left(\int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \exp\left\{\frac{-u(x,y)-\lambda_i c_i(x,y)}{\varepsilon_i}\right\} d\mu(x)d\nu(y) - 1\right)$$

This supremum equals $+\infty$ if γ is not positive and not absolutely continuous with regard to $\mu\otimes\nu$. Let us now denote $F_{\gamma,\lambda}(u) := -\int u d\gamma - \varepsilon_i\left(\int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \exp\left\{\frac{-u(x,y)-\lambda_i c_i(x,y)}{\varepsilon_i}\right\} d\mu(x)d\nu(y) - 1\right)$. F_{γ,λ_*} is Fréchet differentiable and its maximum is attained for $u^* = \varepsilon_i \log\left(\frac{d\gamma}{d\mu\otimes\nu}\right) + \lambda_i c_i$. Therefore we obtain that

$$\begin{aligned}
V_i^{\lambda *}(-\gamma) &= \varepsilon_i\left(\int \log\left(\frac{d\gamma}{d\mu\otimes\nu}\right) d\gamma + 1 - \gamma(\mathcal{X}\times\mathcal{Y})\right) + \lambda_i \int c_i d\gamma \\
&= \lambda_i \int c_i d\gamma + \varepsilon_i \mathrm{KL}(\gamma_i || \mu \times \nu)
\end{aligned}$$

Thanks to the compactness of $\mathcal{X}\times\mathcal{Y}$, all the V_i^λ for $i \in \{1,...,N\}$ are continuous on $\mathcal{C}^b(\mathcal{X}\times\mathcal{Y})$. Therefore by applying Lemma 9, we obtain that:

$$\inf_{u\in\mathcal{C}^b(\mathcal{X}\times\mathcal{Y})} \sum_i V_i^\lambda(u) + E(u) = \sup_{\substack{\gamma_1...,\gamma_N,\gamma\in\mathcal{M}(\mathcal{X}\times\mathcal{Y})\\ \sum_i \gamma_i = \gamma}} -\sum_i V_i^{\lambda *}(\gamma_i) - E^*(-\gamma)$$

$$\begin{aligned}
\sup_{f\in\mathcal{C}^b(\mathcal{X}),\, g\in\mathcal{C}^b(\mathcal{Y})} & \int f d\mu + \int g d\nu \\
& - \sum_{i=1}^N \varepsilon_i\left(\int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \exp\left\{\frac{f(x)+g(y)-\lambda_i c_i(x,y)}{\varepsilon_i}\right\} d\mu(x)d\nu(y) - 1\right) \\
& = \inf_{\gamma\in\Gamma_{\mu,\nu}^N} \sum_{i=1}^N \lambda_i \int c_i d\gamma_i + \varepsilon_i \mathrm{KL}(\gamma_i || \mu\otimes\nu)
\end{aligned}$$

Therefore by considering the supremum over the $\lambda \in \Delta_N$, we obtain that

$$\begin{aligned}
\sup_{\lambda\in\Delta_N^+} \sup_{f\in\mathcal{C}^b(\mathcal{X}),\, g\in\mathcal{C}^b(\mathcal{Y})} & \int f d\mu + \int g d\nu \\
& - \sum_{i=1}^N \varepsilon_i\left(\int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \exp\left\{\frac{f(x)+g(y)-\lambda_i c_i(x,y)}{\varepsilon_i}\right\} d\mu(x)d\nu(y) - 1\right) \\
& = \sup_{\lambda\in\Delta_N^+} \inf_{\gamma\in\Gamma_{\mu,\nu}^N} \sum_{i=1}^N \lambda_i \int c_i d\gamma_i + \varepsilon_i \mathrm{KL}(\gamma_i || \mu\otimes\nu)
\end{aligned}$$

Let $f : (\lambda,\gamma) \in \Delta_N^+ \times \Gamma_{\mu,\nu}^N \mapsto \sum_{i=1}^N \lambda_i \int c_i d\gamma_i + \varepsilon_i \mathrm{KL}(\gamma_i || \mu\otimes\nu)$. f is clearly concave and continuous in λ. Moreover $\gamma \mapsto \mathrm{KL}(\gamma_i || \mu\otimes\nu)$ is convex and lower semi-continuous for weak topology [Dupuis and Ellis, 2011, Lemma 1.4.3]. Hence f is convex and lower-semi continuous in γ. Δ_N^+ is convex, and $\Gamma_{\mu,\nu}^N$ is compact for weak topology (see Lemma 8). So by Sion's theorem, we get the expected result:

$$\begin{aligned}
&\min_{\gamma\in\Gamma_{\mu,\nu}^N}\sup_{\lambda\in\Delta_N^+}\sum_i\lambda_i\int c_i d\gamma_i+\sum_i\varepsilon_i\mathrm{KL}(\gamma_i||\mu\otimes\nu)\\
&\qquad=\sup_{\lambda\in\Delta_N^+}\sup_{(f,g)\in\mathcal{C}_b(\mathcal{X})\times\mathcal{C}_b(\mathcal{Y})}\int_{\mathcal{X}}f(x)d\mu(x)+\int_{\mathcal{Y}}g(y)d\nu(y)\\
&\qquad-\sum_{i=1}^N\varepsilon_i\left(\int_{\mathcal{X}\times\mathcal{Y}}e^{\frac{f(x)+g(y)-\lambda_i c_i(x,y)}{\varepsilon_i}}d\mu(x)d\nu(y)-1\right)
\end{aligned}$$

Moreove by fixing $\gamma\in\Gamma_{\mu,\nu}^N$, we have

$$\begin{aligned}
&\sup_{\lambda\in\Delta_N^+}\sum_i\lambda_i\int c_i d\gamma_i+\sum_i\varepsilon_i\mathrm{KL}(\gamma_i||\mu\otimes\nu)\\
&\qquad=\max_i\int c_i d\gamma_i+\sum_j\varepsilon_j\mathrm{KL}(\gamma_j||\mu\otimes\nu)
\end{aligned}$$

which concludes the proof in case of continuous costs. A similar proof as the one of the Theorem 25 allows to extend the results for lower semi-continuous cost functions. □

F.7 Appendix: Discrete cases

F.7.1 Exact discrete case

Let $a \in \Delta_N^+$ and $b \in \Delta_m^+$ and $\mathbf{C} := (C_i)_{1\leq i\leq N} \in (\mathbb{R}^{n\times m})^N$ be N cost matrices. Let also $\mathbf{X} := \{x_1, ..., x_n\}$ and $\mathbf{Y} := \{y_1, ..., y_m\}$ two subset of $\mathcal{X}$ and $\mathcal{Y}$ respectively. Moreover we define the two following discrete measure $\mu = \sum_{i=1}^n a_i\delta_{x_i}$ and $\nu = \sum_{i=1}^n b_i\delta_{y_i}$ and for all i, $C_i = (c_i(x_k, y_l))_{1\leq k\leq n, 1\leq l\leq m}$ where $(c_i)_{i=1}^N$ a family of cost functions. The discretized multiple cost optimal transport primal problem can be written as follows:

$$\mathrm{EOT}_{\mathbf{c}}(\mu,\nu) = \widehat{\mathrm{EOT}}_{\mathbf{C}}(a,b) := \inf_{P\in\Gamma_{a,b}^N} \max_i \langle P_i, C_i\rangle$$

where $\Gamma_{a,b}^N := \Big\{(P_i)_{1\leq i\leq N} \in \left(\mathbb{R}_+^{n\times m}\right)^N \text{ s.t. } (\sum_i P_i)\mathbf{1}_m = a \text{ and } (\sum_i P_i^T)\mathbf{1}_n = b\Big\}$. As in the continuous case, strong duality holds and we can rewrite the dual in the discrete case also.

Proposition 33 (Duality for the discrete problem). *Let $a \in \Delta_N^+$ and $b \in \Delta_m^+$ and $\mathbf{C} := (C_i)_{1\leq i\leq N} \in (\mathbb{R}^{n\times m})^N$ be N cost matrices. Strong duality holds for the discrete problem and*

$$\widehat{EOT}_{\mathbf{C}}(a,b) = \sup_{\lambda\in\Delta_N^+} \sup_{(f,g)\in\mathcal{F}_{\mathbf{C}}^\lambda} \langle f, a\rangle + \langle g, b\rangle.$$

where $\mathcal{F}_{\mathbf{C}}^\lambda := \{(f,g) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$ s.t. $\forall i \in \{1,...,N\},\ f\mathbf{1}_m^T + \mathbf{1}_n g^T \leq \lambda_i C_i\}$.

F.7.2 Entropic regularized discrete case

We now extend the regularization in the discrete case. Let $a \in \Delta_n^+$ and $b \in \Delta_m^+$ and $\mathbf{C} := (C_i)_{1\leq i\leq N} \in (\mathbb{R}^{n\times m})^N$ be N cost matrices and $\boldsymbol{\varepsilon} = (\varepsilon_i)_{1\leq i\leq N}$ be nonnegative real numbers. The discretized regularized primal problem is:

$$\widehat{\mathrm{EOT}}_{\mathbf{C}}^{\boldsymbol{\varepsilon}}(a,b) = \inf_{P\in\Gamma_{a,b}^N} \max_i \langle P_i, C_i\rangle - \sum_{i=1}^N \varepsilon_i \mathrm{H}(P_i)$$

where $\mathrm{H}(P) = \sum_{i,j} P_{i,j}(\log P_{i,j} - 1)$ for $P = (P_{i,j})_{i,j} \in \mathbb{R}_+^{n\times m}$ is the discrete entropy. In the discrete case, strong duality holds thanks to Lagrangian duality and Slater sufficient conditions:

Proposition 34 (Duality for the discrete regularized problem). *Let $a \in \Delta_n^+$ and $b \in \Delta_m^+$ and $\mathbf{C} := (C_i)_{1\leq i\leq N} \in (\mathbb{R}^{n\times m})^N$ be N cost matrices and $\boldsymbol{\varepsilon} := (\varepsilon_i)_{1\leq i\leq N}$ be non negative reals. Strong duality holds and by denoting $K_i^{\lambda_i} = \exp(-\lambda_i C_i/\varepsilon_i)$, we have*

$$\widehat{EOT}_{\mathbf{C}}^{\boldsymbol{\varepsilon}}(a,b) = \sup_{\lambda\in\Delta_N^+} \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m} \langle f, a\rangle + \langle g, b\rangle - \sum_{i=1}^N \varepsilon_i \langle e^{\mathbf{f}/\varepsilon_i}, K_i^{\lambda_i} e^{\mathbf{g}/\varepsilon_i}\rangle.$$

The objective function for the dual problem is strictly concave in (λ, f, g) but is neither smooth or strongly convex.

Proof. The proofs in the discrete case are simpler and only involves Lagrangian duality [Boyd et al., 2004, Chapter 5]. Let do the proof in the regularized case, the one for the standard problem follows exactly the same path.

Let $a \in \Delta_N^+$ and $b \in \Delta_m^+$ and $\mathbf{C} := (C_i)_{1\le i\le N} \in (\mathbb{R}^{n\times m})^N$ be N cost matrices.

$$
\begin{aligned}
\widehat{\mathrm{EOT}}_{\mathbf{C}}^{\boldsymbol{\varepsilon}}(a,b) &= \inf_{P\in\Gamma_{a,b}^N} \max_{1\le i\le N} \langle P_i, C_i\rangle - \sum_{i=1}^N \varepsilon_i \mathrm{H}(P_i) \\
&= \inf_{\substack{(t,P)\in\mathbb{R}\times(\mathbb{R}_+^{n\times m})^N \\ (\sum_i P_i)\mathbf{1}_m = a \\ (\sum_i P_i^T)\mathbf{1}_n = b \\ \forall j,\ \langle P_j, C_j\rangle \le t}} t - \sum_{i=1}^N \varepsilon_i \mathrm{H}(P_i) \\
&= \inf_{(t,P)\in\mathbb{R}\times(\mathbb{R}_+^{n\times m})^N}\ \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m,\, \lambda\in\mathbb{R}_+^N} t + \sum_{j=1}^N \lambda_j(\langle P_j, C_j\rangle - t) - \sum_{i=1}^N \varepsilon_i \mathrm{H}(P_i) \\
&\quad + f^T\left(a - \sum_i P_i \mathbf{1}_m\right) + g^T\left(b - \sum_i P_i^T \mathbf{1}_n\right)
\end{aligned}
$$

The constraints are qualified for this convex problem, hence by Slater's sufficient condition [Boyd et al., 2004, Section 5.2.3], strong duality holds and:

$$
\begin{aligned}
\widehat{\mathrm{EOT}}_{\mathbf{C}}^{\boldsymbol{\varepsilon}}(a,b) &= \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m,\, \lambda\in\mathbb{R}_+^N}\ \inf_{(t,P)\in\mathbb{R}\times(\mathbb{R}_+^{n\times m})^N} t + \sum_{j=1}^N \lambda_j(\langle P_j, C_j\rangle - t) - \sum_{j=1}^N \varepsilon_j \mathrm{H}(P_j) \\
&\quad + f^T\left(a - \sum_{j=1}^N P_i \mathbf{1}_m\right) + g^T\left(b - \sum_{j=1}^N P_i^T \mathbf{1}_n\right) \\
&= \sup_{\substack{f\in\mathbb{R}^n \\ g\in\mathbb{R}^m \\ \lambda\in\Delta_N^+}} \langle f, a\rangle + \langle g, b\rangle + \sum_{j=1}^N \inf_{P_j\in\mathbb{R}_+^{n\times m}} \big(\langle P_j, \lambda_j C_j - f\mathbf{1}_n^T - \mathbf{1}_m g^T\rangle - \varepsilon_j \mathrm{H}(P_j)\big)
\end{aligned}
$$

But for every $i = 1, .., N$ the solution of

$$
\inf_{P_j\in\mathbb{R}_+^{n\times m}} \big(\langle P_j, \lambda_j C_j - f\mathbf{1}_n^T - \mathbf{1}_m g^T\rangle - \varepsilon_j \mathrm{H}(P_j)\big)
$$

is

$$
P_j = \exp\left(\frac{f\mathbf{1}_n^T + \mathbf{1}_m g^T - \lambda_j C_j}{\varepsilon_i}\right)
$$

Finally we obtain that

$$\widehat{\mathrm{EOT}}^{\boldsymbol{\varepsilon}}_{\mathbf{C}}(a,b) = \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m,\, \lambda\in\Delta_N^+} \langle f,a\rangle + \langle g,b\rangle - \sum_{k=1}^{N}\varepsilon_k \sum_{i,j}\exp\left(\frac{f_i+g_j-\lambda_k C_k^{i,j}}{\varepsilon_k}\right)$$

□

F.8 Appendix: Other results

F.8.1 Utilitarian and Optimal Transport

Proposition 35. *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. Let* $\mathbf{c} := (c_i)_{1\leq i\leq N}$ *be a family of bounded below continuous cost functions on* $\mathcal{X}\times\mathcal{Y}$*, and* $\mu\in\mathcal{M}_+^1(\mathcal{X})$ *and* $\nu\in\mathcal{M}_+^1(\mathcal{Y})$*. Then we have:*

$$\inf_{(\gamma_i)_{i=1}^N\in\Gamma_{\mu,\nu}^N}\sum_i\int c_i d\gamma_i = W_{\min_i(c_i)}(\mu,\nu) \tag{F.21}$$

Proof. The proof is a by-product of the proof of Theorem 24. The continuity of the costs is necessary since $\min_i(c_i)$ is not necessarily lower semi-continuous when the costs are supposed lower semi-continuous. □

Remark 15. *We thank an anonymous reviewer for noticing that the utilitarian problem can be written also as an Optimal Transport on the space* $\mathcal{Z} = (\mathcal{X}\times\{1,\dots,N\})\times(\mathcal{Y}\times\{1,\dots,N\})$*:*

$$\min_{\gamma\in\tilde{\Gamma}_{\mu,\nu}}\int_{x,i,y,j} c((x,i),(y,j))d\gamma(x,i,y,j)$$

where the constraint space is $\tilde{\Gamma}_{\mu,\nu} := \left\{\gamma\in\mathcal{M}_1^+(\mathcal{Z})\text{ s.t. }\Pi_{\mathcal{X}}\gamma=\mu,\ \Pi_{\mathcal{Y}}\gamma=\nu\right\}$.

F.8.2 MOT generalizes OT

Proposition 36. *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces. Let* $N\geq 0$, $\mathbf{c} = (c_i)_{1\leq i\leq N}$ *be a family of nonnegative lower semi-continuous costs and let us denote for all* $k\in\{1,\dots,N\}$, $\mathbf{c}_k = (c_i)_{1\leq i\leq k}$*. Then for all* $k\in\{1,\dots,N\}$*, there exists a family of costs* $\mathbf{d}_k\in LSC(\mathcal{X}\times\mathcal{Y})^N$ *such that*

$$EOT_{\mathbf{d}_k}(\mu,\nu) = EOT_{\mathbf{c}_k}(\mu,\nu) \tag{F.22}$$

Proof. For all $k\in\{1,...,N\}$, we define $\mathbf{d}_k := (c_1,...,(N-k+1)\times c_k,...,(N-k+1)\times c_k)$. Therefore, thanks to Lemma 11 we have

$$\mathrm{EOT}_{\mathbf{d}_k}(\mu,\nu) = \sup_{\lambda\in\Delta_N^+}\mathrm{W}_{c_\lambda}(\mu,\nu) \tag{F.23}$$

$$= \sup_{(\lambda,\gamma)\in\Delta_n^k}\inf_{\gamma\in\Gamma_{\mu,\nu}}\int_{\mathcal{X}\times\mathcal{Y}}\min(\lambda_1 c_1,..,\lambda_{k-1}c_{k-1},\lambda_k c_k)d\gamma \tag{F.24}$$

where $\Delta_n^k := \{(\lambda,\gamma)\in\Delta_N^+\times\mathbb{R}_+:\quad \gamma = (N-k+1)\times\min(\lambda_k,...,\lambda_N)\}$. First remarks that

$$\gamma = 1-\sum_{i=1}^{k-1}\lambda_i \iff (N-k+1)\times\min(\lambda_k,...,\lambda_N) = \sum_{i=k}^N\lambda_i \tag{F.25}$$

$$\iff \lambda_k = ... = \lambda_N \tag{F.26}$$

But in that case $(\lambda_1, ..., \lambda_{k-1}, \gamma) \in \Delta_k$ and therefore we obtain that

$$\mathrm{EOT}_{\mathbf{d}_k}(\mu,\nu) \geq \sup_{\lambda\in\Delta_k} \inf_{\gamma\in\Gamma_{\mu,\nu}} \int_{\mathcal{X}\times\mathcal{Y}} \min(\lambda_1 c_1, .., \lambda_{k-1}c_{k-1}, \gamma c_k) d\gamma = \mathrm{EOT}_{\mathbf{c}_k}(\mu,\nu)$$

Finally by definition we have $\gamma \leq \sum_{i=k}^N \lambda_i = 1 - \sum_{i=1}^{k-1} \lambda_i$ and therefore

$$\int_{\mathcal{X}\times\mathcal{Y}} \min(\lambda_1 c_1, .., \lambda_{k-1}c_{k-1}, \gamma c_k) d\gamma \leq \int_{\mathcal{X}\times\mathcal{Y}} \min\left(\lambda_1 c_1, .., \lambda_{k-1}c_{k-1}, \left(1 - \sum_{i=1}^{k-1}\lambda_i\right) c_k\right)$$

Then we obtain that

$$\mathrm{EOT}_{\mathbf{d}_k}(\mu,\nu) \leq \mathrm{EOT}_{\mathbf{c}_k}(\mu,\nu)$$

and the result follows. □

Proposition 37. *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces and $\mathbf{c} := (c_i)_{1\leq i\leq N}$ a family of nonnegative lower semi-continuous costs on $\mathcal{X}\times\mathcal{Y}$. We suppose that, for all i, $c_i = N \times c_1$. Then for any $(\mu,\nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$*

$$EOT_{\mathbf{c}}(\mu,\nu) = EOT_{c_1}(\mu,\nu) = W_{c_1}(\mu,\nu). \tag{F.27}$$

Proof. Let $c := (c_i)_{1\leq i\leq N}$ such that for all i, $c_i = c_1$. for all $(x,y) \in \mathcal{X}\times\mathcal{Y}$ and $\lambda \in \Delta_N^+$, we have:

$$c_\lambda(x,y) := \min_i(\lambda_i c_i(x,y)) = \min_i(\lambda_i) c_1(x,y)$$

Therefore we obtain from Lemma 11 that

$$\mathrm{EOT}_c(\mu,\nu) = \sup_{\lambda\in\Delta_N^+} \mathrm{W}_{c_\lambda}(\mu,\nu) \tag{F.28}$$

But we also have that:

$$\begin{aligned}
\mathrm{W}_{c_\lambda}(\mu,\nu) &= \inf_{\gamma\in\Gamma(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \min_i(\lambda_i c_i(x,y)) d\gamma(x,y) \\
&= \min_i(\lambda_i) \inf_{\gamma\in\Gamma(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c_1(x,y) d\gamma(x,y) \\
&= \min_i(\lambda_i) \mathrm{W}_{c_1}(\mu,\nu)
\end{aligned}$$

Finally by taking the supremum over $\lambda \in \Delta_N^+$ we conclude the proof. □

F.8.3 Regularized EOT tends to EOT

Proposition 38. *For $(\mu,\nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$ we have $\lim_{\boldsymbol{\varepsilon}\to 0} EOT^{\boldsymbol{\varepsilon}}_{\mathbf{c}}(\mu,\nu) = EOT_{\mathbf{c}}(\mu,\nu)$.*

Proof. Let $(\boldsymbol{\varepsilon}_l = (\varepsilon_{l,1},\dots,\varepsilon_{l,N}))_l$ a sequence converging to 0. Let $\gamma_l = (\gamma_{l,1},\dots,\gamma_{l,N})$ be the optimum of $\mathrm{EOT}^{\boldsymbol{\varepsilon}_l}_{\mathbf{c}}(\mu,\nu)$. By Lemma 8, up to an extraction, $\gamma_l \to \gamma^\star = (\gamma_1^\star,\dots,\gamma_N^\star) \in \Gamma^N_{\mu,\nu}$. Let now $\gamma = (\gamma_1,\dots,\gamma_N)$ be the optimum of $\mathrm{EOT}_{\mathbf{c}}(\mu,\nu)$. By optimality of γ and γ_l, for all i:

$$0 \le \int c_i d\gamma_{l,i} - \int c_i d\gamma_i \le \sum_i \varepsilon_{l,i}(\mathrm{KL}(\gamma_i||\mu\otimes\nu) - \mathrm{KL}(\gamma_{l,i}||\mu\otimes\nu))$$

By lower semi continuity of $\mathrm{KL}(.||\mu\otimes\nu)$ and by taking the limit inferior as $l \to \infty$, we get for all i, $\liminf_{\ell\to\infty} \int c_i d\gamma_{l,i} = \int c_i d\gamma_i$. Moreover by continuity of $\gamma \to \int c_i d\gamma_i$ we therefore obtain that for all i, $\int c_i d\gamma_i^\star \le \int c_i d\gamma_i$. Then by optimality of γ the result follows. □

F.8.4 Projected Accelerated Gradient Descent

Proposition 39. *Let $a \in \Delta_N^+$ and $b \in \Delta_m^+$ and $\mathbf{C} := (C_i)_{1\le i\le N} \in (\mathbb{R}^{n\times m})^N$ be N cost matrices and $\boldsymbol{\varepsilon} := (\varepsilon,\dots,\varepsilon)$ where $\varepsilon > 0$. Then by denoting $K_i^{\lambda_i} = \exp(-\lambda_i C_i/\varepsilon)$, we have*

$$\widehat{EOT}^{\boldsymbol{\varepsilon}}_{\mathbf{C}}(a,b) = \sup_{\lambda\in\Delta_N^+}\ \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m} F^{\varepsilon}_{\mathbf{C}}(\lambda,f,g) := \langle f,a\rangle + \langle g,b\rangle - \varepsilon\left[\log\left(\sum_{i=1}^N \langle e^{\mathbf{f}/\varepsilon}, K_i^{\lambda_i} e^{\mathbf{g}/\varepsilon}\rangle\right) + 1\right].$$

Moreover, $F^{\varepsilon}_{\mathbf{C}}$ is concave, differentiable and ∇F is $\frac{\max\left(\max_{1\le i\le N}\|C_i\|_\infty^2, 2N\right)}{\varepsilon}$ Lipschitz-continuous on $\mathbb{R}^N \times \mathbb{R}^n \times \mathbb{R}^m$.

Proof. Let $\mathcal{Q} := \left\{P := (P_1,\dots,P_N) \in (\mathbb{R}_+^{n\times m})^N:\quad \sum_{k=1}^N \sum_{i,j} P_k^{i,j} = 1\right\}$. Note that $\Gamma^N_{a,b} \subset \mathcal{Q}$, therefore from the primal formulation of the problem we have that

$$\begin{aligned}\widehat{\mathrm{EOT}}^{\boldsymbol{\varepsilon}}_{\mathbf{C}}(a,b) &= \sup_{\lambda\in\Delta_N^+}\ \inf_{P\in\Gamma^N_{a,b}} \sum_{i=1}^N \lambda_i\langle P_i, C_i\rangle - \varepsilon \mathrm{H}(P_i)\\ &= \sup_{\lambda\in\Delta_N^+}\ \inf_{P\in\mathcal{Q}}\ \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m} \sum_{i=1}^N \lambda_i\langle P_i, C_i\rangle - \varepsilon \mathrm{H}(P_i)\\ &\quad + f^T\left(a - \sum_i P_i \mathbf{1}_m\right) + g^T\left(b - \sum_i P_i^T \mathbf{1}_n\right)\end{aligned}$$

The constraints are qualified for this convex problem, hence by Slater's sufficient condition [Boyd et al., 2004, Section 5.2.3], strong duality holds. Therefore we have

$$\begin{aligned}
\widehat{\mathrm{EOT}}^{\boldsymbol{\varepsilon}}_{\mathbf{C}}(a,b) &= \sup_{\lambda\in\Delta_N^+}\ \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m}\ \inf_{P\in\mathcal{Q}} \sum_{i=1}^N \lambda_i\langle P_i, C_i\rangle - \varepsilon\mathrm{H}(P_i)\\
&+ f^T\left(a - \sum_i P_i\mathbf{1}_m\right) + g^T\left(b - \sum_i P_i^T\mathbf{1}_n\right)\\
&= \sup_{\lambda\in\Delta_N^+}\ \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m} \langle f, a\rangle + \langle g, b\rangle\\
&+ \inf_{P\in\mathcal{Q}} \sum_{k=1}^N\sum_{i,j} P_k^{i,j}\left(\lambda_k C_k^{i,j} + \varepsilon\left(\log\left(P_k^{i,j}\right) - 1\right) - f_i - g_j\right)
\end{aligned}$$

Let us now focus on the following problem:

$$\inf_{P\in\mathcal{Q}} \sum_{k=1}^N\sum_{i,j} P_k^{i,j}\left(\lambda_k C_k^{i,j} + \varepsilon\left(\log\left(P_k^{i,j}\right) - 1\right) - f_i - g_j\right)$$

Note that for all i, j, k and some small δ,

$$P_k^{i,j}\left(\lambda_k C_k^{i,j} - \varepsilon\left(\log\left(P_k^{i,j}\right) - 1\right) - f_i - g_j\right) < 0$$

if $P_k^{i,j} \in (0,\delta)$ and this quantity goes to 0 as $P_k^{i,j}$ goes to 0. Therefore $P_k^{i,j} > 0$ and the problem becomes

$$\inf_{P>0}\ \sup_{\nu\in\mathbb{R}} \sum_{k=1}^N\sum_{i,j} P_k^{i,j}\left(\lambda_k C_k^{i,j} + \varepsilon\left(\log\left(P_k^{i,j}\right) - 1\right) - f_i - g_j\right) + \nu\left(\sum_{k=1}^N\sum_{i,j} P_k^{i,j} - 1\right).$$

The solution to this problem is for all $k \in \{1, .., N\}$,

$$P_k = \frac{\exp\left(\frac{f\mathbf{1}_n^T + \mathbf{1}_m g^T - \lambda_k C_k}{\varepsilon}\right)}{\sum_{k=1}^N \sum_{i,j} \exp\left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon}\right)}$$

Therefore we obtain that

$$\widehat{\mathrm{EOT}}^{\boldsymbol{\varepsilon}}_{\mathbf{C}}(a,b) = \sup_{\lambda\in\Delta_N^+}\ \sup_{f\in\mathbb{R}^n,\, g\in\mathbb{R}^m} \langle f, a\rangle + \langle g, b\rangle$$

$$- \varepsilon \sum_{k=1}^{N} \sum_{i,j} P_k^{i,j} \left[\log \left(\sum_{k=1}^{N} \sum_{i,j} \exp \left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon} \right) \right) + 1 \right]$$

$$= \sup_{\lambda \in \Delta_N^+} \sup_{f \in \mathbb{R}^n,\, g \in \mathbb{R}^m} \langle f, a \rangle + \langle g, b \rangle - \varepsilon \left[\log \left(\sum_{k=1}^{N} \sum_{i,j} \exp \left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon} \right) \right) + 1 \right].$$

From now on, we denote for all $\lambda \in \Delta_N^+$

$$\widehat{\mathrm{EOT}}_{\mathbf{C}}^{\varepsilon,\lambda}(a, b) := \inf_{P \in \Gamma_{a,b}^N} \sum_{i=1}^{N} \lambda_i \langle P_i, C_i \rangle - \varepsilon \mathrm{H}(P_i)$$

$$\widehat{\mathrm{EOT}}_{\mathbf{C}}^{\varepsilon,\lambda}(a, b) := \sup_{f \in \mathbb{R}^n,\, g \in \mathbb{R}^m} \langle f, a \rangle + \langle g, b \rangle - \varepsilon \left[\log \left(\sum_{k=1}^{N} \sum_{i,j} \exp \left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon} \right) \right) + 1 \right]$$

which has just been shown to be dual and equal. Thanks to [Nesterov, 2005, Theorem 1], as for all $\lambda \in \mathbb{R}^N$, $P \in \Gamma_{a,b}^N \to \sum_{i=1}^N \lambda_i \langle P_i, C_i \rangle - \varepsilon \mathrm{H}(P_i)$ is ε-strongly convex, then for all $\lambda \in \mathbb{R}^N$, $(f, g) \to \nabla_{(f,g)} F(\lambda, f, g)$ is $\frac{\|A\|_{1\to 2}^2}{\varepsilon}$ Lipschitz-continuous where A is the linear operator of the equality constraints of the primal problem. Moreover this norm is equal to the maximum Euclidean norm of a column of A. By definition, each column of A contains only $2N$ non-zero elements, which are equal to one. Hence, $\|A\|_{1\to 2} = \sqrt{2N}$. Let us now show that for all $(f, g) \in \mathbb{R}^n \times \mathbb{R}^m$ $\lambda \in \mathbb{R}^N \to \nabla_\lambda F(\lambda, f, g)$ is also Lipschitz-continuous. Indeed we remarks that

$$\frac{\partial^2 F}{\partial \lambda_q \partial \lambda_k} = \frac{1}{\varepsilon \nu^2} [\sigma_{q,1}(\lambda) \sigma_{k,1}(\lambda) - \nu(\sigma_{k,2}(\lambda) \mathbb{1}_{k=q})]$$

where $\mathbb{1}_{k=q} = 1$ iff $k = q$ and 0 otherwise, for all $k \in \{1, ..., N\}$ and $p \geq 1$

$$\sigma_{k,p}(\lambda) = \sum_{i,j} (C_k^{i,j})^p \exp \left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon} \right)$$

$$\nu = \sum_{k=1}^{N} \sum_{i,j} \exp \left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon} \right).$$

Let $v \in \mathbb{R}^N$, and by denoting $\nabla_\lambda^2 F$ the Hessian of F with respect to λ for fixed f, g we obtain first that

$$v^T \nabla_\lambda^2 F v = \frac{1}{\varepsilon \nu^2} \left[\left(\sum_{k=1}^{N} v_k \sigma_{q,1}(\lambda) \right)^2 - \nu \sum_{k=1}^{N} v_k^2 \sigma_{k,2} \right]$$

$$
\begin{aligned}
&\leq \frac{1}{\varepsilon\nu^2}\left(\sum_{k=1}^{N} v_k \sigma_{q,1}(\lambda)\right)^2 \\
&- \frac{1}{\varepsilon\nu^2}\left(\sum_{k=1}^{N} |v_k| \sqrt{\sum_{i,j} \exp\left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon}\right)} \sqrt{\sum_{i,j} (C_k^{i,j})^2 \exp\left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon}\right)}\right)^2 \\
&\leq \frac{1}{\varepsilon\nu^2}\left[\left(\sum_{k=1}^{N} v_k \sigma_{q,1}(\lambda)\right)^2 - \left(\sum_{k=1}^{N} |v_k| \sum_{i,j} |C_k^{i,j}| \exp\left(\frac{f_i + g_j - \lambda_k C_k^{i,j}}{\varepsilon}\right)\right)^2\right] \\
&\leq 0
\end{aligned}
$$

Indeed the last two inequalities come from Cauchy Schwartz. Moreover we have

$$
\begin{aligned}
\frac{1}{\varepsilon\nu^2}\left[\left(\sum_{k=1}^{N} v_k \sigma_{q,1}(\lambda)\right)^2 - \nu \sum_{k=1}^{N} v_k^2 \sigma_{k,2}\right] &= v^T \nabla_\lambda^2 F v \leq 0 \\
-\frac{\sum_{k=1}^{N} v_k^2 \sigma_{k,2}}{\varepsilon\nu} &\leq \\
-\frac{\sum_{k=1}^{N} v_k^2 \max\limits_{1\leq i\leq N}(\|C_i\|_\infty^2)}{\varepsilon} &\leq
\end{aligned}
$$

Therefore we deduce that $\lambda \in \mathbb{R}^N \to \nabla_\lambda F(\lambda, f, g)$ is $\frac{\max\limits_{1\leq i\leq N}(\|C_i\|_\infty^2)}{\varepsilon}$ Lipschitz-continuous, hence $\nabla F(\lambda, f, g)$ is $\frac{\max\left(\max\limits_{1\leq i\leq N}\|C_i\|_\infty^2, 2N\right)}{\varepsilon}$ Lipschitz-continuous on $\mathbb{R}^N \times \mathbb{R}^n \times \mathbb{R}^m$. $\square$

Denote $L := \frac{\max\left(\max\limits_{1\leq i\leq N}\|C_i\|_\infty^2, 2N\right)}{\varepsilon}$ the Lipschitz constant of $F_{\mathbf{C}}^{\varepsilon}$. Moreover for all $\lambda \in \mathbb{R}^N$, let $\text{Proj}_{\Delta_N^+}(\lambda)$ the unique solution of the following optimization problem

$$
\min_{x \in \Delta_N^+} \|x - \lambda\|_2^2. \tag{F.29}
$$

Let us now introduce the following algorithm.

Beck and Teboulle [2009], Tseng [2008] give us that the accelerated projected gradient ascent algorithm achieves the optimal rate for first order methods of $\mathcal{O}(1/k^2)$ for smooth functions. To perform the projection we use the algorithm proposed in Shalev-Shwartz and Singer [2006] which finds the solution of (F.29) after $\mathcal{O}(N\log(N))$ algebraic operations [Wang and Carreira-Perpinan, 2013].

F.8.5 Fair cutting cake problem

Let $\mathcal{X}$, be a set representing a cake. The aim of the cutting cake problem is to divide it in $\mathcal{X}_1, \ldots, \mathcal{X}_N$ disjoint sets among the N individuals. The utility for a single individual i for a slice S is denoted

Algorithm 14: Accelerated Projected Gradient Ascent Algorithm

Input: $\mathbf{C} = (C_i)_{1\leq i\leq N}, a, b, \varepsilon, L$ **Init:** $f^{-1} = f^0 \leftarrow \mathbf{0}_n;\ g^{-1} = g^0 \leftarrow \mathbf{0}_m;$
$\lambda^{-1} = \lambda^0 \leftarrow (1/N, ..., 1/N) \in \mathbb{R}^N$ **for** $k = 1, 2, ...$ **do**
$(v, w, z)^T \leftarrow (\lambda^{k-1}, f^{k-1}, g^{k-1})^T +$
$\frac{k-2}{k+1}\big((\lambda^{k-1}, f^{k-1}, g^{k-1})^T - (\lambda^{k-2}, f^{k-2}, g^{k-2})^T\big); \lambda^k \leftarrow$
$\mathrm{Proj}_{\Delta_N^+}\big(v + \frac{1}{L}\nabla_\lambda F_{\mathbf{C}}^\varepsilon(v, w, z)\big); (g^k, f^k)^T \leftarrow (w, z)^T + \frac{1}{L}\nabla_{(f,g)} F_{\mathbf{C}}^\varepsilon(v, w, z).$
end
Result: λ, f, g

$V_i(S)$. It is often assumed that $V_i(\mathcal{X}) = 1$ and that V_i is additive for disjoint sets. There exists many criteria to assess fairness for a partition $\mathcal{X}_1, \dots, \mathcal{X}_N$ such as proportionality ($V_i(\mathcal{X}_i) \geq 1/N$), envy-freeness ($V_i(\mathcal{X}_i) \geq V_i(\mathcal{X}_j)$) or equitability ($V_i(\mathcal{X}_i) = V_j(\mathcal{X}_j)$). A possible problem to solve equitability and proportionality in the cutting cake problem is the following:

$$\inf_{\substack{\mathcal{X}_1,\dots,\mathcal{X}_N \\ \sqcup_{i=1}^N \mathcal{X}_i = \mathcal{X}}} \max_i V_i(\mathcal{X}_i) \tag{F.30}$$

Note that here we do not want to solve the problem under equality constraints since the problem might not be well defined. Moreover the existence of the optimum is not immediate. A natural relaxation of this problem is when there is a divisible quantity of each element of the cake ($x \in \mathcal{X}$). In that case, the cake is no more a set but rather a distribution on this set μ. Following the primal formulation of EOT, it is clear that it is a relaxation of the cutting cake problem where the goal is to divide the cake viewed as a distribution. For the cutting cake problem with two cakes $\mathcal{X}$ and $\mathcal{Y}$, the problem can be cast as follows:

$$\inf_{\substack{\mathcal{X}_1,\dots,\mathcal{X}_N \text{ s.t. } \sqcup_{i=1}^N \mathcal{X}_i = \mathcal{X} \\ \mathcal{Y}_1,\dots,\mathcal{Y}_N \text{ s.t. } \sqcup_{i=1}^N \mathcal{Y}_i = \mathcal{Y}}} \max_i V_i(\mathcal{X}_i, \mathcal{Y}_i) \tag{F.31}$$

Here EOT is the relaxation of this problem where we split the cakes viewed as distributions instead of sets themselves. Note that in this problem, the utility of the agents are coupled.

F.9 Appendix: Illustrations and Experiments

F.9.1 Primal Formulation

Here we show the couplings obtained when we consider three negative costs $\tilde{c}_i$ which corresponds to the situation where we aim to obtain a fair division of goods between three agents. Moreover we show the couplings obtained according to the transport viewpoint where we consider the opposite of these three negative cost functions, i.e. $c_i := -\tilde{c}_i$. We can see that the couplings obtained in the two situations are completely different, which is expected. Indeed in the fair division problem, we aim at finding couplings which maximize the total utility of each agent ($\int c_i d\gamma_i^1$) while ensuring that their are equal while in the other case, we aim at finding couplings which minimize the total transportation cost of each agent ($\int c_i d\gamma_i^2$) while ensuring that their are equal. Obviously we always have that

$$\forall i \quad \int c_i d\gamma_i^2 \leq \int c_i d\gamma_i^1.$$

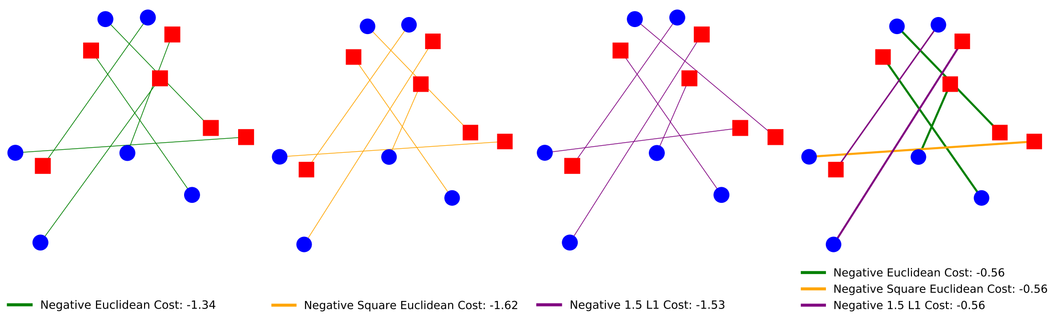


Figure F.4: Comparison of the optimal couplings obtained from standard OT for three different costs and EOT in case of negative costs (i.e. utilities). Blue dots and red squares represent the locations of two discrete uniform measures. *Left, middle left, middle right*: Kantorovich couplings between the two measures for negative Euclidean cost ($-\|\cdot\|_2$), negative square Euclidean cost ($-\|\cdot\|_2^2$) and negative 1.5 L1 norm ($-\|\cdot\|_1^{1.5}$) respectively. *Right*: Equitable and optimal division of the resources between the $N = 3$ different negative costs (i.e. utilities) given by EOT. Note that the partition between the agents is equitable (i.e. utilities are equal) and proportional (i.e. utilities are larger than $1/N$.

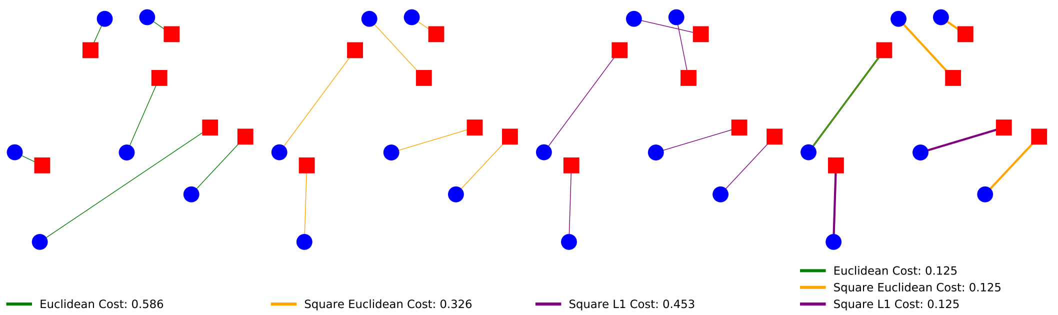


Figure F.5: Comparison of the optimal couplings obtained from standard OT for three different costs and EOT in case of postive costs. Blue dots and red squares represent the locations of two discrete uniform measures. *Left, middle left, middle right*: Kantorovich couplings between the two measures for Euclidean cost ($\|\cdot\|_2$), square Euclidean cost ($\|\cdot\|_2^2$) and 1.5 L1 norm ($\|\cdot\|_1^{1.5}$) respectively. *Right*: transport couplings of EOT solving Eq. (F.1). Note that each cost contributes equally and its contribution is lower than the smallest OT cost.

F.9.2 Dual Formulation

Here we show the dual variables obtained in the exact same settings as in the primal illustrations. Figure F.6 shows the dual associated to the primal problem exposed in Figure F.4 and Figure F.7 shows the dual associated to the primal problem exposed in Figure F.5.

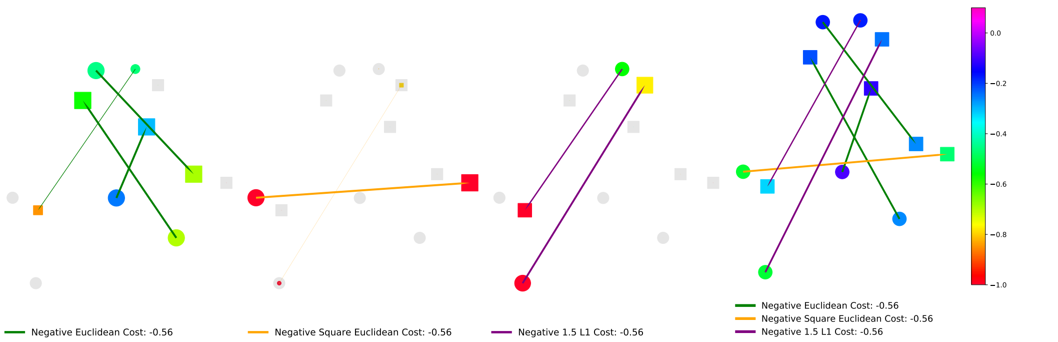


Figure F.6: *Left, middle left, middle right*: the size of dots and squares is proportional to the weight of their representing atom in the distributions μ_k^* and ν_k^* respectively. The utilities f_k^* and g_k^* for each point in respectively μ_k^* and ν_k^* are represented by the color of dots and squares according to the color scale on the right hand side. The gray dots and squares correspond to the points that are ignored by agent k in the sense that there is no mass or almost no mass in distributions μ_k^* or ν_k^*. *Right*: the size of dots and squares are uniform since they correspond to the weights of uniform distributions μ and ν respectively. The values of f^* and g^* are given also by the color at each point. Note that each agent gets exactly the same total utility, corresponding exactly to EOT. This value can be computed using dual formulation (F.5) and for each figure it equals the sum of the values (encoded with colors) multiplied by the weight of each point (encoded with sizes).

Transport viewpoint of the Dual Formulation. Assume that the N agents are not able to solve the primal problem (F.1) which aims at finding the cheapest equitable partition of the work among the N agents for transporting the distributions of goods μ to the distributions of stores ν. Moreover assume that there is an external agent who can do the transportation work for them with the following pricing scheme: he or she splits the logistic task into that of collecting and then delivering the goods, and will apply a collection price $\tilde{f}(x)$ for one unit of good located at x (no matter where that unit is sent to), and a delivery price $\tilde{g}(y)$ for one unit to the location y (no matter from which place that unit comes from). Then the external agent for transporting some goods μ to some stores ν will charge $\int_{x\in\mathcal{X}} \tilde{f}(x)d\mu(x) + \int_{y\in\mathcal{Y}} \tilde{g}(y)d\nu(y)$. However he or she has the constraint that the pricing must be equitable among the agents and therefore wants to ensure that each agent will pay exactly $\frac{1}{N}\int_{x\in\mathcal{X}} \tilde{f}(x)d\mu(x) + \int_{y\in\mathcal{Y}} \tilde{g}(y)d\nu(y)$. Denote $f = \frac{\tilde{f}}{N}$, $g = \frac{\tilde{g}}{N}$ and therefore the price paid by each agent becomes $\int_{x\in\mathcal{X}} f(x)d\mu(x) + \int_{y\in\mathcal{Y}} g(y)d\nu(y)$. Moreover, to ensure that each agent will not pay more than he would if he was doing the job himself or herself, he or she must guarantee that for all $\lambda \in \Delta_N^+$, the pricing scheme (f,g) satisfies:

$$f \oplus g \leq \min(\lambda_i c_i).$$

Indeed under this constraint, it is easy for the agents to check that they will never pay more than what they would pay if they were doing the transportation task as we have

$$\int_{x\in\mathcal{X}} f(x)d\mu(x) + \int_{y\in\mathcal{Y}} g(y)d\nu(y) \leq \int_{\mathcal{X}\times\mathcal{Y}} \min_i(\lambda_i c_i)d\gamma$$

which holds for every γ in particular for $\gamma^* = \sum_{i=1}^N \gamma_i^*$ optimal solution of the primal problem (F.1) from which follows

$$\begin{aligned}
\int_{x\in\mathcal{X}} f(x)d\mu(x) + \int_{y\in\mathcal{Y}} g(y)d\nu(y) &\leq \sum_{i=1}^N \int_{\mathcal{X}\times\mathcal{Y}} \min_i(\lambda_i c_i)d\gamma_i^* \\
&\leq \sum_{i=1}^N \lambda_i \int_{\mathcal{X}\times\mathcal{Y}} c_i d\gamma_i^* \\
&= \mathrm{EOT}_{\mathbf{c}}(\mu,\nu)
\end{aligned}$$

Therefore the external agent aims to maximise his or her selling price under the above constraints which is exactly the dual formulation of our problem.

Another interpretation of the dual problem when the cost are non-negative can be expressed as follows. Let us introduce the subset of $(\mathcal{C}^b(\mathcal{X}) \times \mathcal{C}^b(\mathcal{Y}))^N$:

$$\mathcal{G}_{\mathbf{c}}^N := \left\{(f_k, g_k)_{k=1}^N \text{ s.t. } \forall k,\ f_k \oplus g_k \leq c_k\right\}$$

Let us now show the following reformulation of the problem. See Appendix F.9.2 for the proof.

Proposition 40. *Under the same assumptions of Proposition 28, we have*

$$\begin{aligned} EOT_{\mathbf{c}}(\mu,\nu) = \sup_{(f_k,g_k)_{k=1}^N \in \mathcal{G}_{\mathbf{c}}^N} \inf_{\substack{t\in\mathbb{R} \\ (\mu_k,\nu_k)_{k=1}^N \in \Upsilon_{\mu,\nu}^N}} & \quad t \\ \text{s.t. } \forall k, \int f_k d\mu_k + \int g_k d\nu_k &= t \end{aligned} \tag{F.32}$$

Proof. Let us first introduce the following Lemma which guarantees that compacity of $\Upsilon_{\mu,\nu}^N$ for the weak topology.

Lemma 13. *Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces, and μ and ν two probability measures respectively on $\mathcal{X}$ and $\mathcal{Y}$. Then $\Upsilon_{\mu,\nu}^N$ is sequentially compact for the weak topology induced by $\|\gamma\| = \max_{i=1,..,N} \|\mu_i\|_{TV} + \|\nu_i\|_{TV}$.*

Proof. Let $(\gamma^n)_{n\geq 0}$ a sequence in $\Upsilon_{\mu,\nu}^N$, and let us denote for all $n \geq 0$, $\gamma^n = (\mu_i^n, \nu_i^n)_{i=1}^N$. We first remarks that for all $i \in \{1, ..., N\}$ and $n \geq 0$, $\|\mu_i^n\|_{\text{TV}} \leq 1$ and $\|\nu_i^n\|_{\text{TV}} \leq 1$ therefore for all $i \in \{1, ..., N\}$, $(\mu_i^n)_{n\geq 0}$ and $(\nu_i^n)_{n\geq 0}$ are uniformly bounded. Moreover as $\{\mu\}$ and $\{\nu\}$ are tight, for any $\delta > 0$, there exists $K \subset \mathcal{X}$ and $L \subset \mathcal{Y}$ compact such that $\mu(K^c) \leq \delta$ and $\nu(L^c) \leq \delta$. Then, we obtain that for any for all $i \in \{1, ..., N\}$, $\mu_i^n(K^c) \leq \delta$ and $\nu_i^n(L^c) \leq \delta$. Therefore, for all $i \in \{1, ..., N\}$, $(\mu_i^n)_{n\geq 0}$ and $(\nu_i^n)_{n\geq 0}$ are tight and uniformly bounded and Prokhorov's theorem [Dupuis and Ellis, 2011, Theorem A.3.15] guarantees for all $i \in \{1, ..., N\}$, $(\mu_i^n)_{n\geq 0}$ and $(\nu_i^n)_{n\geq 0}$ admit a weakly convergent subsequence. By extracting a common convergent subsequence, we obtain that $(\gamma^n)_{n\geq 0}$ admits a weakly convergent subsequence. By continuity of the projection, the limit also lives in $\Upsilon_{\mu,\nu}^N$ and the result follows. □

We can now prove the Proposition. We have that for any $\lambda \in \Delta_N$

$$\begin{aligned} &\sup_{(f,g)\in\mathcal{F}_{\mathbf{c}}^\lambda} \int_{x\in\mathcal{X}} f(x)d\mu(x) + \int_{y\in\mathcal{Y}} g(y)d\nu(y) \\ &\leq \sup_{(f_k,g_k)_{k=1}^N \in \mathcal{G}_{\mathbf{c}}^N} \inf_{(\mu_k,\nu_k)_{i=1}^N \in \Upsilon_{\mu,\nu}^N} \sum_{k=1}^N \lambda_k \left[\int_{x\in\mathcal{X}} f_k(x)d\mu_k(x) + \int_{y\in\mathcal{Y}} g_k(y)d\nu_k(y)\right] \\ &\leq \mathrm{EOT}_{\mathbf{c}}(\mu,\nu) \end{aligned}$$

Then by taking the supremum over $\lambda \in \Delta_N$, and by applying Theorem 24 we obtain that

$$\mathrm{EOT}_{\mathbf{c}}(\mu,\nu) = \sup_{\lambda\in\Delta_N} \sup_{(f_k,g_k)_{k=1}^N \in \mathcal{G}_{\mathbf{c}}^N} \inf_{(\mu_k,\nu_k)_{k=1}^N \in \Upsilon_{\mu,\nu}^N} \sum_{k=1}^N \lambda_k \left[\int_{x\in\mathcal{X}} f_k(x)d\mu_k(x) + \int_{y\in\mathcal{Y}} g_k(y)d\nu_k(y)\right]$$

Let $\mathcal{G}_{\mathbf{c}}^N$ and $\Upsilon_{\mu,\nu}^N$ be endowed respectively with the uniform norm and the norm defined in Lemma 13. Note that the objective is linear and continuous with respect to $(\mu_k, \nu_k)_{k=1}^N$ and also $(f_k, g_k)_{k=1}^N$. Moreover the spaces $\mathcal{G}_{\mathbf{c}}^N$ and $\Upsilon_{\mu,\nu}^N$ are clearly convex. Finally thanks to Lemma 13, $\Upsilon_{\mu,\nu}^N$ is compact with respect to the weak topology we can apply Sion's theorem Sion [1958] and we obtain that

$$\mathrm{EOT}_{\mathbf{c}}(\mu,\nu) = \sup_{(f_k,g_k)_{k=1}^N \in \mathcal{G}_{\mathbf{c}}^N} \inf_{(\mu_k,\nu_k)_{k=1}^N \in \Upsilon_{\mu,\nu}^N} \sup_{\lambda\in\Delta_N} \sum_{k=1}^N \lambda_k \left[\int_{x\in\mathcal{X}} f_k(x) d\mu_k(x) + \int_{y\in\mathcal{Y}} g_k(y) d\nu_k(y)\right]$$

Let us now fix $(f_k, g_k)_{k=1}^N \in \mathcal{G}_{\mathbf{c}}^N$ and $(\mu_k, \nu_k)_{k=1}^N \in \Upsilon_{\mu,\nu}^N$, therefore we have:

$$\begin{aligned}
&\sup_{\lambda\in\Delta_N} \sum_{k=1}^N \lambda_k \left[\int_{x\in\mathcal{X}} f_k(x) d\mu_k(x) + \int_{y\in\mathcal{Y}} g_k(y) d\nu_k(y)\right] \\
&= \sup_{\lambda} \inf_{t}\, t \times \left(1 - \sum_{i=1}^N \lambda_i\right) + \sum_{k=1}^N \lambda_k \left[\int_{x\in\mathcal{X}} f_k(x) d\mu_k(x) + \int_{y\in\mathcal{Y}} g_k(y) d\nu_k(y)\right] \\
&= \inf_{t} \sup_{\lambda}\, t + \sum_{k=1}^N \lambda_k \left[\int_{x\in\mathcal{X}} f_k(x) d\mu_k(x) + \int_{y\in\mathcal{Y}} g_k(y) d\nu_k(y) - t\right] \\
&= \inf_{t} \left\{ t \text{ s.t. } \forall k,\ \int f_k d\mu_k + \int g_k d\nu_k = t \right\}
\end{aligned}$$

where the inversion is possible as the Slater's conditions are satisfied and the result follows.

□

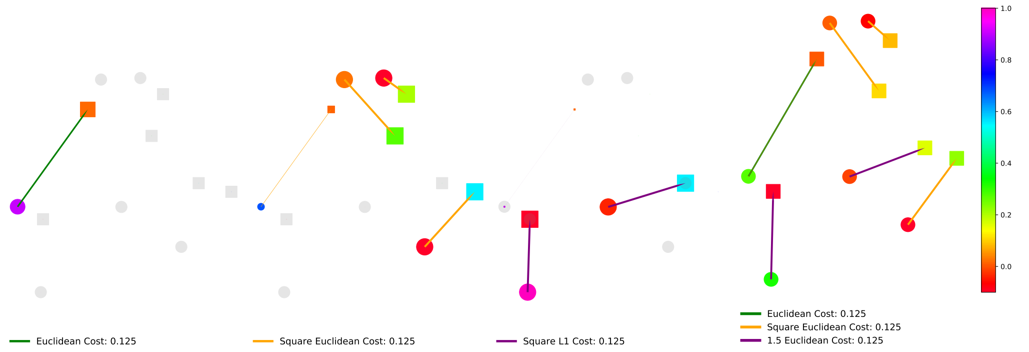


Figure F.7: *Left, middle left, middle right*: the size of dots and squares is proportional to the weight of their representing atom in the distributions μ_k^* and ν_k^* respectively. The collection "cost" f_k^* for each point in μ_k^*, and its delivery counterpart g_k^* in ν_k^* are represented by the color of dots and squares according to the color scale on the right hand side. The gray dots and squares correspond to the points that are ignored by agent k in the sense that there is no mass or almost no mass in distributions μ_k^* or ν_k^*. *Right*: the size of dots and squares are uniform since they corresponds to the weights of uniform distributions μ and ν respectively. The values of f^* and g^* are given also by the color at each point. Note that each agent earns exactly the same amount of money, corresponding exactly EOT cost. This value can be computed using dual formulation (F.5) or its reformulation (F.32) and for each figure it equals the sum of the values (encoded with colors) multiplied by the weight of each point (encoded with sizes).

F.9.3 Approximation of the Dudley Metric

Figure F.8 illustrates the convergence of the entropic regularization approximation when $\epsilon \to 0$. To do so we plot the relative error from the ground truth defined as RE $:= \frac{\mathrm{EOT}_{\mathbf{c}}^{\varepsilon} - \beta_d}{\beta_d}$ for different regularizations where β_d is obtained by solving the exact linear program and $\mathrm{EOT}_{\mathbf{c}}^{\varepsilon}$ is obtained by our proposed Alg. 13.

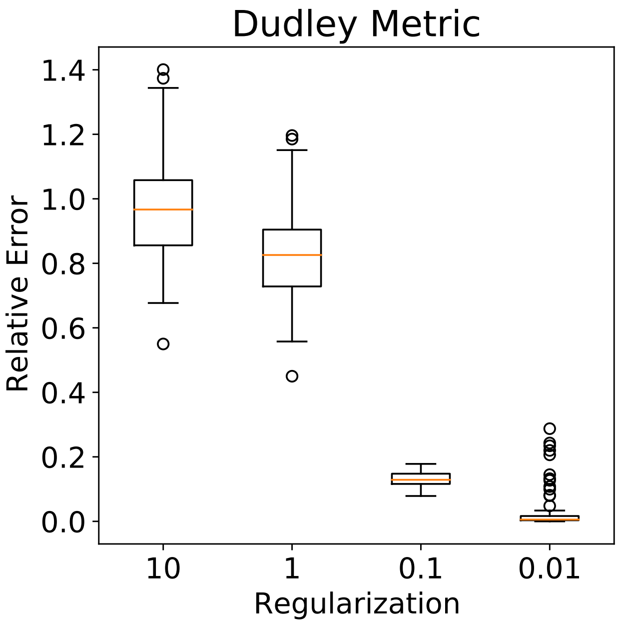


Figure F.8: In this experiment, we draw 100 samples from two normal distributions and we plot the relative error from ground truth for different regularizations. We consider the case where two costs are involved: $c_1 = 2 \times \mathbf{1}_{x \neq y}$, and $c_2 = d$ where d is the Euclidean distance. This case corresponds exactly to the Dudley metric (see Proposition 31). We remark that as $\varepsilon \to 0$, the approximation error goes also to 0.

G An Asymptotic Test for Conditional Independence using Analytic Kernel Embeddings

We propose a new conditional dependence measure and a statistical test for conditional independence. The measure is based on the difference between analytic kernel embeddings of two well-suited distributions evaluated at a finite set of locations. We obtain its asymptotic distribution under the null hypothesis of conditional independence and design a consistent statistical test from it. We conduct a series of experiments showing that our new test outperforms state-of-the-art methods both in terms of type-I and type-II errors even in the high dimensional setting.

G.1 Introduction

We consider the problem of testing whether two variables X and Y are independent given a set of confounding variables Z, which can be formulated as a hypothesis testing problem of the form:

$$H_0 : \ X \perp Y|Z \qquad \text{vs.} \qquad H_1 : X \not\perp Y|Z.$$

Testing for conditional independence (CI) is central in a wide variety of statistical learning problems. For example, it is at the core of graphical modeling [Lauritzen, 1996, Koller and Friedman, 2009], causal discovery [Pearl, 2009, Glymour et al., 2019], variable selection [Candès et al., 2018], dimensionality reduction [Li, 2018], and biomedical studies [Richardson and Gilks, 1993, Dobra et al., 2004, Markowetz and Spang, 2007].

Testing for H_0 in such applications is known to be a highly challenging task [Shah and Peters, 2020, Neykov et al., 2021]. A large line of work has focused on the design of measures for conditional dependence based for example on kernel methods Fukumizu et al. [2008], Sheng and Sriperumbudur [2019], Park and Muandet [2020], Huang et al. [2020c] and rank statistics Azadkia and Chatterjee [2021], Shi et al. [2021b]. Testing for conditional independence is even a more difficult as it requires both designing a test statistic which measures the conditional dependencies and controlling its quantiles. Indeed, existing tests may fail to control the type-I error, especially when the confounding set of variables is high-dimensional with a complex dependency structure Bergsma [2004]. Furthermore, even if the test is valid, the availability of limited data makes the problem of discriminating between the null and alternative hypotheses extremely difficult, resulting in a test of low power. These challenges has motivated the development of a series of practical methods attempting to reliably test for conditional independence. These include tests based on kernels [Zhang et al., 2012, Doran et al., 2014, Strobl et al., 2019, Zhang et al., 2017], ranks Runge [2018], Mittag [2018], models [Sen et al., 2017, 2018, Chalupka et al., 2018, Shah and Peters, 2020], permutations and samplings [Berrett et al., 2020, Candès et al., 2018, Bellot and van der Schaar, 2019, Shi et al., 2021a, Javanmard and Mehrabi, 2021], and optimal transport Warren [2021].

In this paper, we propose a new kernel-based test for conditional independence with asymptotic theoretical guarantees. Taking inspiration from Chwialkowski et al. [2015], Jitkrittum et al. [2017], Scetbon and Varoquaux [2019b], we use the ℓ^p distance between two well-chosen analytic kernel mean embeddings evaluated at a finite set of locations. We show that this measure encodes the conditional dependence relation of the random variables under study. Under common assumptions on the richness of the RKHS, we derive the asymptotic null distribution of our measure, and design a simple nonparametric test that is distribution-free under the null hypothesis. Furthermore, we show that our test is consistent. Lastly, we validate our theoretical claims and study the performance of the proposed approach using simulated conditionally (in)dependent data and show that our testing procedure outperforms state-of-the-art methods.

G.1.1 Related Work

Zhang et al. [2012] propose a kernel based-test (KCIT), by leveraging the characterization of conditional independence derived in [Daudin, 1980] to form a test statistic. The authors of this work obtain the asymptotic null distribution of the proposed statistic and derived a practical procedure from it to test for H_0. However, one main practical issue of the proposed test is that the asymptotic null distribution of their statistic cannot be computed directly as it involved unknown quantities. To address this problem, the authors propose to approximate it either with Monte Carlo simulations or by fitting a Gamma distribution. In our work, we propose a new kernel-based statistic to test for conditional independence and show that its asymptotic null distribution is simply the standard normal distribution. In addition Zhang et al. [2012] extended the Gaussian process (GP) regression framework to the multi-output case, which allowed them to find the hyperparameters involved in the test statistic, maximizing the marginal likelihood. We also deploy a similar optimization procedure to that of Zhang et al. [2012], however, in our case the output of the GP regression is univariate and therefore more computationally efficient.

Other CI tests proposed in the literature suggest testing relaxed forms of conditional independence. For instance, Shah and Peters [2020] propose the generalised covariance measure (GCM) which only characterises weak conditional dependence Daudin [1980] and Zhang et al. [2017] propose a kernel-based test which focuses only on individual effects of the conditioning variable Z on X and Y. Some other tests are based on the knowledge of the conditional distributions in order to measure conditional dependencies. For example Candès et al. [2018] assume that one has access to the exact conditional distributions, Bellot and van der Schaar [2019], Shi et al. [2021a] approximate them using generative models and Sen et al. [2017] consider model-based methods to generate samples from the conditional distributions. In our work, we design a test statistic which characterizes the exact conditional independence of random variables and obtain its asymptotic null distribution without assuming any knowledge on the conditional distributions. Under some mild assumptions on the RKHSs considered, we also derive an approximate test statistic which admits the same asymptotic distribution and obtain a simple testing procedure from it.

G.2 Background and Notations

We first recall some notions on kernels and mean embeddings which will be useful in the derivation of our conditional independence test. Let $(\mathcal{D}, \mathcal{A})$ be a Borel measurable space and denote $\mathcal{M}_1^+(\mathcal{D})$ the space of Borel probability measures on $\mathcal{D}$. Let also (H, k) be a measurable RKHS on $\mathcal{D}$, i.e. a functional Hilbert space satisfying the reproducing property: for all $f \in H, x \in \mathcal{D}$, $f(x) = \langle f, k_x \rangle_H$. Let $\nu \in \mathcal{M}_1^+(\mathcal{D})$. If $\mathbb{E}_{x\sim\nu}[\sqrt{k(x,x)}]$ is finite, we define for all $t \in \mathcal{D}$ the *mean embedding* as $\mu_{\nu,k}(t) := \int_{x\in\mathcal{D}} k(x,t)d\nu(x)$. Note that $\mu_{\nu,k}$ is the unique element in H satisfying for all $f \in H, \mathbb{E}_{x\sim\nu}(f(x)) = \langle \mu_{\nu,k}, f\rangle_H$. If $\nu \mapsto \mu_{\nu,k}$ is injective, then the kernel k is said to be *characteristic*. This property is essential for the separation property to be verified when defining a kernel metric between distributions, such as the MMD [Gretton et al., 2012], or the ℓ^p distance [Scetbon and Varoquaux, 2019b].

ℓ^p-distance between mean embeddings. Let k be a definite positive, characteristic, continuous, and bounded kernel on $\mathbb{R}^d$ and $p \geq 1$ an integer. Scetbon and Varoquaux [2019b] showed that given an absolutely continuous Borel probability measure Γ on $\mathbb{R}^d$, the following function defined for any $(P, Q) \in \mathcal{M}_1^+(\mathbb{R}^d) \times \mathcal{M}_1^+(\mathbb{R}^d)$ as

$$d_p(P,Q) := \left[\int_{\mathbb{R}^d} |\mu_{P,k}(\mathbf{t}) - \mu_{Q,k}(\mathbf{t})|^p d\Gamma(\mathbf{t})\right]^{\frac{1}{p}} \tag{G.1}$$

is a metric on $\mathcal{M}_1^+(\mathbb{R}^d)$. When the kernel k is analytic[1], Scetbon and Varoquaux [2019b] also showed that for any $J \geq 1$,

$$d_{p,J}(P,Q) := \left[\frac{1}{J}\sum_{j=1}^{J} |\mu_{P,k}(\mathbf{t}_j) - \mu_{Q,k}(\mathbf{t}_j)|^p\right]^{\frac{1}{p}}, \tag{G.2}$$

where $(\mathbf{t}_j)_{j=1}^J$ are sampled independently from the Γ distribution, is a random metric[2] on $\mathcal{M}_1^+(\mathbb{R}^d)$.

In what follows, we consider distributions on Euclidean spaces. More precisely, let $d_x, d_y, d_z \geq 1$, $\mathcal{X} := \mathbb{R}^{d_x}$, $\mathcal{Y} := \mathbb{R}^{d_y}$, and $\mathcal{Z} := \mathbb{R}^{d_z}$. Let (X, Z, Y) be a random vector on $\mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$ with law P_{XZY}. We denote by P_{XY}, P_X, and P_Y the law of (X, Y), X, and Y, respectively. We also denote by $\ddot{\mathcal{X}} := \mathcal{X} \times \mathcal{Z}$, $\ddot{X} := (X, Z)$, and $P_{\ddot{X}}$ its law. Let $P_X \otimes P_Y$ be the product of the two measures P_X and P_Y. Given $(H_{\ddot{\mathcal{X}}}, k_{\ddot{\mathcal{X}}})$ and $(H_{\mathcal{Y}}, k_{\mathcal{Y}})$, two measurable reproducing kernel Hilbert spaces (RKHS) on $\ddot{\mathcal{X}}$ and $\mathcal{Y}$, respectively, we define the tensor-product RKHS $H = H_{\ddot{\mathcal{X}}} \otimes H_{\mathcal{Y}}$ associated with its *tensor-product kernel* $k = k_{\ddot{\mathcal{X}}} \otimes k_{\mathcal{Y}}$, defined for all $\ddot{x}, \ddot{x}' \in \ddot{\mathcal{X}}$ and $y, y' \in \mathcal{Y}$, as $k((\ddot{x}, y), (\ddot{x}', y')) = k_{\ddot{\mathcal{X}}}(\ddot{x}, \ddot{x}') \times k_{\mathcal{Y}}(y, y')$.

G.3 A new ℓ^p kernel-based testing procedure

In this section, we present our statistical procedure to test for conditional independence. We begin by introducing a general measure based on the ℓ^p distance d_p between mean embeddings which

[1] An *analytic kernel* on $\mathbb{R}^d$ is a positive definite kernel such that for all $x \in \mathbb{R}^d$, $k(x, \cdot)$ is an analytic function, i.e., a function defined locally by a convergent power series.

[2] A random metric is a random process which satisfies all the conditions for a metric almost-surely.

characterizes the conditional independence. We derive an oracle test statistic for which we obtain its asymptotic distribution under both the null and alternative hypothesis. Then, we provide an efficient procedure to effectively compute an approximation of our oracle statistic and show that it has the exact same asymptotic distribution. To avoid any bootstrap or permutation procedures, we offer a normalized version of our statistic and derive a simple and consistent test from it.

G.3.1 Conditional Independence Criterion

Let us first introduce the criterion we use to define our statistical test. We define a probability measure $P_{\ddot{X}\otimes Y|Z}$ on $\ddot{\mathcal{X}} \times \mathcal{Y}$ as

$$P_{\ddot{X}\otimes Y|Z}(A \times B) := \mathbb{E}_Z\Big[\mathbb{E}_{\ddot{X}|Z}[\mathbf{1}_A|Z]\mathbb{E}_{Y|Z}[\mathbf{1}_B|Z]\Big],$$

for any $(A, B) \in \mathcal{B}(\ddot{\mathcal{X}}) \times \mathcal{B}(\mathcal{Y})$, where $\mathbf{1}_A$ is the characteristic function of a measurable set A and similarly for B. One now characterize the independence of X and Y given Z as follows: $X \perp Y|Z$ if and only if $P_{XZY} = P_{\ddot{X}\otimes Y|Z}$ [Fukumizu et al., 2004, Theorem 8]. Therefore, we have a first simple characterization of the conditional independence: $X \perp Y|Z$ if and only if $d_p(P_{XZY}, P_{\ddot{X}\otimes Y|Z}) = 0$. With this in place, we now state some assumptions on the kernel k considered in the rest of this paper.

Assumption 4. *The kernel* $k : (\ddot{\mathcal{X}} \times \mathcal{Y}) \times (\ddot{\mathcal{X}} \times \mathcal{Y}) \to \mathbb{R}$ *is definite positive, characteristic, bounded, continuous and analytic. Moreover, the kernel* k *is a tensor product of kernels* $k_{\ddot{\mathcal{X}}}$ *and* $k_{\mathcal{Y}}$ *on* $\ddot{\mathcal{X}}$ *and* $\mathcal{Y}$*, respectively.*

It is worth noting that a sufficient condition for the kernel k to be characteristic, bounded, continuous and analytic, is that both kernels $k_{\ddot{\mathcal{X}}}$ and $k_{\mathcal{Y}}$ are characteristic, bounded, continuous and analytic [Szabó and Sriperumbudur, 2018]. For example, if the kernels $k_{\ddot{\mathcal{X}}}$ and $k_{\mathcal{Y}}$ are Gaussian kernels[3] on $\ddot{\mathcal{X}}$ and $\mathcal{Y}$ respectively, then $k = k_{\ddot{\mathcal{X}}} \otimes k_{\mathcal{Y}}$ satisfies Assumption 4 [Jitkrittum et al., 2017]. Using the analyticity of the kernel k, one can work with $d_{p,J}$ defined in (G.2) instead of d_p to characterize the conditional independence.

Proposition 41. *Let* $p \geq 1$, $J \geq 1$, k *be a kernel satisfying Assumption 4,* Γ *an absolutely continuous Borel probability measure on* $\ddot{\mathcal{X}} \times \mathcal{Y}$*, and* $\{(\mathbf{t}_j^{(1)}, t_j^{(2)})\}_{j=1}^J$ *sampled independently from* Γ*. Then* Γ*-almost surely,* $d_{p,J}(P_{XZY}, P_{\ddot{X}\otimes Y|Z}) = 0$ *if and only if* $X \perp Y|Z$*.*

Proof. Recall that $X \perp Y|Z$ if and only if $P_{XZY} = P_{\ddot{X}\otimes Y|Z}$ [Fukumizu et al., 2008]. If k is bounded, characteristic, and analytic, then, by invoking [Scetbon and Varoquaux, 2019b, Theorem 4] we get that $d_{p,J}^p$ is a random metric on the space of Borel probability measures. This concludes the proof. □

The key advantage of using $d_{p,J}(P_{XZY}, P_{\ddot{X}\otimes Y|Z})$ to measure the conditional dependence is that it only requires to compute the differences between the mean embeddings of P_{XZY} and $P_{\ddot{X}\otimes Y|Z}$

[3] A gaussian kernel K on $\mathcal{W} \subset \mathbb{R}^d$ satisfies for all $w, w' \in \mathcal{W}$, $K(w, w') := \exp\Big(\frac{\|w-w'\|_2^2}{2\sigma^2}\Big)$.

at J locations. In what follows, we derive from it a first oracle test statistic for conditional independence.

G.3.2 A First Oracle Test Statistic

When the kernel k considered satisfies Assumption 4, we can obtain a simple expression of our measure $d_{p,J}(P_{XZY}, P_{\ddot{X}\otimes Y|Z})$. Indeed, the tensor formulation of the kernel k allows us to write the mean embedding of $P_{\ddot{X}\otimes Y|Z}$ for any $(\mathbf{t}^{(1)}, t^{(2)}) \in \ddot{\mathcal{X}} \times \mathcal{Y}$ as:

$$\begin{aligned}&\mu_{P_{\ddot{X}\otimes Y|Z}, k_{\ddot{\mathcal{X}}}\cdot k_{\mathcal{Y}}}(\mathbf{t}^{(1)}, t^{(2)}) = \\ &\mathbb{E}_Z\Big[\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{X})|Z\Big]\mathbb{E}_Y\Big[k_{\mathcal{Y}}(t^{(2)}, Y)|Z\Big]\Big] \,.\end{aligned} \tag{G.3}$$

Then, by defining the witness function as

$$\begin{aligned}\Delta(\mathbf{t}^{(1)}, t^{(2)}) := \mathbb{E}\Big[&\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{X}) - \mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{X})|Z\Big]\Big) \\ &\times \Big(k_{\mathcal{Y}}(t^{(2)}, Y) - \mathbb{E}_Y\Big[k_{\mathcal{Y}}(t^{(2)}, Y)|Z\Big]\Big)\Big],\end{aligned}$$

and by considering $\{(\mathbf{t}_j^{(1)}, t_j^{(2)})\}_{j=1}^J$ sampled independently according to Γ, we get that (see Appendix G.6.1 for more details)

$$d_{p,J}(P_{XZY}, P_{\ddot{X}\otimes Y|Z}) = \left[\frac{1}{J}\sum_{j=1}^{J}\Big|\Delta(\mathbf{t}_j^{(1)}, t_j^{(2)})\Big|^p\right]^{1/p}.$$

Estimation. Given n observations $\{(x_i, z_i, y_i)\}_{i=1}^n$ that are drawn independently from P_{XZY}, we aim at obtaining an estimator of $d_{p,J}^p(P_{XZY}, P_{\ddot{X}\otimes Y|Z})$. To do so, we introduce the following estimate of $\Delta(\mathbf{t}^{(1)}, t^{(2)})$, defined as

$$\begin{aligned}\Delta_n(\mathbf{t}^{(1)}, t^{(2)}) := \frac{1}{n}\sum_{i=1}^{n}&\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{x}_i) - \mathbb{E}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{X})|z_i\Big]\Big) \\ &\times\Big(k_{\mathcal{Y}}(t^{(2)}, y_i) - \mathbb{E}\Big[k_{\mathcal{Y}}(t^{(2)}, Y)|z_i\Big]\Big).\end{aligned}$$

With this in place, a natural candidate to estimate $d_{p,J}^p(P_{XZY}, P_{\ddot{X}\otimes Y|Z})$ (up to the constant J) can be expressed as

$$\mathrm{CI}_{n,p} := \sum_{j=1}^{J}\Big|\Delta_n(\mathbf{t}_j^{(1)}, t_j^{(2)})\Big|^p,$$

where $(\mathbf{t}_1^{(1)}, t_1^{(2)}), \ldots, (\mathbf{t}_J^{(1)}, t_J^{(2)}) \in \ddot{\mathcal{X}} \times \mathcal{Y}$ are sampled independently from Γ.

We now turn to derive the asymptotic distribution of this statistic. For that purpose, define, for all $j \in \{1, \ldots, J\}$ and $i \in \{1, \ldots, n\}$,

$$\begin{aligned} u_i(j) := &\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{x}_i) - \mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X})|Z = z_i\Big]\Big) \\ &\times \Big(k_{\mathcal{Y}}(t_j^{(2)}, y_i) - \mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)}, Y)|Z = z_i\Big]\Big), \end{aligned}$$

$\mathbf{u}_i := (u_i(1), \ldots, u_i(J))^T$ and $\boldsymbol{\Sigma} := \mathbb{E}(\mathbf{u}_1 \mathbf{u}_1^T)$. We also denote by $\mathbf{S}_n := \frac{1}{n}\sum_{i=1}^n \mathbf{u}_i$. Observe that $\mathrm{CI}_{n,p} = \|\mathbf{S}_n\|_p^p$. In the following proposition we obtain the asymptotic distribution of our statistic $\mathrm{CI}_{n,p}$.

Proposition 42. *Suppose that Assumption 4 is verified. Let $p \geq 1$, $J \geq 1$ and $((\mathbf{t}_1^{(1)}, t_1^{(2)}), \ldots, (\mathbf{t}_J^{(1)}, t_J^{(2)})) \in (\ddot{\mathcal{X}} \times \mathcal{Y})$. Then, under H_0, we have: $\sqrt{n}\mathbf{S}_n \to \mathcal{N}(0, \boldsymbol{\Sigma})$. Moreover, under H_1, if $((\mathbf{t}_j^{(1)}, t_j^{(2)}))_{j=1}^J$ are sampled independently according to Γ, then Γ-almost surely, for any $q \in \mathbb{R}$, $\lim_{n\to\infty} P(n^{p/2}\mathrm{CI}_{n,p} \geq q) = 1$.*

Proof. Recall that $\mathbf{S}_n = \frac{1}{n}\sum_{i=1}^n \mathbf{u}_i$ where $\mathbf{u}_i$ are i.i.d. samples. Under H_0, $\mathbb{E}[\mathbf{u}_i] = 0$. Using the Central Limit Theorem, we get: $\sqrt{n}\mathbf{S}_n \to \mathcal{N}(0, \boldsymbol{\Sigma})$. Using the analyticity of the kernel k, under H_1, Γ-almost surely, there exists a $j \in \{1, \ldots, J\}$ such that $\mathbb{E}[u_1(j)] \neq 0$. Therefore, we can deduce that Γ-almost surely, $\mathbf{S} := \mathbb{E}[\mathbf{u_1}] \neq 0$. Now, for all $q > 0$, we get: $P(n^{p/2}\mathrm{CI}_{n,p} > q) \to 1$ because $\mathrm{CI}_{n,p} \to \|\mathbf{S}\|_p^p$ when $n \to \infty$. □

From the above proposition, we can define a consistent statistical test at level $0 < \alpha < 1$, by rejecting the null hypothesis if $n^{p/2}\mathrm{CI}_{n,p}$ is larger than the $(1-\alpha)$ quantile of the asymptotic null distribution, which is the law associated with $\|X\|_p^p$, where X follows the multivariate normal distribution $\mathcal{N}(0, \boldsymbol{\Sigma})$. However, in practice, $\mathrm{CI}_{n,p}$ cannot be computed as it requires the access to samples from the conditional means involved in the statistic, namely $\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X})|Z\Big]$ and $\mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)}, Y)|Z\Big]$ for all $j \in \{1, \ldots, J\}$, which are unknown. Below, we show how to estimate these conditional means by using Regularized Least-Squares (RLS) estimators.

G.3.3 Approximation of the Test Statistic

Our goal here is to estimate $\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X})|Z = \cdot\Big]$ and $\mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)}, Y)|Z = \cdot\Big]$ for all $j \in \{1, \ldots, J\}$ in order to effectively approximate of our statistic. To do so, we consider kernel-based regularized least squares (RLS) estimators. Let $1 \leq r \leq n$ and $\{(x_i, z_i, y_i)\}_{i=1}^r$ be a subset of r samples. Let also $j \in \{1, \ldots, J\}$, and denote by $H_{\mathcal{Z}}^{1,j}$ and $H_{\mathcal{Z}}^{2,j}$ two separable RKHSs on $\mathcal{Z}$. Denote also by $k_{\mathcal{Z}}^{1,j}$ and $k_{\mathcal{Z}}^{2,j}$ their associated kernels and $\lambda_{j,r}^{(1)}, \lambda_{j,r}^{(2)} > 0$ the regularization parameters involved in the RLS regressions. Then, the RLS estimators are the unique solutions of the following problems:

$$\min_{h \in H_{\mathcal{Z}}^{2,j}} \frac{1}{r}\sum_{i=1}^r \Big(h(z_i) - k_{\mathcal{Y}}(t_j^{(2)}, y_i)\Big)^2 + \lambda_{j,r}^{(2)}\|h\|_{H_{\mathcal{Z}}^{2,j}}^2 \text{ and}$$

$$\min_{h \in H_{\mathcal{Z}}^{1,j}} \frac{1}{r} \sum_{i=1}^{r} \Big(h(z_i) - k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, (x_i, z_i)) \Big)^2 + \lambda_{j,r}^{(1)} \|h\|_{H_{\mathcal{Z}}^{1,j}}^2 ,$$

which we denote by $h_{j,r}^{(2)}$ and $h_{j,r}^{(1)}$, respectively. These estimators have simple expressions in term of the kernels involved. For example, let $k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X}_r) := [k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, (x_1, z_1)), \dots, k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, (x_r, z_r))]^T$, then for any $z \in \mathcal{Z}$, the estimator $h_{j,r}^{(1)}$ can be expressed as

$$\begin{aligned} h_{j,r}^{(1)}(z) &= \sum_{i=1}^{r} [\alpha_{j,r}^{(1)}]_i k_{\mathcal{Z}}^{1,j}(z_i, z) \ , \ \text{ with} \\ \alpha_{j,r}^{(1)} &:= (\mathbf{K}_{r,\mathcal{Z}}^{1,j} + r\lambda_{j,r}^{(1)} \mathrm{Id}_r)^{-1} k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X}_r) \in \mathbb{R}^r, \end{aligned}$$

where $\mathbf{K}_{r,\mathcal{Z}}^{1,j} := (k_{\mathcal{Z}}^{1,j}(z_i, z_j))_{1 \le i,j \le r}$. Similarly, we obtain simple expressions of $h_{j,r}^{(2)}$. We can now introduce our new estimator of the witness function at each location $(\mathbf{t}_j^{(1)}, t_j^{(2)})$ as follows:

$$\begin{aligned} \widetilde{\Delta}_{n,r}(\mathbf{t}_j^{(1)}, t_j^{(2)}) := \frac{1}{n} \sum_{i=1}^{n} &\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{x}_i) - h_{j,r}^{(1)}(z_i) \Big) \\ &\times \Big(k_{\mathcal{Y}}(t_j^{(2)}, y_i) - h_{j,r}^{(2)}(z_i) \Big), \end{aligned}$$

and the proposed test statistic becomes

$$\widetilde{\mathrm{CI}}_{n,r,p} := \sum_{j=1}^{J} \Big| \widetilde{\Delta}_{n,r}(\mathbf{t}_j^{(1)}, t_j^{(2)}) \Big|^p .$$

Asymptotic Distribution. To get the asymptotic distribution, we need to make two extra assumptions. Let us define, for $m \in \{1, 2\}$ and $j \in \{1, \dots, J\}$, $L_Z^{m,j}$—the operator on $L^2(\mathcal{Z}, P_Z)$ as $L_Z^{m,j}(g)(\cdot) = \int_{\ddot{\mathcal{X}}} k_{\mathcal{Z}}^{m,j}(\cdot, z) g(z) dP_Z(z)$.

Assumption 5. *There exists $Q > 0$, and $\gamma \in [0, 1]$ such that for all $\lambda > 0$, $m \in \{1, 2\}$ and $j \in \{1, \dots, J\}$:*

$$Tr((L_Z^{m,j} + \lambda I)^{-1} L_Z^{m,j}) \le Q \lambda^{-\gamma}.$$

Assumption 6. *There exists $2 \ge \beta > 1$ such that for any $j \in \{1, \dots, J\}$, $(\mathbf{t}^{(1)}, t^{(2)}) \in \ddot{\mathcal{X}} \times \mathcal{Y}$,*

$$\begin{aligned} \mathbb{E}_{\ddot{X}} \Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{X}) | Z = \cdot \Big] &\in \mathcal{R}\bigg(\Big[L_Z^{1,j} \Big]^{\beta/2} \bigg), \\ \mathbb{E}_{Y} \Big[k_{\mathcal{Y}}(t^{(2)}, Y) | Z = \cdot \Big] &\in \mathcal{R}\bigg(\Big[L_Z^{2,j} \Big]^{\beta/2} \bigg), \end{aligned}$$

where $\mathcal{R}\left(\left[L_Z^{m,j}\right]^{\beta/2}\right)$ *is the image space of* $\left[L_Z^{m,j}\right]^{\beta/2}$. *Moreover, there exists* $L, \sigma > 0$ *such that for all* $l \geq 2$ *and* P_Z*-almost all* $z \in \mathcal{Z}$

$$\mathbb{E}_{\ddot{X}}\left[\left|k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{X}) - \mathbb{E}_Y\left[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{X})\right]\right|^l\right] \leq \frac{l!\sigma^2 L^{l-2}}{2},$$
$$\mathbb{E}_{|Z=z}\left[\left|k_{\mathcal{Y}}(t^{(2)}, Y) - \mathbb{E}_{Y|Z=z}\left[k_{\mathcal{Y}}(t^{(2)}, Y)\right]\right|^l\right] \leq \frac{l!\sigma^2 L^{l-2}}{2}.$$

These assumptions are central in our proofs and are common in kernel statistic studies [Caponnetto and De Vito, 2007, Fischer and Steinwart, 2020, Rudi and Rosasco, 2017]. Under these assumptions, Fischer and Steinwart [2020] proved optimal learning rates for RLS in RKHS norm, which is essential to guarantee that our new statistic $\widetilde{\mathrm{CI}}_{n,r,p}$, estimated with RLS, has the same asymptotic law as our oracle estimator $\mathrm{CI}_{n,p}$.

To derive the asymptotic distribution of our new test statistic, we also need to define for all $j \in \{1, \ldots, J\}$ and $i \in \{1, \ldots, n\}$, $\widetilde{u}_{i,r}(j) := (k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{x}_i) - h_{j,r}^{(1)}(z_i))(k_{\mathcal{Y}}(t_j^{(2)}, y_i) - h_{j,r}^{(2)}(z_i))$, $\widetilde{\mathbf{u}}_{i,r} := (\widetilde{u}_{i,r}(1), \ldots, \widetilde{u}_{i,r}(J))^T$, and $\widetilde{\mathbf{S}}_{n,r} := \frac{1}{n}\sum_{i=1}^n \widetilde{\mathbf{u}}_{i,r}$. Note that $\widetilde{\mathrm{CI}}_{n,r,p} = \|\widetilde{\mathbf{S}}_{n,r}\|_p^p$. In the following proposition, we show the asymptotic behavior of the statistic of interest. The proof of this proposition is given in Appendix G.6.2.

Proposition 43. *Suppose that Assumptions 4-5-6 are verified. Let* $p \geq 1$, $J \geq 1$, $((\mathbf{t}_1^{(1)}, t_1^{(2)}), \ldots, (\mathbf{t}_J^{(1)}, t_J^{(2)})) \in (\ddot{\mathcal{X}} \times \mathcal{Y})^J$, r_n *such that* $n^{\frac{\beta+\gamma}{2\beta}} \in o(r_n)$ *and* $\lambda_{r_n} = r_n^{-\frac{1}{1+\gamma}}$. *Then, under* H_0, *we have* $\sqrt{n}\widetilde{\mathbf{S}}_{n,r_n} \to \mathcal{N}(0, \boldsymbol{\Sigma})$. *Moreover, under* H_1, *if the* $((\mathbf{t}_j^{(1)}, t_j^{(2)}))_{j=1}^J$ *are sampled independently according to* Γ, *then* Γ*-almost surely, for any* $q \in \mathbb{R}$, $\lim_{n\to\infty} P(n^{p/2}\widetilde{\mathrm{CI}}_{n,r_n,p} \geq q) = 1$.

From the above proposition, we can derive a consistent test at level α for $0 < \alpha < 1$. Indeed, we obtain the asymptotic null distribution of $n^{p/2}\widetilde{\mathrm{CI}}_{n,r_n,p}$ and we show that under the alternative hypothesis H_1, Γ-almost surely, $n^{p/2}\widetilde{\mathrm{CI}}_{n,r_n,p}$ is arbitrarily large as n goes to infinity. For a fixed level α, the test rejects H_0 if $n^{p/2}\widetilde{\mathrm{CI}}_{n,r_n,p}$ exceeds the $(1-\alpha)$-quantile of its asymptotic null distribution and this test is therefore consistent. For example, when $p \in \{1, 2\}$, the asymptotic null distribution of $n^{p/2}\widetilde{\mathrm{CI}}_{n,r_n,p}$ is either a sum of correlated Nakagami variables[4] ($p = 1$) or a sum of correlated chi square variables ($p = 2$). However, computing the quantiles of these asymptotic null distributions can be computationally expensive as it requires a bootstrap or permutation procedure. In the following, we consider a different approach in which we normalize the statistic to obtain a simple asymptotic null distribution.

[4] the probability density function of a Nakagami distribution of parameters $m \geq \frac{1}{2}$ and $\omega > 0$ is for all $x \geq 0$, $f(x, m, \omega) = \frac{2m^m}{G(m)\omega^m} x^{2m-1} \exp\left(\frac{-m}{\omega}x^2\right)$ where G is the Euler Gamma function.

G.3.4 Normalization of the Test Statistic

Herein, we consider a normalized variant of our statistic $\widetilde{\mathrm{CI}}_{n,r,p}$ in order to obtain a tractable asymptotic null distribution. Denote $\boldsymbol{\Sigma}_{n,r} := \frac{1}{n}\sum_{i=1}^{n}\widetilde{\boldsymbol{u}}_{i,r}\widetilde{\boldsymbol{u}}_{i,r}^{T}$ and let $\delta_n > 0$, then the normalized statistic considered is given by

$$\widetilde{\mathrm{NCI}}_{n,r,p} := \|(\boldsymbol{\Sigma}_{n,r} + \delta_n \mathrm{Id}_J)^{-1/2}\widetilde{\mathbf{S}}_{n,r}\|_p^p.$$

In the next proposition, we show that our normalized approximate statistic converges in law to the standard multivariate normal distribution. The proof is given in Appendix G.6.3.

Proposition 44. *Suppose that Assumptions 4-5-6 are verified. Let $p \geq 1$, $J \geq 1$, $((\mathbf{t}_1^{(1)}, t_1^{(2)}), \ldots, (\mathbf{t}_J^{(1)}, t_J^{(2)})) \in (\ddot{\mathcal{X}} \times \mathcal{Y})^J$, r_n such that $n^{\frac{\beta+\gamma}{2\beta}} \in o(r_n)$, $\lambda_n = r_n^{-\frac{1}{1+\gamma}}$ and $(\delta_n)_{n\geq 0}$ a sequence of positive real numbers such that $\lim_{n\to\infty}\delta_n = 0$. Then, under H_0, we have $\sqrt{n}(\boldsymbol{\Sigma}_{n,r} + \delta_n Id_J)^{-1/2}\mathbf{S}_{n,r_n} \to \mathcal{N}(0, Id_J)$. Moreover, under H_1, if the $((\mathbf{t}_j^{(1)}, t_j^{(2)}))_{j=1}^{J}$ are sampled independently according to Γ, then Γ-almost surely, for any $q \in \mathbb{R}$, $\lim_{n\to\infty} P(n^{p/2}\widetilde{\mathrm{NCI}}_{n,r_n,p} \geq q) = 1$.*

Remark 16. *We emphasize that J need not increase with n for test consistency. Note also that the regularization parameter δ_n allows to ensure that $(\boldsymbol{\Sigma}_{n,r} + \delta_n Id_J)^{-1/2}$ can be stably computed. In practice, δ_n requires no tuning, and can be set to be a very small constant.*

Our normalization procedure allows us to derive a simple statistical test, which is distribution-free under the null hypothesis.

Statistical test at level α: Compute $n^{p/2}\widetilde{\mathrm{NCI}}_{n,r,p}$, choose the threshold τ corresponding to the $(1-\alpha)$ quantile of the asymptotic null distribution, and reject the null hypothesis whenever $n^{p/2}\widetilde{\mathrm{NCI}}_{n,r,p}$ is larger than τ. For example, if $p = 2$, the threshold τ is the $(1-\alpha)$-quantile of $\chi^2(J)$, i.e., a sum of J *independent* standard χ^2 variables.

Total Complexity: Our normalized statistic $\widetilde{\mathrm{NCI}}_{n,r,p}$ requires first to compute $\alpha_{j,r}^{(1)}$ and $\alpha_{j,r}^{(2)}$. These quantities can be evaluated in at most $\mathcal{O}(r^2 d + r^3)$ algebraic operations where d corresponds to the computational cost of evaluating the kernels involved in the RLS regressions. We will use the above for the complexity analysis of our method, although one can apply the Coppersmith–Winograd algorithm [Coppersmith and Winograd, 1987] that reduces the computational cost to $\mathcal{O}(r^2 d + r^{2.376})$. Once $\alpha_{j,r}^{(1)}$ and $\alpha_{j,r}^{(2)}$ are available, evaluating the RLS estimators $h_{j,r}^{(1)}$ and $h_{j,r}^{(2)}$ require only $\mathcal{O}(rd)$ operations. Then $\widetilde{\Delta}_{n,r}$ can be evaluated in $\mathcal{O}(nrd + r^2 d + r^3)$ operations and $\widetilde{\mathrm{CI}}_{n,r,p}$ has therefore a computational complexity of $\mathcal{O}(J(nrd + r^2 d + r^3))$. The computation of $\widetilde{\mathrm{NCI}}_{n,r,p}$ requires inverting a $J \times J$ matrix $\boldsymbol{\Sigma}_{n,r} + \delta_n \mathrm{Id}_J$, but this is fast and numerically stable: we empirically observe that only a small value of J is required (see Section G.4), e.g. less than 10. Finally the total computational cost to evaluate $\widetilde{\mathrm{NCI}}_{n,r,p}$ is $\mathcal{O}(J(nrd + r^2 d + r^3) + nJ^2 + J^3)$.

G.3.5 Hyperparameters

The hyperparameters of our statistics $\widetilde{\mathrm{NCI}}_{n,r,p}$ fall into two categories: those directly involved with the test and those of the regression. We assume from now on that all the kernels involved in the computation of our statistics are *Gaussian kernels*, and consider n i.i.d. observations $\{(x_i, z_i, y_i)\}_{i=1}^n$.

The first category includes both the choice of the locations $((t_x, t_z)_j, (t_y)_j))_{j=1}^J$ on which differences between the mean embeddings are computed and the choice of the kernels $k_{\ddot{\mathcal{X}}}$ and $k_{\mathcal{Y}}$. Each location t_x, t_y, t_z is randomly chosen according to a Gaussian variable with mean and covariance of $\{x_i\}_{i=1}^n$, $\{y_i\}_{i=1}^n$, and $\{z_i\}_{i=1}^n$, respectively. As we consider Gaussian kernels, we should also choose the bandwidths. Here, we restrict ourselves to one-dimensional kernel bandwidths $\sigma_{\mathcal{X}}$, $\sigma_{\mathcal{Y}}$, and $\sigma_{\mathcal{Z}}$ for the kernels $k_{\mathcal{X}}$, $k_{\mathcal{Y}}$, and $k_{\mathcal{Z}}$, respectively. More precisely, we select the median of $\{\|x_i - x_j\|\}_{i,j=1}^n$, $\{\|y_i - y_j\|\}_{i,j=1}^n$, and $\{\|z_i - z_j\|\}_{i,j=1}^n$ for $\sigma_{\mathcal{X}}$, $\sigma_{\mathcal{Y}}$, and $\sigma_{\mathcal{Z}}$, respectively.

The other category contains all the kernels $k^{m,j}$ and the regularization parameters $\lambda_{j,r}^{(m)}$ involved in the RLS problems. These parameters should be selected carefully to avoid either underfitting of the regressions, which may increase the type-I error, or overfitting, which may result in a large type-II error. To optimize these, similarly to Zhang et al. [2012], we consider a GP regression that maximizes the likelihood of the observations. While carrying out a precise GP regression can be prohibitive, in practice, we run this method only on a batch of size 200 observations randomly selected and we perform only 10 iterations for choosing the hyperparameters involved in the RLS problems. Hence, our optimization procedure does not affect the total computational cost as it is independent of the number of observations n.

G.4 Experiments

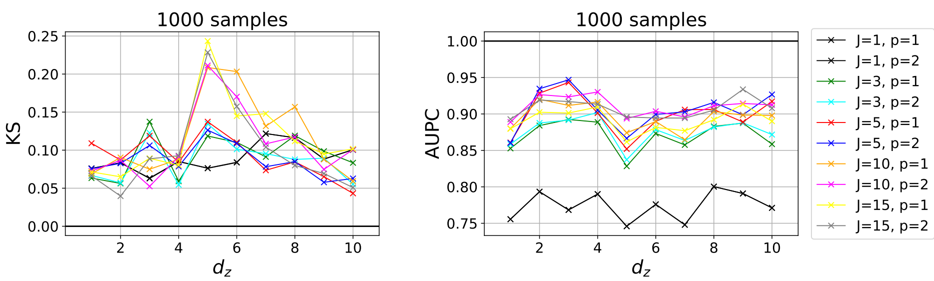


Figure G.1: Comparison of the KS statistic (*left*) and the AUPC (*right*) of our test statistic $\widetilde{\mathrm{NCI}}_{n,r,p}$ when the data is generated respectively from the models defined in (G.4) and (G.5) with Gaussian noises for multiple p and J. For each problem, we draw $n = 1000$ samples and repeat the experiment 100 times. We set $r = 1000$ and report the results obtained when varying the dimension d_z of each problem from 1 to 10. Observe that when $J = 1$, for all $p \geq 1$ $\widetilde{\mathrm{NCI}}_{n,r,1} = \widetilde{\mathrm{NCI}}_{n,r,p}$, therefore there is only one common black curve.

The goal of this section is three fold: (i) to investigate the effects of the parameters J and p on the performances of our method, (ii) to validate our theoretical results depicted in Propositions 42 and 44, and (iii) to compare our method with those proposed in the literature. In more

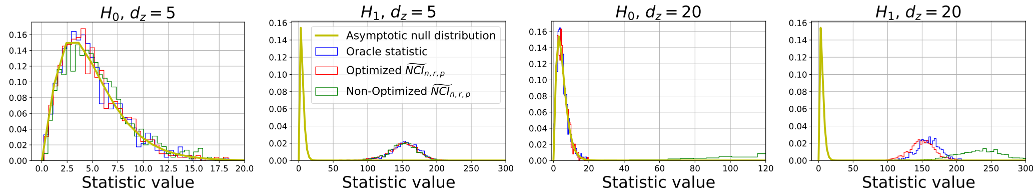


Figure G.2: Comparisons between the empirical distributions of the normalized version of the oracle statistic $\widehat{\mathrm{CI}}_{n,p}$ and the approximate normalized statistic $\widetilde{\mathrm{NCI}}_{n,r,p}$, with the theoretical asymptotic null distribution when the data is generated either from the model defined in (G.6) (*left*) or the one defined in (G.7) (*right*). We set the dimension of Z to be either $d_z = 5$ (*top row*) or $d_z = 20$ (*bottom row*). For each problem, we draw $n = 1000$ samples and repeat the experiment 1000 times. In all the experiments, we set $J = 5$ and $p = 2$, thus the asymptotic null distribution follows a $\chi^2(5)$. Observe that both the oracle statistic and the approximated one recover the true asymptotic distribution under the null hypothesis. When H_1 holds, we can see that the two statistics manage to reject the null hypothesis. This figure also illustrates the empirical distribution of our approximate statistic when we do not optimize the hyperparameters involved in the RLS estimators: in this case we do not control the type-I error in the high dimensional setting.

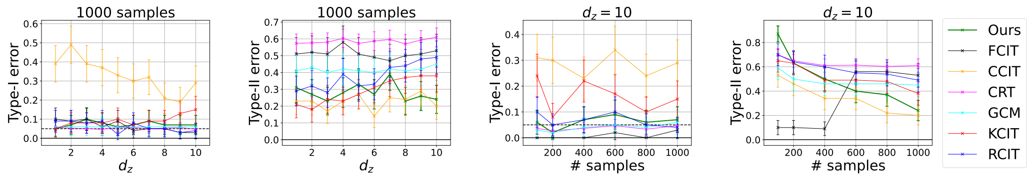


Figure G.3: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line) and the type-II error (lower is better) of our test procedure with other SoTA tests on the two problems presented in (G.4) and (G.5) with Gaussian noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): type-I and type-II errors obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): type-I and type-II errors obtained by each test when varying the number of samples n from 100 to 1000; here, the dimension d_z is fixed and equals to 10.

detail, we first compare the performance of our method, both in terms of both power and type-I error, by varying the hyperparameters J and p. We show that our method is robust to the choice of p, and also show that the power increases as J increases. Then, we explore synthetic toy problems where one can derive an explicit formulation of the conditional means involved in our test statistic. In these cases, we can compute our proposed oracle statistic $\widehat{\mathrm{CI}}_{n,p}$ and its normalized version, allowing us to show that under the null hypothesis we recover the theoretical asymptotic null distribution obtained in Proposition 42. We also reach to similar conclusions regarding our approximate normalized test statistic, $\widetilde{\mathrm{NCI}}_{n,r,p}$. In addition, in this experiment, we investigate the effect of the proposed optimization procedure for choosing the hyperparameters involved in the RLS estimators of $\widetilde{\mathrm{NCI}}_{n,r,p}$, and show its benefits. Finally, we demonstrate on several synthetic experiments that our proposed testing procedure outperforms state-of-the-art (SoTA) methods both in terms of statistical power and type-I error, even in the high dimensional setting.

Benchmarks. We consider 6 synthetic data sets and compare the power and type-I error of our test $\widetilde{\mathrm{NCI}}_{n,r,p}$ to the following 6 existing CI methods: **KCIT** [Zhang et al., 2012], **RCIT** [Strobl et al., 2019], **CCIT** [Sen et al., 2017], **CRT** [Candès et al., 2018] using correlation statistic from [Bellot and van der Schaar, 2019], **FCIT** [Chalupka et al., 2018] and **GCM** [Shah and Peters, 2020]. Software packages of all the above tests are freely available online and each experiment was run on a single CPU.

Evaluation. To evaluate the performance of the tests, we consider four metrics. Under H_0, we report either the Kolmogorov-Smirnov (KS) test statistic between the distribution of p-values returned by the tests and the uniform distribution on $[0, 1]$, or the type-I errors at level $\alpha = 0.05$. Note that a valid conditional independence test should control the type-I error rate at any level α. Here, a test that generates a p-value that follows the uniform distribution over $[0, 1]$ will achieve this requirement. The latter property of the p-values translates to a small KS statistic value. Under H_1, we compute either the area under the power curve (AUPC) of the empirical cumulative density function of the p-values returned by the tests, or the resulting type-II error. A conditional test has higher power when its AUPC is closer to one. Alternatively, the smaller the type-II error is, the more powerful the test is.

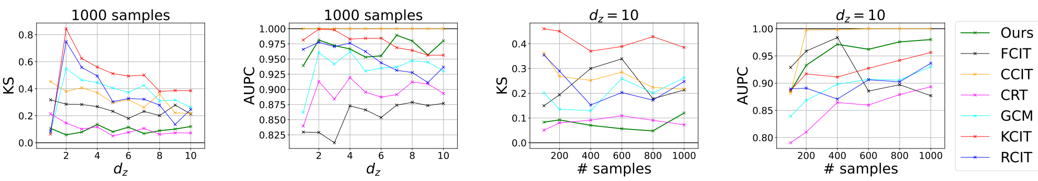


Figure G.4: Comparison of the KS statistic and the AUPC of our testing procedure with other SoTA tests on the two problems presented in Eq. (G.8) and Eq. (G.9) with Laplace noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): the KS statistic and AUPC (respectively) obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): the KS and AUPC (respectively), obtained by each test when varying the number of samples n from 100 to 1000; here, the dimension d_z is fixed and equals to 10.

Effects of p, J and r. Our first experiment studies the effects of p and J on our proposed method. In addition we investigate the sensitivity of the method when varying the rank regression r both in term of performance and time. To do so, we follow the synthetic experiment proposed in Strobl et al. [2019]. To evaluate the type-I error, we generate data that follows the model:

$$X = f_1(\varepsilon_x),\ Y = f_2(\varepsilon_y),\ \text{and Z} \sim \mathcal{N}(0_d, I_{d_z}), \tag{G.4}$$

where Z, ε_x, and ε_y are samples from jointly independent standard Gaussian or Laplace distributions, and f_1 and f_2 are smooth functions chosen uniformly from the set $\{(\cdot), (\cdot)^2, (\cdot)^3, \tanh(\cdot), \exp(-|\cdot|)\}$. To compare the power of the tests, we also consider the model:

$$X = f_1(\varepsilon_x + 0.8\varepsilon_b), Y = f_2(\varepsilon_y + 0.8\varepsilon_b), \tag{G.5}$$

where ε_b is sampled from a standard Gaussian or Laplace distribution. In Figure G.1, we compare the KS statistic and the AUPC of our method when varying p and J. That figure shows that (i) our method is robust to the choice of p, and (ii) the performances of the test do not necessarily increase as J increases. In Figure G.5 (see Appendix G.7.2), we also show that the power of the test is not very sensible to the choice of the rank r, however, we observe that the type-I error decreases as the rank r increases. Armed with theses observations, in the following experiments, we always set $p = 2$, $J = 5$ and $r = n$ for our method.

Illustrations of our theoretical findings. The following experiment confirms that validity of our theoretical results from Propositions 42 and 44. For that purpose, we generate two synthetic data sets for which either H_0 or H_1 holds. Concretely, we define a first triplet (X, Y, Z) as follows:

$$X = \mathrm{P}_1(Z) + \varepsilon_x, \;\; Y = \mathrm{P}_1(Z) + \varepsilon_y. \tag{G.6}$$

Above, ε_x and ε_y follow two independent standard normal distributions, $Z \sim \mathcal{N}(0_{d_z}, \Sigma)$ with $\Sigma \in \mathbb{R}^{d_z \times d_z}$. The covariance matrix Σ is obtained by multiplying product of a random matrix whose entries are independent and follow standard normal distribution, by its transpose, and P_1 is a projection onto the first coordinate. As a result, in this case, we have that $X \perp Y \mid Z$. We also consider a modification of the above data generating function for which H_1 holds. This is done by adding a noise component ε_b that is shared across X and Y as follows:

$$X = \mathrm{P}_1(Z) + \varepsilon_x + \varepsilon_b, \;\; Y = \mathrm{P}_1(Z) + \varepsilon_y + \varepsilon_b, \tag{G.7}$$

where ε_b follows the standard normal distribution. Since we consider *Gaussian kernels*, we can obtain an explicit formulation of $\mathbb{E}_{\ddot{X}}\left[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X})|Z = \cdot\right]$ and $\mathbb{E}_Y\left[k_{\mathcal{Y}}(t_j^{(2)}, Y)|Z = \cdot\right]$ for both data generation functions. See Appendix G.7.1 for more details. Consequently, we are able to compute both the normalized version of our oracle statistic $\widehat{\mathrm{CI}}_{n,p}$ and our approximate normalized statistic $\widetilde{\mathrm{NCI}}_{n,r,p}$. In Figure G.2, we show that both statistics manage to recover the asymptotic distribution under H_0, and reject the null hypothesis under H_1. In addition, we show that in the high dimensional setting, only our optimized version of $\widetilde{\mathrm{NCI}}_{n,r,p}$—obtained by optimizing the hyperparameters involved in the RLS estimators of our statistic—manages to recover the asymptotic distribution under H_0.

Comparisons with existing tests. In our next experiments, we compare the performance of our method (implemented with the optimized version of our statistic) with state-of-the-art techniques for conditional independence testing. We first study the two data generating functions from (G.4) and (G.5). For each of these problems, we consider two settings. In the first, we fix the dimension d_z while varying the number of samples n. In the second, we fix the number of samples while varying the dimension of the problem. To evaluate the performance of the tests, we compare the type-I errors at level $\alpha = 0.05$ under the first model (G.4), and, for second model (G.5), we evaluate the power of the test by presenting the type-II error. Figures G.3 (Gaussian case) and G.9 (Laplace case) demonstrate that our method consistently controls the type-I error and obtains a power similar to the best SoTA tests. In Figures G.7 and G.10, we also compare the KS statistic and the AUPC of the different tests, and obtain similar conclusions. In addition, we investigate the high dimensional regime and show in Figure G.8 and G.11 that our test is the only one which manages to control the type-I error while being competitive in term of power with other methods. See Appendix G.7.3 for more details.

We now conduct another series of experiments that build upon the synthetic data sets presented in [Zhang et al., 2012, Li and Fan, 2020, Doran et al., 2014, Bellot and van der Schaar, 2019]. To compare type-I error rates, we generate simulated data for which H_0 is true:

$$X = f_1(\bar{Z} + \varepsilon_x), Y = f_2(\bar{Z} + \varepsilon_y). \tag{G.8}$$

Above, $\bar{Z}$ is the average of $Z = (Z_1, \cdots, Z_{d_z})$, ε_x and ε_y are sampled independently from a standard Gaussian or Laplace distribution, and f_1 and f_2 are smooth functions chosen uniformly from the set $\{(\cdot), (\cdot)^2, (\cdot)^3, \tanh(\cdot), \exp(-|\cdot|)\}$. To evaluate the power, we consider the following data generating function:

$$X = f_1(\bar{Z} + \varepsilon_x) + \varepsilon_b, Y = f_2(\bar{Z} + \varepsilon_y) + \varepsilon_b, \tag{G.9}$$

where ε_b is a standard Gaussian or Laplace distribution. As in the previous experiment, for each model, we study two settings by either fixing the dimension d_z, or the sample size n. In Figure G.4 (Laplace case) and G.13 (Gaussian case), we compare the KS and the AUPC of our method with the SoTA tests and demonstrate that our procedure manages to be powerful while controlling the type-I error. In Figures G.12 and G.15, we also compare the type-I and type-II errors of the different tests, and obtain similar conclusions. In addition, we investigate the high dimensional regime and show in Figure G.14 and G.17 that our test outperforms all the other proposed methods in most of the settings. See Appendix G.7.4 for more details.

G.5 Conclusion

We introduced a new kernel-based statistic for testing CI. We derived its asymptotic null distribution and designed a simple testing procedure that emerges from it. To our knowledge, we are the first article to propose an asymptotic test for CI with a tractable null distribution. Using various synthetic experiments, we demonstrated that our approach is competitive with other SoTA methods both in terms of type-I and type-II errors, even in the high dimensional setting.

G.6 Appendix: Proofs

G.6.1 On the Formulation of the Witness Function

Let $(\mathbf{t}_j)_{j=1}^J$ sampled independently from the Γ distribution, then by definition of $d_{p,J}(\cdot,\cdot)$, we have that

$$d_{p,J}(P_{XZY}, P_{\ddot{X}\otimes Y|Z}) := \left[\frac{1}{J}\sum_{j=1}^J \left|\mu_{P_{XZY},k_{\ddot{\mathcal{X}}}\cdot k_{\mathcal{Y}}}(\mathbf{t}_j) - \mu_{P_{\ddot{X}\otimes Y|Z},k_{\ddot{\mathcal{X}}}\cdot k_{\mathcal{Y}}}(\mathbf{t}_j)\right|^p\right]^{\frac{1}{p}},$$

Moreover thanks to Assumption 4, we have that for any $(\mathbf{t}^{(1)}, t^{(2)}) \in \ddot{\mathcal{X}} \times \mathcal{Y}$

$$\mu_{P_{\ddot{X}\otimes Y|Z},k_{\ddot{\mathcal{X}}}\cdot k_{\mathcal{Y}}}(\mathbf{t}^{(1)}, t^{(2)}) = \mathbb{E}_Z\left[\mathbb{E}_{\ddot{X}}\left[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)}, \ddot{X})|Z\right]\mathbb{E}_Y\left[k_{\mathcal{Y}}(t^{(2)}, Y)|Z\right]\right],$$

and

$$\mu_{P_{XZY},k_{\ddot{\mathcal{X}}}\cdot k_{\mathcal{Y}}}(\mathbf{t}^{(1)},t^{(2)}) = \mathbb{E}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})k_{\mathcal{Y}}(t^{(2)},Y)\Big] .$$

Let us now introduce the following witness function

$$\Delta(\mathbf{t}^{(1)},t^{(2)}) := \mathbb{E}\Big[\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X}) - \mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})|Z\Big]\Big) \times \Big(k_{\mathcal{Y}}(t^{(2)},Y) - \mathbb{E}_{Y}\Big[k_{\mathcal{Y}}(t^{(2)},Y)|Z\Big]\Big)\Big] .$$

Therefore we obtain that

$$\begin{aligned}\Delta(\mathbf{t}^{(1)},t^{(2)}) &= \mathbb{E}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})(k_{\mathcal{Y}}(t^{(2)},Y)\Big] \\ &- \mathbb{E}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})\mathbb{E}_{Y}\Big[k_{\mathcal{Y}}(t^{(2)},Y)|Z\Big]\Big] \\ &+ \mathbb{E}\Big[\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})|Z\Big]\mathbb{E}_{Y}\Big[k_{\mathcal{Y}}(t^{(2)},Y)|Z\Big]\Big] \\ &- \mathbb{E}\Big[\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})|Z\Big]k_{\mathcal{Y}}(t^{(2)},Y)\Big] .\end{aligned}$$

Now remarks that

$$\begin{aligned}\mathbb{E}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})\mathbb{E}_{Y}\Big[k_{\mathcal{Y}}(t^{(2)},Y)|Z\Big]\Big] &= \mathbb{E}\Big[\mathbb{E}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})\mathbb{E}_{Y}\Big[k_{\mathcal{Y}}(t^{(2)},Y)|Z\Big]|Z\Big]\Big] \\ &= \mathbb{E}\Big[\mathbb{E}_{Y}\Big[k_{\mathcal{Y}}(t^{(2)},Y)|Z\Big]\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})|Z\Big]\Big] .\end{aligned}$$

Simiarly, we have that

$$\mathbb{E}\Big[\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})|Z\Big]k_{\mathcal{Y}}(t^{(2)},Y)\Big] = \mathbb{E}\Big[\mathbb{E}_{Y}\Big[k_{\mathcal{Y}}(t^{(2)},Y)|Z\Big]\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})|Z\Big]\Big]$$

from which follows that

$$\begin{aligned}\Delta(\mathbf{t}^{(1)},t^{(2)}) &= \mathbb{E}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})(k_{\mathcal{Y}}(t^{(2)},Y)\Big] - \mathbb{E}\Big[\mathbb{E}_{Y}\Big[k_{\mathcal{Y}}(t^{(2)},Y)|Z\Big]\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}^{(1)},\ddot{X})|Z\Big]\Big] \\ &= \mu_{P_{XZY},k_{\ddot{\mathcal{X}}}\cdot k_{\mathcal{Y}}}(\mathbf{t}^{(1)},t^{(2)}) - \mu_{P_{\ddot{X}\otimes Y|Z},k_{\ddot{\mathcal{X}}}\cdot k_{\mathcal{Y}}}(\mathbf{t}^{(1)},t^{(2)}) .\end{aligned}$$

G.6.2 Proof of Proposition 43

Proof. For all $j \in [J]$:

$$\begin{aligned}&\sqrt{n}\widetilde{\Delta}_{n,r}(\mathbf{t}_j^{(1)},t_j^{(2)}) && \text{(G.10)} \\ &= \sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{x}_i) - h_{j,r}^{(1)}(z_i)\Big)\Big(k_{\mathcal{Y}}(t_j^{(2)},y_i) - h_{j,r}^{(2)}(z_i)\Big) \\ &= \sqrt{n}\Delta_n(\mathbf{t}_j^{(1)},t_j^{(2)}) && \text{(G.11)}\end{aligned}$$

$$+\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{x}_i)-\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{X})|Z=z_i\Big]\Big)\Big(\mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)},Y)|Z=z_i\Big]-h_{j,r}^{(2)}(z_i)\Big) \tag{G.12}$$

$$+\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\Big(\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{X})|Z=z_i\Big]-h_{j,r}^{(1)}(z_i)\Big)\Big(k_{\mathcal{Y}}(t_j^{(2)},y_i)-\mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)},Y)|Z=z_i\Big]\Big) \tag{G.13}$$

$$+\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\Big(\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{X})|Z=z_i\Big]-h_{j,r}^{(1)}(z_i)\Big)\Big(\mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)},Y)|Z=z_i\Big]-h_{j,r}^{(2)}(z_i)\Big) \tag{G.14}$$

Let us treat the four terms of this decomposition. The term (G.11) has been treated by Propostion 42, and satisfies, under the null hypothesis H_0

$$\sqrt{n}\Delta_n(\mathbf{t}_j^{(1)},t_j^{(2)})$$
$$\to_{n\to\infty}\mathcal{N}\Big(0,\mathbb{E}\Big[\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{X})-\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{X})|Z\Big]\Big)\Big(k_{\mathcal{Y}}(t_j^{(2)},Y)-\mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)},Y)|Z\Big]\Big)\Big]\Big)$$

Let us now show that the last term (G.14) converges towards 0 in probability. Let us denote for all j, $e_j^{(1)}: z\to\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{X})|Z=z\Big]$ and $e_j^{(2)}: z\to\mathbb{E}_Y\Big[k_{\ddot{\mathcal{X}}}(t_j^{(2)},Y)|Z=z\Big]$, both elements of $H_{\mathcal{Z}}$ by Assumption 6. Then we have, for all $i\in[n]$:

$$\Big(e_j^{(1)}(z_i)-h_{j,r}^{(1)}(z_i)\Big)\Big(e_j^{(2)}(z_i)-h_{j,r}^{(2)}(z_i)\Big)=\langle\Big(e_j^{(1)}-h_{j,r}^{(1)}\Big)\otimes\Big(e_j^{(2)}-h_{j,r}^{(2)}\Big),k_{\mathcal{Z}}(z_i,\cdot)\otimes k_{\mathcal{Z}}(z_i,\cdot)\rangle.$$

Then we deduce, by denoting: $\mu_{ZZ}:=\mathbb{E}[k_{\mathcal{Z}}(Z,\cdot)k_{\mathcal{Z}}(Z,\cdot)]$ and $\hat{\mu}_{ZZ}:=\frac{1}{n}\sum_{i=1}^n k_{\mathcal{Z}}(z_i,\cdot)k_{\mathcal{Z}}(z_i,\cdot)$, that

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\Big(\mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)},\ddot{X})|Z=z_i\Big]-h_{j,r}^{(1)}(z_i)\Big)\Big(\mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)},Y)|Z=z_i\Big]-h_{j,r}^{(2)}(z_i)\Big)\\
=\langle\Big(e_j^{(1)}-h_{j,r}^{(1)}\Big)\otimes\Big(e_j^{(2)}-h_{j,r}^{(2)}\Big),\frac{1}{n}\sum_{i=1}^{n}k_{\mathcal{Z}}(z_i,\cdot)\otimes k_{\mathcal{Z}}(z_i,\cdot)\rangle\end{aligned}$$
$$=\langle\Big(e_j^{(1)}-h_{j,r}^{(1)}\Big)\otimes\Big(e_j^{(2)}-h_{j,r}^{(2)}\Big),\mu_{ZZ}\rangle+\langle\Big(e_j^{(1)}-h_{j,r}^{(1)}\Big)\otimes\Big(e_j^{(2)}-h_{j,r}^{(2)}\Big),\hat{\mu}_{ZZ}-\mu_{ZZ}\rangle\ .$$

Then remarks that:

$$\begin{aligned}|\langle\Big(e_j^{(1)}-h_{j,r}^{(1)}\Big)\otimes\Big(e_j^{(2)}-h_{j,r}^{(2)}\Big),\mu_{ZZ}\rangle|&=|\mathbb{E}_Z\Big[\Big(e_j^{(1)}(Z)-h_{j,r}^{(1)}(Z)\Big)\Big(e_j^{(2)}(Z)-h_{j,r}^{(2)}(Z)\Big)\Big]|\\
&\le\|e_j^{(1)}-h_{j,r}^{(1)}\|_{L^2(P_Z)}\|e_j^{(2)}-h_{j,r}^{(2)}\|_{L^2(P_Z)}\end{aligned}$$

Under the Assumptions 5-6, for $\lambda_r = \frac{1}{r^{\beta+\gamma}}$, we have, using the results from Fischer and Steinwart [2020]: $\|e_j^{(1)} - h_{j,r}^{(1)}\|_{L^2(P_Z)}^2 \leq \frac{C\tau^2}{r^{\frac{\beta}{\beta+\gamma}}}$ with probability $1-4e^{-\tau}$ and $\|e_j^{(2)} - h_{j,r}^{(2)}\|_{L^2(P_Z)}^2 \leq \frac{C\tau^2}{r^{\frac{\beta}{\beta+\gamma}}}$ with probability $1-4e^{-\tau}$, for some constant C independent from n and τ. then by union bound, we deduce with probability $1-8e^{-\tau}$ we have:

$$\sqrt{n}|\langle \left(e_j^{(1)} - h_{j,r}^{(1)}\right) \otimes \left(e_j^{(2)} - h_{j,r}^{(2)}\right), \mu_{ZZ}\rangle| \leq \sqrt{n}\frac{C^2\tau^4}{r^{\frac{\beta}{\beta+\gamma}}}$$

Then, if $\sqrt{n} \in o(r^{\frac{\beta}{\beta+\gamma}})$, we have: $\sqrt{n}|\langle \left(e_j^{(1)} - h_{j,r}^{(1)}\right) \otimes \left(e_j^{(2)} - h_{j,r}^{(2)}\right), \mu_{ZZ}\rangle| \to 0$ in probability when $n \to \infty$. Moreover:

$$|\left(e_j^{(1)} - h_{j,r}^{(1)}\right) \otimes \left(e_j^{(2)} - h_{j,r}^{(2)}\right), \hat{\mu}_{ZZ} - \mu_{ZZ}\rangle| \leq \|e_j^{(1)} - h_{j,r}^{(1)}\|_{H_\mathcal{Z}}\|e_j^{(2)} - h_{j,r}^{(2)}\|_{H_\mathcal{Z}}\|\hat{\mu}_{ZZ} - \mu_{ZZ}\|_{H_\mathcal{Z}\otimes H_\mathcal{Z}}\ ,$$

and by Markov inequality, $\|\hat{\mu}_{ZZ} - \mu_{ZZ}\|_{H_\mathcal{Z}\otimes H_\mathcal{Z}} \leq \sqrt{\frac{C'}{n\delta}}$ with probability $1-\delta$ for some constant C'. Moreover, under Assumption 5-6, we have $\|e_j^{(1)} - h_{j,r}^{(1)}\|_{H_\mathcal{Z}} \to 0$ and $\|e_j^{(2)} - h_{j,r}^{(2)}\|_{H_\mathcal{Z}} \to 0$ in probability. Then, we deduce that $\sqrt{n}|\langle \left(e_j^{(1)} - h_{j,r}^{(1)}\right) \otimes \left(e_j^{(2)} - h_{j,r}^{(2)}\right), \hat{\mu}_{ZZ} - \mu_{ZZ}\rangle| \to 0$ in probability. Finally, the term (G.14) goes to 0 in probability.

The terms (G.12) and (G.13) are similar and can be treated the same way. We only focus on the term (G.12). For all $i \in [n]$:

$$\begin{aligned}
&|\frac{1}{n}\sum_{i=1}^n\left(k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{x}_i) - \mathbb{E}_{\ddot{X}}\left[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X})|Z=z_i\right]\right)\left(\mathbb{E}_Y\left[k_\mathcal{Y}(t_j^{(2)}, Y)|Z=z_i\right] - h_{j,r}^{(2)}(z_i)\right)| \\
&\quad = |\frac{1}{n}\sum_{i=1}^n\langle k_{\ddot{\mathcal{X}}}(t_j^{(1)}, \cdot), k_{\ddot{\mathcal{X}}}(\ddot{x}_i, \cdot) - \mathbb{E}_{\ddot{X}}\left[k_{\ddot{\mathcal{X}}}(\ddot{X}, \cdot)|Z=z_i\right]\rangle_{H_{\ddot{\mathcal{X}}}}\langle e_j^{(2)} - h_{j,r}^{(2)}, k_\mathcal{Z}(z_i, \cdot)\rangle_{H_\mathcal{Z}}| \\
&= |\frac{1}{n}\sum_{i=1}^n\langle k_{\ddot{\mathcal{X}}}(t^{(1)}, \cdot) \otimes \left(e_j^{(2)} - h_{j,r}^{(2)}\right), \left(k_{\ddot{\mathcal{X}}}(\ddot{x}_i, \cdot) - \mathbb{E}_{\ddot{X}}\left[k_{\ddot{\mathcal{X}}}(\ddot{X}, \cdot)|Z=z_i\right]\right) \otimes k_\mathcal{Z}(z_i, \cdot)\rangle_{H_{\ddot{\mathcal{X}}}\otimes H_\mathcal{Z}}| \\
&= |\langle k_{\ddot{\mathcal{X}}}(t^{(1)}, \cdot) \otimes \left(e_j^{(2)} - h_{j,r}^{(2)}\right), \frac{1}{n}\sum_{i=1}^n\left(k_{\ddot{\mathcal{X}}}(\ddot{x}_i, \cdot) - \mathbb{E}_{\ddot{X}}\left[k_{\ddot{\mathcal{X}}}(\ddot{X}, \cdot)|Z=z_i\right]\right) \otimes k_\mathcal{Z}(z_i, \cdot)\rangle_{H_{\ddot{\mathcal{X}}}\otimes H_\mathcal{Z}}| \\
&\quad \leq \|k_{\ddot{\mathcal{X}}}(t^{(1)}, \cdot)\|_{H_{\ddot{\mathcal{X}}}}\|e_j^{(2)} - h_{j,r}^{(2)}\|_{H_\mathcal{Z}}\left(\|\hat{\mu}_{\ddot{X}Z}^1 - \mu_{\ddot{X}Z}\|_{H_{\ddot{\mathcal{X}}}\otimes H_\mathcal{Z}} + \|\hat{\mu}_{\ddot{X}}^2 - \mu_{\ddot{X}Z}\|_{H_{\ddot{\mathcal{X}}}\otimes H_\mathcal{Z}}\right)
\end{aligned}$$

where: $\hat{\mu}_{\ddot{X}Z}^1 := \frac{1}{n}\sum_{i=1}^n k_{\ddot{\mathcal{X}}}(\ddot{x}_i, \cdot)\otimes k_\mathcal{Z}(z_i, \cdot)$, $\hat{\mu}_{\ddot{X}Z}^2 := \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\ddot{X}}\left[k_{\ddot{\mathcal{X}}}(\ddot{X}, \cdot)|Z=z_i\right]\otimes k_\mathcal{Z}(z_i, \cdot)$, and $\mu_{\ddot{X}Z} := \mathbb{E}[k_\mathcal{Y}(y, \cdot)k_\mathcal{Z}(z, \cdot)]$.

By the law of large numbers, we have: $\hat{\mu}_{\ddot{X}Z}^1$ and $\hat{\mu}_{\ddot{X}Z}^2$ converge almost surely towards $\mu_{\ddot{X}Z}$. Moreover by Markov inequality, $\|\hat{\mu}_{\ddot{X}Z}^1 - \mu_{\ddot{X}Z}\|_{H_{\ddot{\mathcal{X}}}\otimes H_\mathcal{Z}} \leq \sqrt{\frac{C}{n\delta}}$ with probability

$1-\delta$, and $\|\hat{\mu}^2_{\ddot{X}Z} - \mu_{\ddot{X}Z}\|_{H_{\ddot{\mathcal{X}}}\otimes H_{\mathcal{Z}}} \leq \sqrt{\frac{C}{n\delta}}$ with probability $1-\delta$. Then with probability $1-2\delta$, $\sqrt{n}\Big(\|\hat{\mu}^1_{\ddot{X}Z} - \mu_{\ddot{X}Z}\|_{H_{\ddot{\mathcal{X}}}\otimes H_{\mathcal{Z}}} + \|\hat{\mu}^2_{\ddot{X}Z} - \mu_{\ddot{X}Z}\|_{H_{\ddot{\mathcal{X}}}\otimes H_{\mathcal{Z}}}\Big) \leq 2\sqrt{\frac{C}{\delta}}$. Moreover, under Assumption 5-6, using the results from Fischer and Steinwart [2020], we have that $\|e_j^{(2)} - h_{j,r}^{(2)}\|_{H_{\mathcal{Z}}}$ converges towards 0 in probability. Then the term (G.12) converges in probability towards 0. The same reasoning holds for (G.13).

Finally, by Slutsky's Lemma:

$$\sqrt{n}\widetilde{\Delta}_{n,r}(\mathbf{t}_j^{(1)}, t_j^{(2)}) \\ \to_{n\to\infty} \mathcal{N}\Big(0, \mathbb{E}\Big[\Big(k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X}) - \mathbb{E}_{\ddot{X}}\Big[k_{\ddot{\mathcal{X}}}(\mathbf{t}_j^{(1)}, \ddot{X})|Z\Big]\Big)\Big(k_{\mathcal{Y}}(t_j^{(2)}, Y) - \mathbb{E}_Y\Big[k_{\mathcal{Y}}(t_j^{(2)}, Y)|Z\Big]\Big)\Big]\Big).$$

Now we have:

$$\widetilde{\mathbf{S}}_{n,r} = \Big(\widetilde{\Delta}_{n,r}(\mathbf{t}_j^{(1)}, t_j^{(2)})\Big)_{j\in[J]} = \Big(\Delta_n(\mathbf{t}_j^{(1)}, t_j^{(2)})\Big)_{j\in[J]} + \Big(\widetilde{\Delta}_{n,r}(\mathbf{t}_j^{(1)}, t_j^{(2)}) - \Delta_n(\mathbf{t}_j^{(1)}, t_j^{(2)})\Big)_{j\in[J]}$$

and we have shown that $\sqrt{n}\Big(\widetilde{\Delta}_{n,r_n}(\mathbf{t}_j^{(1)}, t_j^{(2)}) - \Delta_n(\mathbf{t}_j^{(1)}, t_j^{(2)})\Big)_{j\in[J]}$ goes to 0 in probability. Then by Slutsky Lemma and Proposition 42, we get: $\widetilde{\mathbf{S}}_{n,r_n} \to \mathcal{N}(0, \boldsymbol{\Sigma})$.

Let $r > 0$. Under H_1, $\mathbf{S}_{n,r_n} \to \mathbf{S} \neq 0$. Let consider a realization of $(\mathbf{t}_j^{(1)}, t_j^{(2)})_{j\in[J]}$ such that $\|\mathbf{S}\|_p \neq 0$. So $P(n^{p/2}\|\mathbf{S}_{n,r_n}\|_p \geq r) \to 1$ as $n \to \infty$ because $\|\mathbf{S}\|_p \neq 0$. □

G.6.3 Proof of Proposition 44

Proof. First notice that:

$$\begin{aligned}
\widetilde{\boldsymbol{\Sigma}}_{n,r} &:= \frac{1}{n}\sum_{i=1}^{n} \widetilde{\mathbf{u}}_{i,r}\widetilde{\mathbf{u}}_{i,r}^{T} + \delta_n \mathrm{Id}_J \\
&= \widehat{\boldsymbol{\Sigma}}_n + \frac{1}{n}\sum_{i=1}^{n} \widehat{\mathbf{u}}_i(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)^T + \frac{1}{n}\sum_{i=1}^{n}(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)\widehat{\mathbf{u}}_i^T \\
&+ \frac{1}{n}\sum_{i=1}^{n}(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)^T + \delta_n \mathrm{Id}_J
\end{aligned}$$

By the law of large numbers, we get that under H_0: $\widehat{\boldsymbol{\Sigma}}_n \to \boldsymbol{\Sigma}$. Moreover:

$$\left[\frac{1}{n}\sum_{i=1}^{n} \widehat{\mathbf{u}}_i(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)^T\right]_{kl} = \frac{1}{n}\sum_{i=1}^{n}\Big(k_{\mathcal{Y}}(t_k^{(2)}, y_i) - \mathbb{E}_Y\left[k_{\mathcal{Y}}(t_k^{(2)}, Y)|Z = z_i\right]\Big)\Big(\mathbb{E}_{\ddot{X}}\left[k_{\ddot{\mathcal{X}}}(\mathbf{t}_l^{(1)}, \ddot{X})|Z = z_i\right] - h_{l,r}^{(1)}(z$$

which has been proven to converge in probability to 0 in the proof of Proposition 43. Then $\frac{1}{n}\sum_{i=1}^{n} \widehat{\mathbf{u}}_i(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)^T$ converges in probability to 0. Similarly $\frac{1}{n}\sum_{i=1}^{n}(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)\widehat{\mathbf{u}}_i^T$ and $\frac{1}{n}\sum_{i=1}^{n}(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)(\widetilde{\mathbf{u}}_{i,r} - \widehat{\mathbf{u}}_r)^T$ also converge in probability to 0. Then by Slutsky Lemma, $\widetilde{\boldsymbol{\Sigma}}_{n,r}$ converges in probability to $\boldsymbol{\Sigma}$. By Slutsky's lemma (again) and by Proposition 43, we have that: $\widetilde{\boldsymbol{\Sigma}}_{n,r}^{-1}\widetilde{\boldsymbol{S}}_{n,r}$ converges to a standard gaussian distribution $\mathcal{N}(0, \mathrm{Id})$. The second part of the proposition is the same than the proof of Proposition 43. $\square$

G.7 Appendix: Additional Experiments

G.7.1 A note on the computation of Oracle statistic in Figure G.2

To compute the oracle statistic we needed to compute exactly the conditional expectation implied in our statistic. In the case of gaussian kernels and gaussian distributed data for Z, the computation of this conditional expectation is reduced to the computation of moment-generating function of a non-centered χ^2 distribution.

G.7.2 Choice of the rank regression r

In this experiment, we show the effect of the rank regression r on the performances of our proposed method. For that purpose, in Figure G.5, we consider the two problems presented in (G.4) and (G.5) with Gaussian noises and show the type-I and type-II when varying the ratio r/n for multiple sample size n. We observe that the rank r does not affect the power of the method, however we observe that the type-I error decreases as the ratio increases. Therefore the rank r allows in practice to deal with the tradeoff between the computational time and the control of the type-I error.

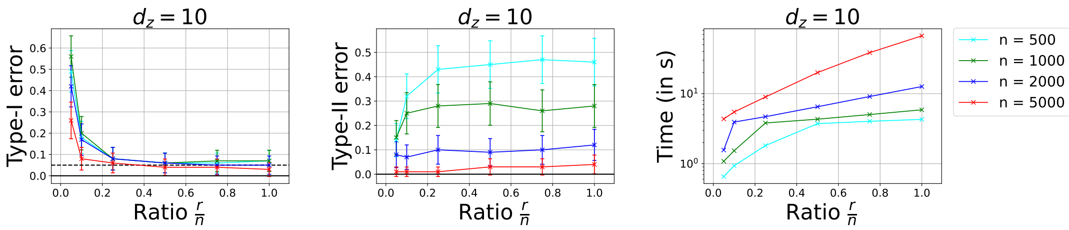


Figure G.5: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line) and the type-II error (lower is better) of our test procedure with other SoTA tests on the two problems presented in (G.4) and (G.5) with Gaussian noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, Middle*): type-I and type-II errors obtained by each test when varying the ratio regression rank/total number of samples for different number of samples. (*Right*): time in seconds (log-scale) to compute the statistic when varying the ratio regression rank/total number of samples for different number of samples.

G.7.3 Additional experiments on Problems (G.4) and (G.5)

Gaussian Case

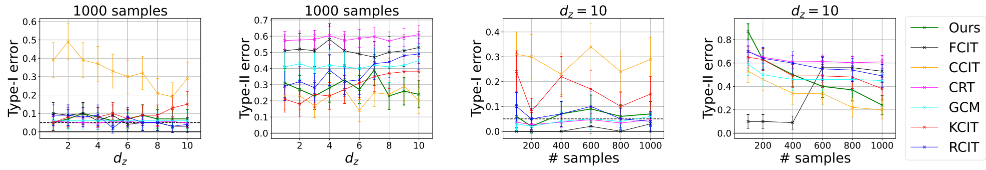


Figure G.6: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line) and the type-II error (lower is better) of our test procedure with other SoTA tests on the two problems presented in (G.4) and (G.5) with Gaussian noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): type-I and type-II errors obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): type-I and type-II errors obtained by each test when varying the number of samples n from 100 to 1000; the dimension d_z is fixed and equals 10.

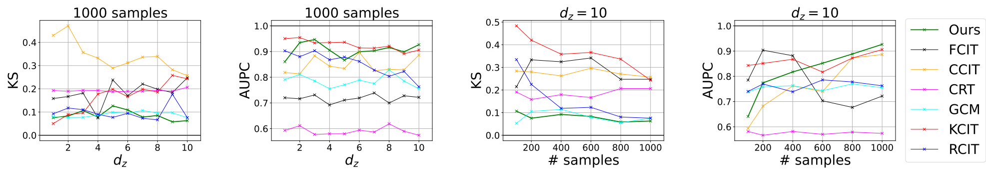


Figure G.7: Comparison of the KS statistic (lower is better) and the AUPC (higher is better) of our testing procedure with other SoTA tests on the two problems presented in (G.4) and (G.5) with Gaussian noises. Each point in the figures is obtained by repeating the experiment forx 100 independent trials. (*Left, middle-left*): the KS and AUPC obtained by each test when varying the dimension d_z from 1 to 10, while fixing the number of samples n to 1000. (*Middle-right, right*): the KS and AUPC obtained by each test when varying the number of samples n from 100 to 1000, while fixing the dimension d_z to 10.

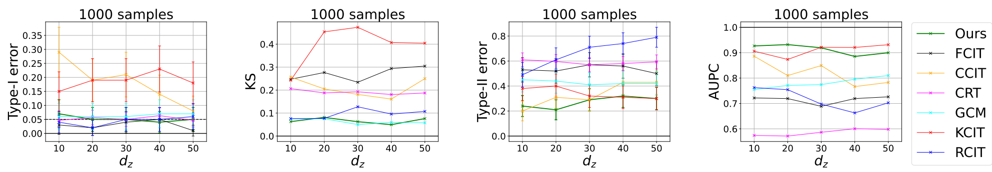


Figure G.8: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line), type-II error (lower is better), KS statistic and the AUPC of our testing procedure with other SoTA tests on the two problems presented in Eq. (G.4) and Eq. (G.5) with Gaussian noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. In each plot the dimension d_z is varying from 10 to 50; here, the number of samples n is fixed and equals to 1000.

Laplace Case

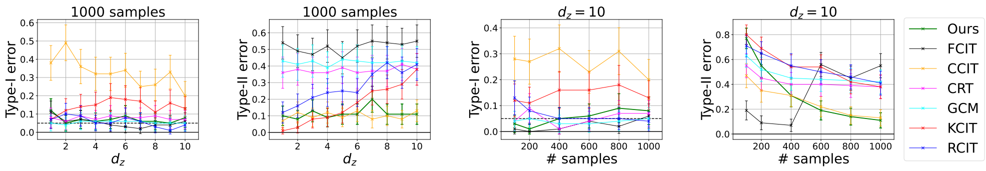


Figure G.9: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line) and the type-II error (lower is better) of our test procedure with other SoTA tests on the two problems presented in (G.4) and (G.5) with Laplace noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): type-I and type-II errors obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): type-I and type-II errors obtained by each test when varying the number of samples n from 100 to 1000; here, the dimension d_z is fixed and equals to 10.

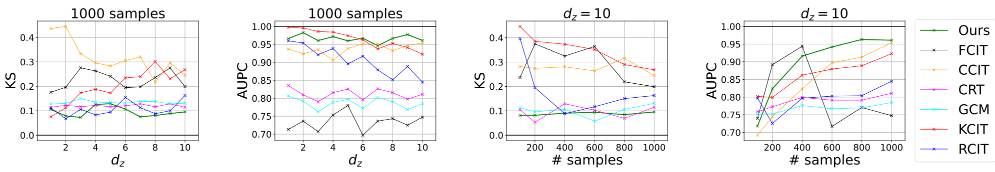


Figure G.10: Comparison of the KS statistic and the AUPC of our testing procedure with other SoTA tests on the two problems presented in Eq. (G.4) and Eq. (G.5) with Laplace noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): the KS statistic and AUPC (respectively) obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): the KS and AUPC (respectively), obtained by each test when varying the number of samples n from 100 to 1000; here, the dimension d_z is fixed and equals to 10.

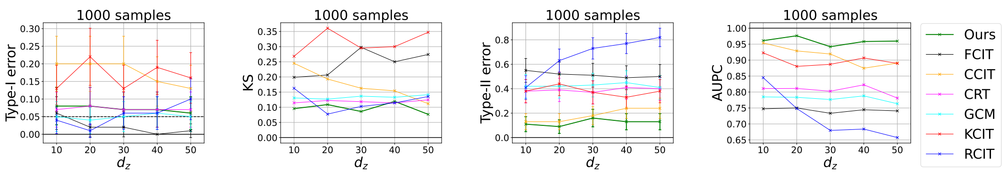


Figure G.11: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line), type-II error (lower is better), KS statistic and the AUPC of our testing procedure with other SoTA tests on the two problems presented in Eq. (G.4) and Eq. (G.5) with Laplace noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. In each plot the dimension d_z is varying from 10 to 50; here, the number of samples n is fixed and equals to 1000.

G.7.4 Additional experiments on Problems (G.8) and (G.9)

Gaussian Case

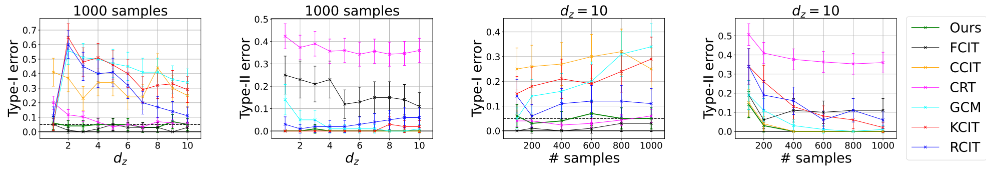


Figure G.12: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line) and the type-II error (lower is better) of our test procedure with other SoTA tests on the two problems presented in (G.8) and (G.9) with Gaussian noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): type-I and type-II errors obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): type-I and type-II errors obtained by each test when varying the number of samples n from 100 to 1000; here, the dimension d_z is fixed and equals to 10.

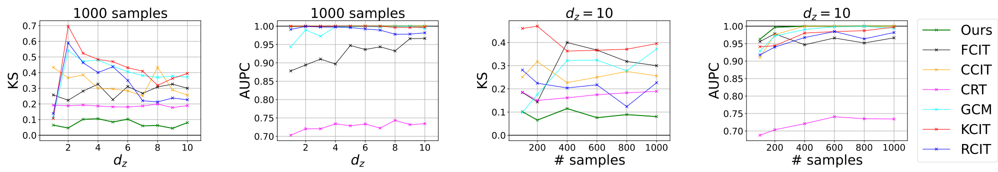


Figure G.13: Comparison of the KS statistic and the AUPC of our testing procedure with other SoTA tests on the two problems presented in Eq. (G.8) and Eq. (G.9) with Gaussian noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): the KS statistic and AUPC (respectively) obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): the KS and AUPC (respectively), obtained by each test when varying the number of samples n from 100 to 1000; the dimension d_z is fixed and equals 10.

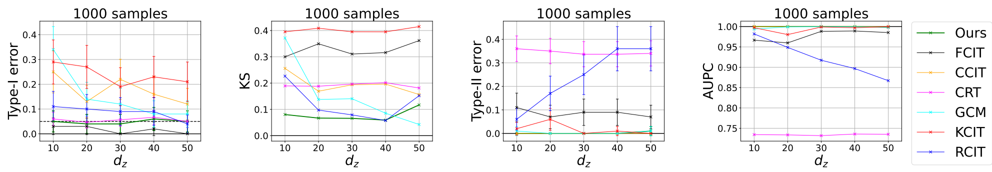


Figure G.14: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line), type-II error (lower is better), KS statistic and the AUPC of our testing procedure with other SoTA tests on the two problems presented in Eq. (G.8) and Eq. (G.9) with Gaussian noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. In each plot the dimension d_z is varying from 10 to 50; here, the number of samples n is fixed and equals to 1000.

Laplace Case

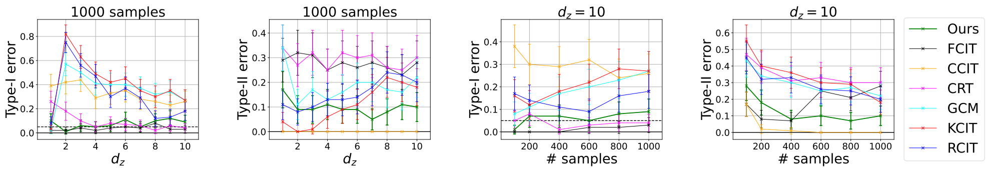


Figure G.15: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line) and the type-II error (lower is better) of our test procedure with other SoTA tests on the two problems presented in (G.8) and (G.9) with Laplace noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): type-I and type-II errors obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): type-I and type-II errors obtained by each test when varying the number of samples n from 100 to 1000; the dimension d_z is fixed and equals 10.

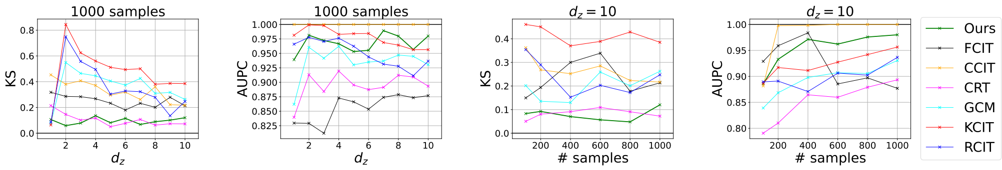


Figure G.16: Comparison of the KS statistic and the AUPC of our testing procedure with other SoTA tests on the two problems presented in Eq. (G.8) and Eq. (G.9) with Laplace noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. (*Left, middle-left*): the KS statistic and AUPC (respectively) obtained by each test when varying the dimension d_z from 1 to 10; here, the number of samples n is fixed and equals to 1000. (*Middle-right, right*): the KS and AUPC (respectively), obtained by each test when varying the number of samples n from 100 to 1000; the dimension d_z is fixed and equals 10.

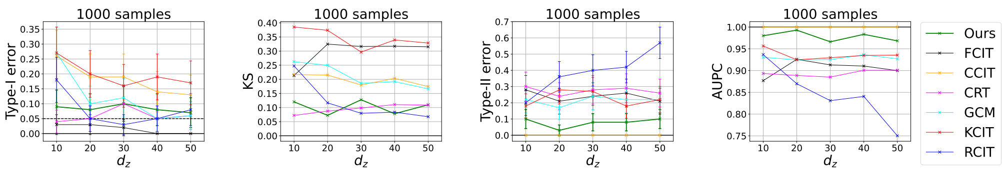


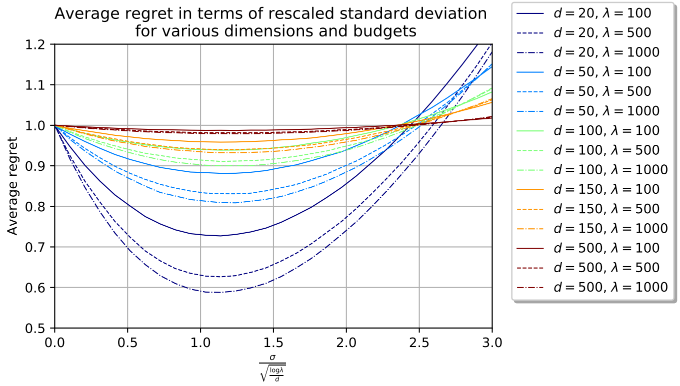
Figure G.17: Comparison of the type-I error at level $\alpha = 0.05$ (dashed line), type-II error (lower is better), KS statistic and the AUPC of our testing procedure with other SoTA tests on the two problems presented in Eq. (G.8) and Eq. (G.9) with Laplace noises. Each point in the figures is obtained by repeating the experiment for 100 independent trials. In each plot the dimension d_z is varying from 10 to 50; here, the number of samples n is fixed and equals to 1000.

H Variance Reduction for Better Sampling in Continuous Domains

Design of experiments, random search, initialization of population-based methods, or sampling inside an epoch of an evolutionary algorithm uses a sample drawn according to some probability distribution for approximating the location of an optimum. Recent papers have shown that 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. We confirm this statement, provide explicit values for this reshaping of the search distribution depending on the population size λ and the dimension d, and validate our results experimentally.

H.1 Introduction

We consider the setting in which one aims to locate an optimal solution $x^* \in \mathbb{R}^d$ for a given black-box problem $f : \mathbb{R}^d \to \mathbb{R}$ through a parallel evaluation of λ solution candidates. A simple, yet effective strategy for this *one-shot optimization* setting is to choose the λ candidates from a normal distribution $\mathcal{N}(\mu, \sigma^2)$, typically centered around an *a priori* estimate μ of the optimum and using a variance σ^2 that is calibrated according to the uncertainty with respect to the optimum. Random independent sampling is – despite its simplicity – still a very commonly used and performing good technique in one-shot optimization settings. There also exist more sophisticated sampling strategies like Latin Hypercube Sampling (LHS McKay et al. [1979b]), or quasi-random constructions such as Sobol, Halton, Hammersley sequences Dick and Pillichshammer [2010], Matoušek [2010] – see Bergstra and Bengio [2012], Cauwet et al. [2019] for examples. However, no general superiority of these strategies over random sampling can be observed when the benchmark set is sufficiently diverse Bossek et al. [2019]. It is therefore not surprising that in several one-shot settings – for example, the design of experiments Niederreiter [1992], McKay et al. [1979a], Hammersley [1960], Atanassov [2004] or the initialization (and sometimes also further iterations) of evolution strategies – the solution candidates are frequently sampled from random independent distributions (though sometimes improved by mirrored sampling Teytaud et al. [2006]). A surprising finding was recently communicated in Cauwet et al. [2019], where the authors consider the setting in which the optimum x^* is known to be distributed according to a standard normal distribution $\mathcal{N}(0, I_d)$, and the goal is to minimize the distance of the best of the λ samples to this optimum. In the context of evolution strategies, one would formulate this problem as minimizing the sphere function with a normally distributed optimum. Intuitively, one might guess that sampling the λ candidates from the same prior distribution, $\mathcal{N}(0, I_d)$, should be optimal. This intuition, however, was disproved in Cauwet et al. [2019], where it is shown that – unless



d	λ	σ^*	$\sigma = 1$
20	100	**0.73**	0.88
	500	**0.63**	0.72
	1000	**0.59**	0.66
50	100	**0.89**	1.23
	500	**0.83**	1.10
	1000	**0.81**	1.05
100	100	**0.94**	1.44
	500	**0.91**	1.33
	1000	**0.90**	1.29
150	100	**0.96**	1.53
	500	**0.94**	1.44
	1000	**0.93**	1.41
500	100	**0.99**	1.74
	500	**0.98**	1.68
	1000	**0.98**	1.66

Figure H.1: Average regret, normalized by d, on the sphere function for various dimensions and budgets in terms of rescaled standard deviation. Each mean has been estimated from $100,000$ samples. Table on the right: Average regret for $\sigma^* = \sqrt{\log(\lambda)/d}$ and $\sigma = 1$.

the sample size λ grows exponentially fast in the dimension d – the median quality of sampling from $\mathcal{N}(0, I_d)$ is worse than that of sampling a single point, namely the center point 0. A similar observation was previously made in Rahnamayan and Wang [2009], without mathematically proven guarantees.

Our Theoretical Result. It was left open in Cauwet et al. [2019] how to optimally scale the variance σ^2 when sampling the λ solution candidates from a normal distribution $\mathcal{N}(0, \sigma^2 I_d)$. While the result from Cauwet et al. [2019] suggests to use $\sigma = 0$, we show in this work that a more effective strategy exists. More precisely, we show that setting $\sigma^2 = \min\{1, \Theta(\log(\lambda)/d)\}$ is asymptotically optimal, as long as λ is sub-exponential, but growing in d. Our variance scaling factor reduces the median approximation error by a $1-\varepsilon$ factor, with $\varepsilon = \Theta(\log(\lambda)/d)$. We also prove that no constant variance nor any other variance scaling as $\omega(\log(\lambda)/d)$ can achieve such an approximation error. Note that several optimization algorithms operate with rescaled sampling. Our theoretical results therefore set the mathematical foundation for empirical rules of thumb such as, for example, used in e.g. Rahnamayan and Wang [2009], Esmailzadeh and Rahnamayan [2011], Mahdavi et al. [2016], Esmailzadeh and Rahnamayan [2012], Ergezer and Sikder [2011], Yang et al. [2011], Cauwet et al. [2019].

Our Empirical Results. We complement our theoretical analyses by an empirical investigation of the rescaled sampling strategy. Experiments on the sphere function confirm the results. We also show that our scaling factor for the variance yields excellent performance on two other benchmark problems, the Cigar and the Rastrigin function. Finally, we demonstrate that these improvements are not restricted to the one-shot setting by applying them to the initialization of iterative optimization strategies. More precisely, we show a positive impact on the initialization of Bayesian optimization algorithms Jones et al. [1998] and on differential evolution Storn and Price [1997].

Related Work. While the most relevant works for our study have been mentioned above, we briefly note that a similar surprising effect as observed here is the "Stein phenomenon" Stein [1956], James and Stein [1961]. Although an intuitive way to estimate the mean of a standard gaussian distribution is to compute the empirical mean, Stein showed that this strategy is sub-optimal w.r.t. mean squared error and that the empirical mean needs to be rescaled by some factor to be optimal.

H.2 Problem Statement and Related Work

The context of our theoretical analysis is *one-shot optimization*. In one-shot optimization, we are allowed to select λ points $x_1, \ldots, x_\lambda \in \mathbb{R}^d$. The quality $f(x_i)$ of these points is evaluated, and we measure the performance of our samples in terms of simple regret Bubeck et al. [2009] $\min_{i=1,\ldots,\lambda} f(x_i) - \inf_{x\in\mathbb{R}^d} f(x)$.[1] That is, we aim to minimize the distance – measured in *quality space* – of the best of our points to the optimum. This formulation, however, also covers the case in which we aim to minimize the distance to the optimum in the *search space*: we simply take as f the root of the sphere function $f_{x^*} : \mathbb{R}^d \to \mathbb{R}, x \mapsto \|x - x^*\|^2$, where here and in the following $\|.\|$ denotes the Euclidean norm.

Rescaled Random Sampling for Randomly Placed Optimum. In the setting studied in Sec. H.3 we assume that the optimum x^* is sampled from the standard multivariate Gaussian distribution $\mathcal{N}(0, I_d)$, and that we aim to minimize the regret $\min_{i=1,\ldots,\lambda}\|x_i - x^*\|^2$ through i.i.d. samples $x_i \sim \mathcal{N}(0, \sigma^2 I_d)$. That is, in contrast to the classical *design of experiments* (DoE) setting, we are only allowed to choose the scaling factor σ, whereas in DoE more sophisticated (often quasi-random and space-filling designs – which are typically not i.i.d. samples) are admissible. Intuitively, one might be tempted to guess that $\sigma = 1$ should be a good choice, as in this case the λ points are chosen from the same distribution as the optimum x^*. This intuition, however, was refuted in [Cauwet et al., 2019, Theorem 1], where is was shown that the middle point sampling strategy, which uses $\sigma = 0$ (i.e., all λ points collapse to $(0, \ldots, 0)$) yields smaller regret than sampling from $\mathcal{N}(0, I_d)$ unless λ grows exponentially in d. More precisely, it is shown in Cauwet et al. [2019] that, for this regime of λ and d, the median of $\|x^*\|^2$ is smaller than the median of $\|x_i - x^*\|^2$ for i.i.d. $x_i \in \mathcal{N}(0, I_d)$. This shows that sampling a single point can be better than sampling λ points with the wrong scaling factor, unless the budget λ is very large.
Our goal is to improve upon the middle point strategy, by deriving a scaling factor σ such that the λ i.i.d. samples yield smaller regret with a decent probability. More precisely, we aim at identifying σ such that

$$\mathbb{P}\left[\min_{1\le i\le\lambda}\|x_i - x^*\|^2 \le (1-\varepsilon)\|x^*\|^2\right] \ge \delta, \tag{H.1}$$

for some $\delta \ge 1/2$ and $\varepsilon > 0$ as large as possible. Here, in line with Cauwet et al. [2019], we have switched to regret, for convenience of notation. Cauwet et al. [2019] proposed, without proof, such a scaling factor: our proposal is dramatically better in some regimes.

[1] This requires knowledge of $\inf_x f(x)$, which may not be available in real-world applications. In this case, without loss of generality (this is just for the sake of plotting regret values), the infimum can be replaced by an empirical minimum. In all applications considered in this work the value of $\inf_x f(x)$ is known.

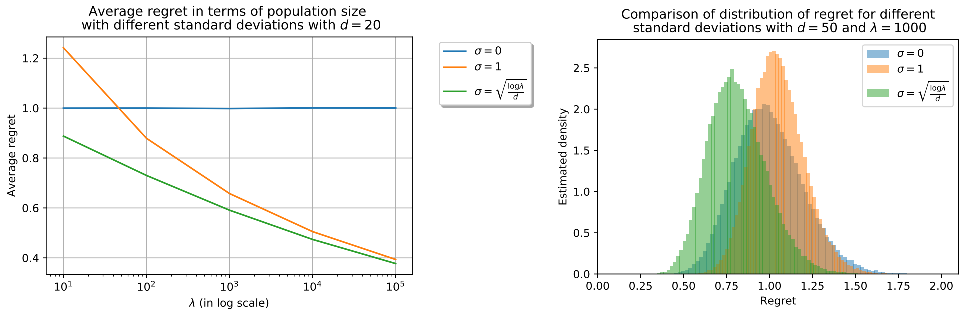


Figure H.2: Comparison of methods: without rescaling ($\sigma = 1$), middle point sampling ($\sigma = 0$), and our rescaling method ($\sigma = \sqrt{\frac{\log \lambda}{d}}$). Each mean has been estimated from 10^5 samples. (On left) Average regret, normalized by d, on the sphere function for diverse population sizes λ at fixed dimension $d = 20$. The gain of rescaling decreases as λ increases. (On right) Distribution of the regret for the strategies on the $50d$-sphere function for $\lambda = 1000$.

H.3 Theoretical Results

We derive sufficient and necessary conditions on the scaling factor σ such that Eq. (H.1) can be satisfied. More precisely, we prove that Eq. (H.1) holds with approximation gain $\varepsilon \approx \log(\lambda)/d$ when the variance σ^2 is chosen proportionally to $\log \lambda/d$ (and λ does not grow too rapidly in d). We then show that Eq. (H.1) cannot be satisfied for $\sigma^2 = \omega(\log(\lambda)/d)$. Moreover, we prove that $\varepsilon = O(\log(\lambda)/d)$, which, together with the first result, shows that our scaling factor is asymptotically optimal. The precise statements are summarized in Theorems 26, 27, and 28, respectively. Proof sketches are available in Sec. H.3. Proofs are left in the full version available on the ArXiv version Meunier et al. [2020b].

Theorem 26 (Sufficient condition on rescaling)**.** *Let $\delta \in [\frac{1}{2}, 1)$. Let $\lambda = \lambda_d$, satisfying :*

$$\lambda_d \to \infty \text{ as } d \to \infty \text{ and } \log(\lambda_d) \in o(d) \tag{H.2}$$

. Then there exist two positive constants c_1, c_2, and d_0, such that for all $d \geq d_0$ it holds that

$$\mathbb{P}\left[\min_{i=1,\dots,\lambda} \|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2\right] \geq \delta \tag{H.3}$$

when x^* is sampled from the standard Gaussian distribution $\mathcal{N}(0, I_d)$, $x_1, \dots, x_\lambda$ are independently sampled from $\mathcal{N}(0, \sigma^2 I_d)$ with $\sigma^2 = \sigma_d^2 = c_2 \log(\lambda)/d$ and $\varepsilon = \varepsilon_d = c_1 \log(\lambda)/d$.

Theorem 26 shows that i.i.d. Gaussian sampling can outperform the middle point strategy derived in Cauwet et al. [2019] (i.e., the strategy using $\sigma^2 = 0$) if the scaling factor σ is chosen appropriately. Our next theorem summarizes our findings for the conditions that are *necessary* for the scaling factor σ^2 to outperform this middle point strategy. This result, in particular, illustrates why neither the natural choice $\sigma = 1$, nor any other constant scaling factor can be optimal.

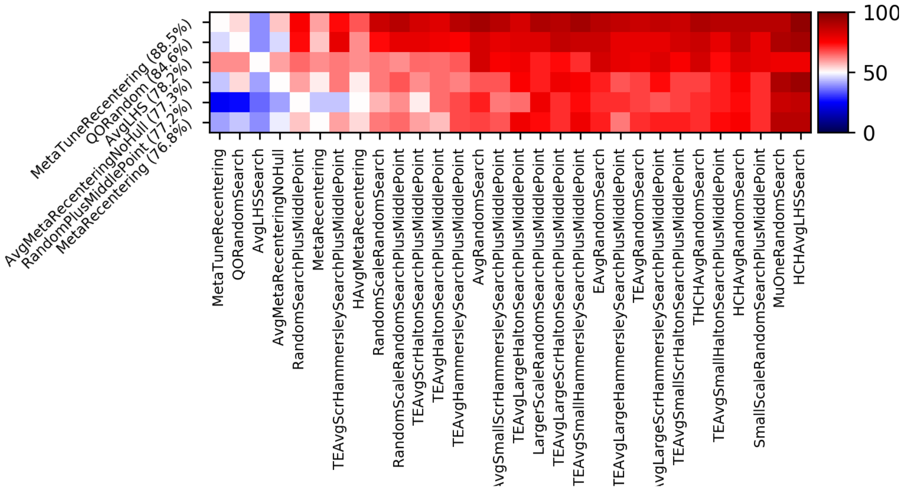


Figure H.3: Comparison of various one-shot optimization methods from the point of view of the simple regret. Reading guide in Sec. H.4.2. Results are averaged over objective functions Cigar, Rastrigin, Sphere in dimension 20, 200, 2000, and budget 30, 100, 3000, 10000, 30000, 100000. `MetaTuneRecentering` performs best overall. Only the 30 best performing methods are displayed as columns, and the 6 best as rows. Red means superior performance of row vs col. Rows and cols ranked by performance.

Theorem 27 (Necessary condition on rescaling)**.** *Consider $\lambda = \lambda_d$ satisfying assumptions (H.2). There exists an absolute constant $C > 0$ such that for all $\delta \in [\frac{1}{2}, 1)$, there exists $d_0 > 0$ such that, for all $d > d_0$ and for all σ the property*

$$\exists \varepsilon > 0, \mathbb{P}\left[\min_{i=1,\ldots,\lambda} \|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2\right] \geq \delta \tag{H.4}$$

for $x^* \sim \mathcal{N}(0, I_d)$ and $x_1, \ldots, x_\lambda$ independently sampled from $\mathcal{N}(0, \sigma^2 I_d)$, implies that $\sigma^2 \leq C\log(\lambda)/d$.

While Theorem 27 induces a necessary condition on the scaling factor σ to improve over the middle point strategy, it does not bound the gain that one can achieve through a proper scaling. Our next theorem shows that the factor derived in Theorem 26 is asymptotically optimal.

Theorem 28 (Upper bound for the approximation factor)**.** *Consider $\lambda = \lambda_d$ satisfying assumptions (H.2). There exists an absolute constant $C' > 0$ such that for all $\delta \in [\frac{1}{2}, 1)$, there exists $d_0 > 0$ such that, for all $d > d_0$ and for all $\varepsilon, \sigma > 0$, it holds that if $\mathbb{P}\left[\min_{i=1,\ldots,\lambda}\|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2\right] \geq \delta$ for $x^* \sim \mathcal{N}(0, I_d)$ and $x_1, \ldots, x_\lambda$ independently sampled from $\mathcal{N}(0, \sigma^2 I_d)$, then $\varepsilon \leq C'\log(\lambda)/d$.*

Proof Sketches. We first notice that as x^* is sampled from a standard normal distribution $\mathcal{N}(0, I_d)$, its norm satisfies $\|x^*\|^2 = d + o(d)$ as $d \to \infty$. We then use that, conditionally to x^*, it holds that

$$\mathbb{P}\left[\min_{i\in[\lambda]}\|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2 \middle| x^*\right] = 1 - \left(1 - \mathbb{P}\left[\|x - x^*\|^2 \leq (1-\varepsilon)\|x^*\|^2 \middle| x^*\right]\right)^\lambda$$

We therefore investigate when the condition

$$\mathbb{P}\big[\|x - x^*\|^2 \leq (1-\varepsilon)\|x^*\|^2 \big| x^*\big] > 1 - (1-\delta)^{\frac{1}{\lambda}} \tag{H.5}$$

is satisfied. To this end, we make use of the fact that the squared distance $\|x^*\|^2$ of x^* to the middle point 0 follows the central $\chi^2(d)$ distribution, whereas, for a given point $x^* \in \mathbb{R}^d$, the distribution of the squared distance $\|x - x^*\|^2/\sigma^2$ for $x \sim \mathcal{N}(0, \sigma^2 I_d)$ follows the non-central $\chi^2(d, \mu)$ distribution with non-centrality parameter $\mu := \|x^*\|^2/\sigma^2$. Using the concentration inequalities provided in [Zhang and Zhou, 2018, Theorem 7] for non-central χ^2 distributions, we then derive sufficient and necessary conditions for condition (H.5) to hold. With this, and using assumptions (H.2), we are able to derive the results from Theorems 26, 27, and 28.

H.4 Experimental Performance Comparisons

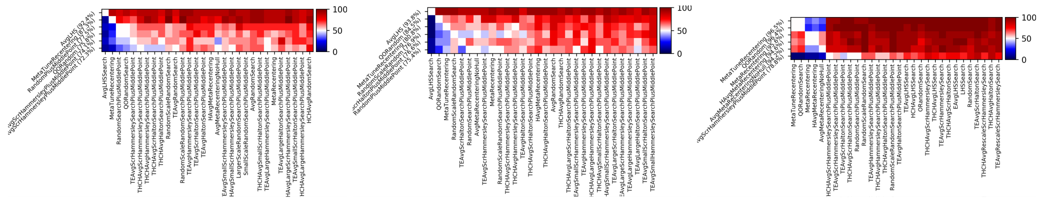
The theoretical results presented above are in asymptotic terms, and do not specify the constants. We therefore complement our mathematical investigation with an empirical analysis of the rescaling factor. Whereas results for the setting studied in Sec. H.3 are presented in Sec. H.4.1, we show in Sec. H.4.2 that the advantage of our rescaling factor is not limited to minimizing the distance in search space. More precisely, we show that the rescaled sampling achieves good results also in a classical DoE task, in which we aim for minimizing the regret for the Cigar and for the Rastrigin functions. Finally, we investigate in Sec. H.4.3 the impact of initializing two common optimization heuristics, Bayesian Optimization (BO) and differential evolution (DE), by a population sampled from the Gaussian distribution $\mathcal{N}(0, \sigma^2 I_d)$ using our rescaling factor $\sigma = \sqrt{\log(\lambda)/d}$.

H.4.1 Validation of Our Theoretical Results on the Sphere Function

Fig. H.1 displays the normalized average regret $\frac{1}{d}\mathbb{E}\big[\min_{i=1,\ldots,\lambda}\|x^* - x_i\|^2\big]$ in terms of $\sigma/\sqrt{\log(\lambda)/d}$ for different dimensions and budgets. We observe that the best parametrization of σ is around $\sqrt{\log(\lambda)/d}$ in all displayed cases. Moreover, we also see that – as expected – the gain of the rescaled sampling over the middle point sampling ($\sigma = 0$) goes to 0 as $d \to \infty$ (i.e. we get a result closer to the case $\sigma = 0$ as dimension goes to infinity). We also see that, for the regimes plotted in Fig. H.1, the advantage of the rescaled variance grows with the budget λ. Figure H.2 (on left) displays the average regret (average over multiple samplings and multiple positions of the optimum) as a function of increasing values of λ for the different rescaling methods ($\sigma \in \{0, \sqrt{\log\lambda/d}, 1\}$). We remark, unsurprisingly, that the gain of rescaling is diminishing as $\lambda \to \infty$. Finally, Figure H.2 (on right) shows the distribution of regrets for the different rescaling methods. The improvement of the expected regret is not at the expense of a higher dispersion of the regret.

H.4.2 Comparison with the DoEs Available in Nevergrad

Motivated by the significant improvements presented above, we now investigate whether the advantage of our rescaling factor translates to other optimization tasks. To this end, we first analyze a DoE setting, in which an underlying (and typically not explicitly given) function f is to be minimized through a parallel evaluation of λ solution candidates $x_1, \ldots, x_\lambda$, and regret is measured



Sphere function Cigar function Rastrigin function

Figure H.4: Same experiment as Fig. H.3, but separately over each objective function. Results are still averaged over 6 distinct budgets (30, 100, 3000, 10000, 30000, 100000) and 3 distinct dimensionalities (20, 200, 2000). `MetaTuneRecentering` performs well in each case, and is not limited to the sphere function for which it was derived. Variants of LHS are sometimes excellent and sometimes not visible at all (only the 30 best performing methods are shown).

in terms of $\min_i f(x_i) - \inf_x f(x)$. In the broader machine learning literature, and in particular in the context of hyper-parameter optimization, this setting is often referred to as *one-shot optimization* Bergstra and Bengio [2012], Cauwet et al. [2019].

Experimental Setup. All our experiments are implemented and freely available in the Nevergrad platform Rapin and Teytaud [2018]. Results are presented as shown in Fig. H.3. Typically, the six best methods are displayed as rows. The 30 best performing methods are presented as columns. The order for rows and for columns is the same: algorithms are ranked by their average winning frequency, measured against all other algorithms in the portfolio. The heatmaps show the fraction of runs in which algorithm x (row) outperformed algorithm y (column), averaged over all settings and all replicas (i.e. random repetitions). The settings are typically sweepings over various budgets, dimensions, and objective functions.[2]For each tested (algorithm, problem) pair, 20 independent runs are performed: a case with N settings is thus based on a total number of $20 \times N$ runs. The number N of distinct problems is at least 6 and often high in the dozens, hence the minimum number of independent runs is at least 120.

Algorithm Portfolio. Several rescaling methods are already available on Nevergrad. A large fraction of these have been implemented by the authors of Cauwet et al. [2019]; in particular:

- The replacement of one sample by the center. These methods are named "`midpointX`" or "`XPlusMiddlePoint`", where `X` is the original method that has been modified that way.
- The rescaling factor `MetaRecentering` derived in Cauwet et al. [2019]: $\sigma = \frac{1+\log(\lambda)}{4\log(d)}$.
- The quasi-opposite methods suggested in Rahnamayan and Wang [2009], with prefix "`QO`": when x is sampled, then another sample $c - rx$ is added, with r uniformly drawn in $[0, 1]$ and c the center of the distribution.

We also include in our comparison a different type of one-shot optimization techniques, independent of the present work, currently available in the platform: they use the information obtained

[2]Detailed results for individual settings are available at http://dl.fbaipublicfiles.com/nevergrad/allxps/list.html.

from the sampled points to recommend a point x that is not necessarily one of the λ evaluated ones. These *"one-shot+1"* strategies have the prefix "`Avg`". We keep all these and all other sampling strategies available in Nevergrad for our experiments. We add to this existing Nevergrad portfolio our own rescaling strategy, which uses the scaling factor derived in Sec. H.3; i.e., $\sigma = \sqrt{\log(\lambda)/d}$. We refer to this sampling strategy as `MetaTuneRecentering`, defined below. Both scaling factors `MetaRecentering` Cauwet et al. [2019] and `MetaTuneRecentering` (our equations) are applied to quasirandom sampling (more precisely, scrambled Hammersley Hammersley [1960], Atanassov [2004]) rather than random sampling. We provide detailed specifications of these methods and the most important ones below, whereas we skip the dozens of other methods: they are open sourced in Nevergrad Rapin and Teytaud [2018].

From $[0,1]^d$ to Gaussian quasi-random, random or LHS sampling: Random sampling, quasi-random sampling, Latin Hypercube Sampling (or others) have a well known definition in $[0,1]^d$ (for quasi-random, see Halton Halton [1960] or Hammersley Hammersley [1960], possibly boosted by scrambling Atanassov [2004]; for LHS, see McKay et al. [1979a]). To extend to multidimensional Gaussian sampling, we use that if U is a uniform random variable on $[0,1]$ and Φ the standard Gaussian CDF, then $\Phi^{-1}(U)$ simulates a $\mathcal{N}(0,1)$ distribution. We do so on each dimension: this provides a Gaussian quasi-random, random or LHS sampling.

Then, one can rescale the Gaussian quasi-random sampling with the corresponding factor σ for `MetaRecentering`($\sigma = \frac{1+\log(\lambda)}{4\log(d)}$ Cauwet et al. [2019]) and `MetaTuneRecentering`($\sigma = \sqrt{\log(\lambda)/d}$): for $i \leq \lambda$ and $j \leq d$, $x_{i,j} = \sigma\phi^{-1}(h_{i,j})$ where $h_{i,j}$ is the j^{th} coordinate of a i^{th} Scrambled-Hammersley point.

Results for the Full DoE Testbed in Nevergrad. Fig. H.3 displays aggregated results for the Sphere, the Cigar, and the Rastrigin functions, for three different dimensions and six different budgets. We observe that our `MetaTuneRecentering` strategy performs best, with a winning frequency of 80%. It positively compares against all other strategies from the portfolio, with the notable exception of `AvgLHS`, which, in fact, compares favorably against every single other strategy, but with a lower average winning frequency of 73.6%. Note here that `AvgLHS` is one of the **"oneshot+1"** strategies, i.e., it has not only one more sample, but it is also allowed to sample its recommendation adaptively, in contrast to our fully parallel `MetaTuneRecentering` strategy. It performs poorly in some cases (Rastrigin) and does not make sense as an initialization (Sect. H.4.3).

Selected DoE Tasks. Fig. H.4 breaks down the aggregated results from Fig. H.3 to the three different functions. We see that `MetaTuneRecentering` scores second on sphere (where `AvgLHS` is winning), third on Cigar (after `AvgLHS` and `QORandom`), and first on Rastrigin. This fine performance is remarkable, given that the portfolio contains quite sophisticated and highly tuned methods. In addition, the `AvgLHS` methods, sometimes performing better on the sphere, besides using more capabilities than we do (as it is a "oneshot+1" method), had poor results for Rastrigin (not even in the 30 best methods). On sphere, the difference to the third and following strategies is significant (87.3% winning rate against 77.5% for the next runner-up). On Cigar, the differences between the first four strategies are greater than 4 percentage points each, whereas on Rastrigin the average winning frequencies of the first five strategies is comparable, but significantly larger than that of the sixth one (which scores 78.8% against >94.2% for the first five DoEs). Fig. H.5 zooms into the results for the sphere function, and breaks them further down by available budget λ (note that the results are still averaged over the three tested dimensions). `MetaTuneRecentering` scores second in all six cases. A breakdown of the results for sphere by dimension (and aggregated over the six available budgets) is provided in Fig. H.6 and Fig. H.7. For dimension 20, we see that `MetaTuneRecentering` ranks third, but, interestingly, the two first methods are "oneshot+1" style (`Avg` prefix). In dimension 200, `MetaTuneRecentering` ranks second, with considerable advantage over the third-ranked strategy (88.0% vs. 80.8%). Finally, for the largest tested dimension, $d = 2000$, our method ranks first, with an average winning frequency of 90.5%.

H.4.3 Application to Iterative Optimization Heuristics

We now move from the one-shot settings considered thus far to *iterative optimization*, and show that our scaling factor can also be beneficial in this context. More precisely, we analyze the impact of initializing efficient global optimization (EGO Jones et al. [1998], a special case of Bayesian optimization) and differential evolution (DE Storn and Price [1997]) by a population that is sampled from a distribution that uses our variance scaling scheme. It is well known that a proper initialization can be very critical for the performance of these solvers; see Feurer et al. [2015], Surry and Radcliffe [1996], Rahnamayan and Wang [2009], Maaranen et al. [2004], Bossek et al. [2020] for discussions. Fig. H.8 summarizes the results of our experiments. As in the previous setups, we compare against existing methods from the Nevergrad platform, to which we have just added our rescaling factor termed `MetaTuneRecentering`. For each initialization scheme, four different initial population sizes are considered: denoting by d the dimension, by w the parallelism (i.e., the number of workers), and by b the total budget that the algorithms can spend on optimizing the given optimization task, the initial population λ is set as $\lambda = \sqrt{b}$ for `Sqrt`, as $\lambda = d$ for `Dim`, $\lambda = w$ for no suffix, and as $\lambda = 30$ when the suffix is `30`. As in Sec. H.4.2 we superpose our scaling scheme on top of the quasi-random Scrambled Hammersley sequence suggested in Cauwet et al. [2019], but we also consider random initialization rather than quasi-random (indicated by the suffix "`R`") and Latin Hypercube Sampling McKay et al. [1979a] (suffix "`LHS`"). The left chart in Fig. H.8 is for the Bayesian optimization case. It aggregates results for 48 settings, which stem from Nevergrad's "parahdbo4d" suite. It comprises the four benchmark problems Sphere, Cigar, Ellipsoid and Hm. Results are averaged over the total budgets $b \in \{25, 31, 37, 43, 50, 60\}$, dimension $d \in \{20, 2000\}$, and parallelism $w = \max(d, \lfloor b/6 \rfloor)$. We observe that a BO version using our `MetaTuneRecentering` performs best, and that several other variants using this scaling appear among the top-performing configurations. The chart on the right of Fig. H.8 summarizes results for Differential Evolution. Since DE can handle larger budgets, we consider here a total number of 100 settings, which correspond to the testcase named "paraalldes" in Nevergrad. In this suite, results are averaged over budgets $b \in \{10, 100, 1000, 10000, 100000\}$, dimensions $d \in \{5, 20, 100, 500, 2500\}$, parallelism $w = \max(d, \lfloor b/6 \rfloor)$, and again the objective functions Sphere, Cigar, Ellipsoid, and Hm. Specialized versions of DE perform best for this testcase, but we see that DE initialized with our `MetaTuneRecentering` strategy ranks fifth (outperformed only by ad hoc variants of DE), with an overall winning frequency that is not much smaller than that of the top-ranked `NoisyDE` strategy (76.3% for `ChainDEwithMetaTuneRecentering` vs. 81.7% for `NoisyDE`) - and almost always outperforms the rescaling used in the original Nevergrad.

H.5 Conclusions and Future Work

We have investigated the scaling of the variance of random sampling in order to minimize the expected regret. While previous work Cauwet et al. [2019] had already shown that, in the context of the sphere function, the optimal scaling factor is not identical to that of the prior distribution from which the optimum is sampled (unless the sample size is exponentially large in the dimension), it did not answer the question how to scale the variance optimally. We have proven that a standard deviation scaled as $\sigma = \sqrt{\log(\lambda)/d}$ gives, with probability at least $1/2$, a sample that is significantly closer to the optimum than the previous known strategies. We have also proven that the gain achieved by our scaling strategy is asymptotically optimal and that any decent scaling factor is asymptotically at most as large as our suggestion.
The empirical assessment of our rescaled sampling strategy confirmed decent performance not only on the sphere function, but also on other classical benchmark problems. We have furthermore given indications that the sampling might help improve state-of-the-art numerical heuristics based on differential evolution or using Bayesian surrogate models. Our proposed one-shot method performs best in many cases, sometimes outperformed by e.g. `AvgLHS`, but is stable on a wide range of problems and meaningful also as an initialization method (as opposed to `AvgLHS`). Whereas our theoretical results can be extended to quadratic forms (by conservation of barycenters through linear transformations), an extension to wider families of functions (e.g., families of functions with order 2 Taylor expansion) is not straightforward. Apart from extending our results to broader function classes, another direction for future work comprises extensions to the multi-epoch case. Our empirical results on DE and BO gives a first indication that a properly scaled variance can also be beneficial in iterative sampling. Note, however, that in the latter case, we only adjusted the initialization, not the later sampling steps. This forms another promising direction for future work.

H.6 Appendix: Relevant Concentration Bounds for χ^2 Distributions

We recall some basic definitions and properties of the central and the non-central χ^2 distributions, which are needed in the proofs of Theorems 26 and 27.

Definition 31. *(Central χ^2-distribution) Let $X_1, \ldots, X_d$ be d independent random variables drawn from the standard normal distribution $\mathcal{N}(0,1)$. Then the random variable $U = X_1^2 + \ldots + X_d^2$ follows a central $\chi^2(d)$ distribution with d degrees of freedom.*

As mentioned previously, the squared distance $\|x^*\|^2$ of x^* to the middle point 0 follows the central $\chi^2(d)$ distribution. This is thus also the distribution of the performance of the random sampling strategy using $\sigma^2 = 0$. In our proofs we will make use of the following properties of this distribution.

Property 1. *(Properties of χ^2 distribution) Let $U \sim \chi^2(d)$. Then $\mathbb{E}(U) = d$, $var(U) = 2d$, and for all $t \in [0,1]$ it holds that $\mathbb{P}\big[|\frac{U}{d} - 1| \geq t\big] \leq 2\exp\Big(-\frac{dt^2}{8}\Big)$.*

While the central χ^2 distribution suffices for the analysis of the middle point sampling strategy, *non-central χ^2 distribution* are required in the analysis of our Gaussian sampling with rescaled variance.

Definition 32. *(Non-central χ^2-distribution) Let $X_1, ..., X_d$ be independently drawn random variables satisfying $X_i \sim \mathcal{N}(\mu_i, 1)$. Let $U = X_1^2 + ... + X_d^2$. The random variable U follows a central $\chi^2(d, \mu)$ distribution with d degrees of freedom and non-centrality parameter $\mu = \sum_{i=1}^{d} \mu_i^2$.*

Note here that the non-central χ^2 distribution only depends on $\sum_{i=1}^{d} \mu_i^2$, but not on the individual values $(\mu_1, ..., \mu_d)$. Note further that, for a given point $x^* \in \mathbb{R}^d$, the distribution of the squared distance $\|x - x^*\|^2$ for $x \sim \mathcal{N}(0, I)$ follows the non-central $\chi^2(d, \mu)$ distribution with non-centrality parameter $\mu := \|x^*\|^2$.

We recall some important properties of the non-central χ^2 distribution.

Property 2. *(Properties of the non-central χ^2 distribution) Let $U \sim \chi^2(d, \mu)$. Then $\mathbb{E}(U) = d + \mu$, $var(U) = 2(d + 2\mu)$, and for any $\beta > 1$ there exist positive constants C_1, C_β such that for all $x \leq (\mu + d)/\beta$ it holds that*

$$P(U \leq -x) \geq C_1 \exp\left\{\left(-C_\beta \frac{x^2}{2\mu + d}\right)\right\}. \tag{H.6}$$

Moreover, for all $x > 0$, it holds that

$$P(U \leq -x) \leq \exp\left\{\left(-\frac{1}{4}\frac{x^2}{2\mu + d}\right)\right\}. \tag{H.7}$$

Proofs for the concentration inequalities H.6 and H.7 can be found in [Zhang and Zhou, 2018, Theorem 7].

H.7 Appendix: Proof of Theorem 26 (Sufficient condition)

Proof. We now present the proof of Theorem 26, the sufficient condition for the scaling factor σ^2 to be beneficial over sampling the middle point. Let δ, λ and d satisfy the conditions of Theorem 26. Let $\varepsilon, \sigma > 0$. By the law of total probability it holds that, for all $t \leq 1$,

$$\begin{aligned}
&\mathbb{P}\left[\min_{i\in[\lambda]} \|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2\right] \\
&= \mathbb{P}\left[\min_{i\in[\lambda]} \|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2 \mid |\frac{\|x^*\|^2}{d} - 1| \leq t\right] \mathbb{P}\left[|\frac{\|x^*\|^2}{d} - 1| \leq t\right] \\
&+ \mathbb{P}\left[\min_{i\in[\lambda]} \|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2 | |\frac{\|x^*\|^2}{d} - 1| > t\right] \mathbb{P}\left[|\frac{\|x^*\|^2}{d} - 1| > t\right].
\end{aligned}$$

Eq. H.3 is therefore satisfied if

$$\mathbb{P}\left[\min_{i\in[\lambda]}\|x^*-x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2 \Big| \left|\frac{\|x^*\|^2}{d}-1\right| \leq t\right]\mathbb{P}\left[\left|\frac{\|x^*\|^2}{d}-1\right| \leq t\right] \geq \delta.$$

This equation, in turn, is satisfied if for all y with $|\frac{\|y\|^2}{d}-1| \leq t$ it holds that

$$\mathbb{P}\left[\min_{i\in[\lambda]}\|x^*-x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2 \Big| x^*=y\right] \geq \frac{\delta}{\mathbb{P}\left[\left|\frac{\|x^*\|^2}{d}-1\right| \leq t\right]}. \tag{H.8}$$

For the following computations, we fix $t := d^{-1/3}$ and we set $\delta' := \delta/\mathbb{P}\left[\left|\frac{\|x^*\|^2}{d}-1\right| \leq t\right]$.

Let x^* be such that $|\frac{\|x^*\|^2}{d}-1| \leq t$. Then, conditionally to x^*, we have

$$\begin{aligned}
&\mathbb{P}\left[\min_{i\in[\lambda]}\|x^*-x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2 \Big| x^*\right]\\
&= 1-\mathbb{P}\left[\min_{i\in[\lambda]}\|x^*-x_i\|^2 \geq (1-\varepsilon)\|x^*\|^2 \Big| x^*\right]\\
&= 1-\mathbb{P}\left[\|x-x^*\|^2 \geq (1-\varepsilon)\|x^*\|^2 \big| x^*\right]^\lambda\\
&= 1-\left(1-\mathbb{P}\left[\|x-x^*\|^2 \leq (1-\varepsilon)\|x^*\|^2 \big| x^*\right]\right)^\lambda
\end{aligned}$$

for an x is distributed as a normal distribution $\mathcal{N}(0,\sigma^2 I)$. We recall that for such an x the distribution of the term $\|x-x^*\|^2/\sigma^2$ (for fixed x^*) follows the non-central $\chi^2(d,\mu)$ distribution with non-centrality parameter $\mu := \|x^*\|^2/\sigma^2$. We therefore obtain (through simple algebraic manipulations) that condition (H.8) holds if and only if

$$\mathbb{P}\left[U \leq (1-\varepsilon)\frac{\|x^*\|^2}{\sigma^2}\right] \geq 1-(1-\delta')^{1/\lambda},$$

with $U \sim \chi^2(d,\mu)$. Let $Y := U - \left(\frac{\|x^*\|^2}{\sigma^2}+d\right)$. Then the previous condition is equivalent to

$$\mathbb{P}\left[Y \leq -\left(\varepsilon\frac{\|x^*\|^2}{\sigma^2}+d\right)\right] \geq 1-(1-\delta)^{1/\lambda}.$$

According to the concentration inequality H.6, it holds that for any $\beta > 1$, there exist constants $C_1 > 0$ and $C_\beta > 0$ such that if

$$\varepsilon\frac{\|x^*\|^2}{\sigma^2}+d \leq \frac{1}{\beta}\left(\frac{\|x^*\|^2}{\sigma^2}+d\right), \tag{H.9}$$

then

$$\mathbb{P}\left(Y \leq -\left(\varepsilon\frac{\|x^*\|^2}{\sigma^2}+d\right)\right) \geq C_1\exp\left\{\left(-C_\beta\frac{(\varepsilon\frac{\|x^*\|^2}{\sigma^2}+d)^2}{2\frac{\|x^*\|^2}{\sigma^2}+d}\right)\right\}.$$

We deduce a sufficient condition for (H.8), by noting that it is satisfied if, for all x^* such that $|\frac{\|x^*\|^2}{d} - 1| \leq t$, it holds that

$$\frac{\left(\varepsilon\frac{\|x^*\|^2}{\sigma^2} + d\right)^2}{2\frac{\|x^*\|^2}{\sigma^2} + d} \leq A_\lambda, \tag{H.10}$$

with $A_\lambda := -\frac{1}{C_\beta}\big(\log\big(1 - (1-\delta')^{1/\lambda}\big) - \log C_1\big)$.

Let us now fix $\beta := 2$, $\varepsilon := c_1\frac{\log\lambda}{d}$ and $\sigma^2 := c_2\frac{\log\lambda}{d}$, with $c_1 := \frac{1}{3C_\beta}$ and $c_2 := c_1$. We show that, with these choices of β, ε and σ, inequalities (H.9) and H.10 are satisfied if d is sufficiently large and x^* satisfies $|\frac{\|x^*\|^2}{d} - 1| \leq t$. To this end, first note that

$$\frac{\varepsilon\frac{\|x^*\|^2}{\sigma^2} + d}{(\frac{\|x^*\|^2}{\sigma^2} + d)} \leq \frac{\frac{c_1}{c_2}(1+t) + 1}{\frac{d}{c_2\log\lambda}(1-t) + 1}.$$

Under the assumptions stated in (H.2) the term $\frac{\frac{c_1}{c_2}(1+t)+1}{\frac{d}{c_2\log\lambda}(1-t)+1}$ converges to zero as $d \to \infty$. We therefore obtain that, for d sufficiently large and x^* satisfying $|\frac{\|x^*\|^2}{d} - 1| \leq t$, it holds that

$$\frac{\varepsilon\frac{\|x^*\|^2}{\sigma^2} + d}{\frac{\|x^*\|^2}{\sigma^2} + d} \leq \frac{1}{\beta},$$

which proves (H.9).

To show (H.10), we first note that

$$\frac{(\varepsilon\frac{\|x^*\|^2}{\sigma^2} + d)^2}{2\frac{\|x^*\|^2}{\sigma^2} + d} \leq \frac{\left(\frac{c_1}{c_2}(1+t) + 1\right)^2}{2\frac{d}{c_2\log\lambda}(1-t) + 1}.$$

Under the assumptions stated in (H.2), and since $d \to \infty$, we approximate

$$\frac{\frac{c_1}{c_2}(1+t) + 1}{\frac{d}{c_2\log\lambda}(1-t) + 1} = \frac{c_2}{2}\left(\frac{c_1}{c_2} + 1\right)^2 \log\lambda + o(\log\lambda) = \frac{2}{3C_\beta}\log\lambda + o(\log\lambda)$$

and $A_\lambda = \frac{1}{C_\beta}\log\lambda + o(\log\lambda)$, which shows that condition H.10 holds for d sufficiently large and x^* satisfying $|\frac{\|x^*\|^2}{d} - 1| \leq t$. □

H.8 Appendix: Proof of Theorem 27 (Necessary condition)

Proof. We now prove the necessary condition which we have stated in Theorem 27. Let d, λ, ε, and σ satisfy the condition of Theorem 27. As in the beginning of the proof for Theorem 26, we can deduce the following necessary condition. For all $t \leq 1$ it holds that

$$\mathbb{P}\left[\min_{i\in[\lambda]}\|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2 \middle| |\frac{\|x^*\|^2}{d} - 1| \leq t\right]\mathbb{P}\left[|\frac{\|x^*\|^2}{d} - 1| \leq t\right] + \mathbb{P}\left[|\frac{\|x^*\|^2}{d} - 1| > t\right] \geq \delta$$

Then there exists x^* such that $|\frac{\|x^*\|^2}{d} - 1| \leq t$ and

$$\mathbb{P}\left[\min_{i\in[\lambda]}\|x^* - x_i\|^2 \leq (1-\varepsilon)\|x^*\|^2 \middle| x^*\right] \geq \frac{\delta - \mathbb{P}\left[|\frac{\|x^*\|^2}{d} - 1| > t\right]}{\mathbb{P}\left[|\frac{\|x^*\|^2}{d} - 1| \leq t\right]}. \tag{H.11}$$

Set $\delta' := \frac{\delta - \mathbb{P}\left[|\frac{\|x^*\|^2}{d} - 1| > t\right]}{\mathbb{P}\left[|\frac{\|x^*\|^2}{d} - 1| \leq t\right]}$. Then the necessary condition (H.11) can be written as

$$\mathbb{P}\left[Y \leq -\left(\varepsilon\frac{\|x^*\|^2}{\sigma^2} + d\right)\right] \geq 1 - (1-\delta')^{1/\lambda}$$

with $Y := U - (\frac{\|x^*\|^2}{\sigma^2} + d)$ and U being distributed according to a non-central χ^2 distribution with d degrees of freedom and non-centrality parameter $\|x^*\|^2/\sigma^2$. According to the concentration bound (H.7), we have

$$\mathbb{P}\left(Y \leq -\left(\varepsilon\frac{\|x^*\|^2}{\sigma^2} + d\right)\right) \leq \exp\left\{\left(-\frac{1}{4}\frac{(\varepsilon\frac{\|x^*\|^2}{\sigma^2} + d)^2}{2\frac{\|x^*\|^2}{\sigma^2} + d}\right)\right\}.$$

Condition (H.11) therefore requires

$$\exp\left\{\left(-\frac{1}{4}\frac{(\varepsilon\frac{\|x^*\|^2}{\sigma^2} + d)^2}{2\frac{\|x^*\|^2}{\sigma^2} + d}\right)\right\} \geq 1 - (1-\delta')^{1/\lambda}.$$

From this we derive $\varepsilon \leq \left(\sqrt{\tilde{A}_\lambda\left(2\frac{\|x^*\|^2}{\sigma^2} + d\right)} - d\right)\frac{\sigma^2}{\|x^*\|^2}$, with $\tilde{A}_\lambda = -4\log\left(1 - (1-\delta')^{1/\lambda}\right)$. As $\varepsilon > 0$, we obtain that

$$\sigma^2 < \tilde{\sigma}^2 := 2\frac{\|x^*\|^2/d}{\frac{d}{\tilde{A}_\lambda} - 1}.$$

Fixing $t = d^{-1/3}$ and considering the requirements stated in (H.2) we obtain that $\tilde{\sigma} = 2\frac{\tilde{A}_\lambda}{d} + o\left(\frac{\tilde{A}_\lambda}{d}\right) = 8\frac{\log \lambda}{d} + o\left(\frac{\log \lambda}{d}\right)$, which concludes the proof of the necessary condition, as it shows $\sigma^2 \in O\left(\frac{\log \lambda_d}{d}\right)$. □

H.9 Appendix: Proof of Theorem 28 (Upper Bound for the Gain)

Proof. The proof of Theorem 28 uses the same argument as the one of Theorem 27. We have proved that σ^2 must be between 0 and $\tilde{\sigma} = 2\frac{\|x^*\|^2/d}{\frac{d}{\tilde{A}_\lambda}-1}$. Then we get that:

$$\varepsilon \leq \sup_{\sigma\in[0,\tilde{\sigma}]} \left(\sqrt{\tilde{A}_\lambda\left(2\frac{\|x^*\|^2}{\sigma^2}+d\right)}-d\right)\frac{\sigma^2}{\|x^*\|^2}.$$

Noticing that:

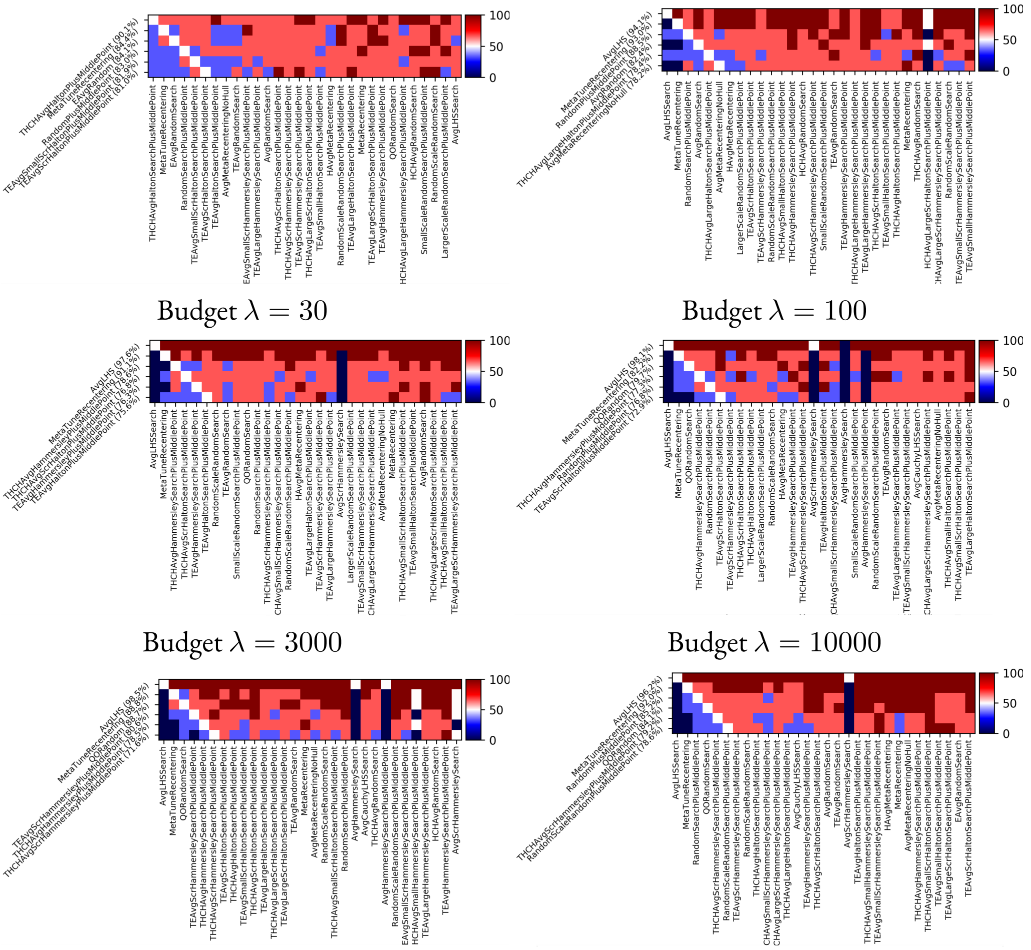
$$\begin{aligned}
&\sup_{\sigma\in[0,\tilde{\sigma}]} \left(\sqrt{\tilde{A}_\lambda\left(2\frac{\|x^*\|^2}{\sigma^2}+d\right)}-d\right)\frac{\sigma^2}{\|x^*\|^2} \\
&\qquad = \sup_{\alpha\in[0,1]} \left(\sqrt{\tilde{A}_\lambda\left(2\frac{\|x^*\|^2}{\alpha\tilde{\sigma}^2}+d\right)}-d\right)\frac{\alpha\tilde{\sigma}^2}{\|x^*\|^2}
\end{aligned}$$

We get after simple algebraic simplifications and for d sufficiently large under assumptions (H.2):

$$\begin{aligned}
&\sup_{\sigma\in[0,\tilde{\sigma}]} \left(\sqrt{\tilde{A}_\lambda\left(2\frac{\|x^*\|^2}{\sigma^2}+d\right)}-d\right)\frac{\sigma^2}{\|x^*\|^2} \\
&\qquad \leq \frac{d\tilde{\sigma}^2}{\|x^*\|^2}\sup_{\alpha\in[0,1]} \alpha\left(\sqrt{\alpha^{-1}+\frac{\tilde{A}_\lambda}{d^2}}-1\right) \\
&\qquad \leq \frac{d\tilde{\sigma}^2}{\|x^*\|^2}\sup_{\alpha\in[0,1]} \alpha\left(\sqrt{\alpha^{-1}+1}-1\right) \\
&\qquad \leq 8\frac{\log \lambda}{d} + o\left(\frac{\log \lambda}{d}\right)
\end{aligned}$$

Then $\varepsilon \in O\left(\frac{\log \lambda_d}{d}\right)$, which concludes the proof of Theorem 28.

□

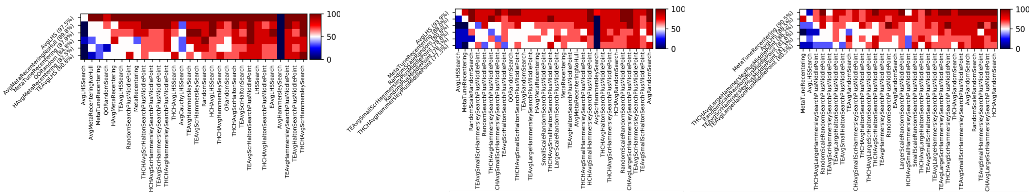


Budget $\lambda = 30$ Budget $\lambda = 100$

Budget $\lambda = 3000$ Budget $\lambda = 10000$

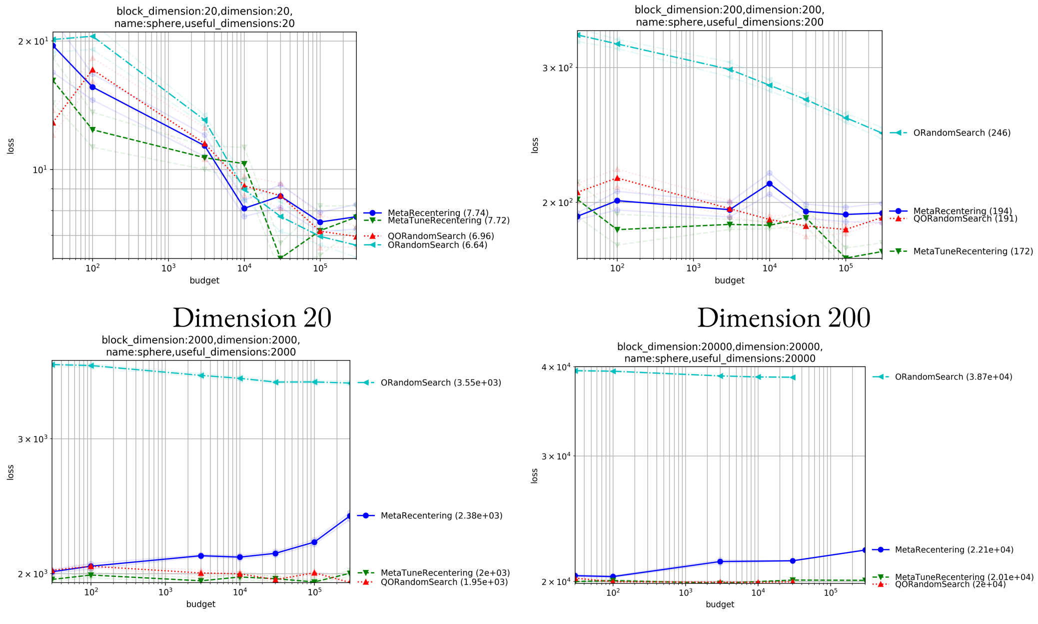
Budget $\lambda = 30000$ Budget $\lambda = 100000$

Figure H.5: Methods ranked by performance on the sphere function, per budget. Results averaged over dimension 20, 200, 2000. `MetaTuneRecentering` performs among the best in all cases. LHS is excellent on this very simple setting, namely the sphere function.



Dimension 20 Dimension 200 Dimension 2000

Figure H.6: Results on the sphere function, per dimensionality. Results are averaged over 6 values of the budget: 30, 100, 3000, 10000, 30000, 100000. Our method becomes better and better as the dimension increases.



Dimension 20 Dimension 200

Dimension 2000 Dimension 20000

Figure H.7: Same context as Fig. H.6, with x-axis = budget and y-axis = average simple regret. We see the failure of `MetaRecentering` in the worsening performance as budget goes to infinity: the budget has an impact on σ which becomes worse, hence worse overall performance. We note that quasi-opposite sampling can perform decently in a wide range of values. Opposite Sampling is not much better than random search in high-dimension. Our `MetaTuneRecentering` shows decent performance: in particular, simple regret decreases as $\lambda \to \infty$.

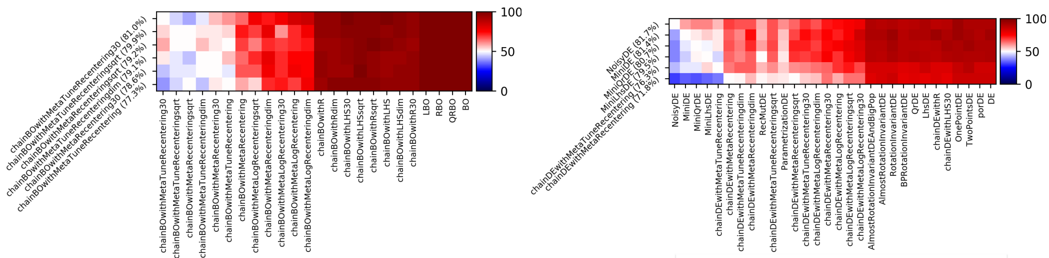


Figure H.8: Performance comparison of different strategies to initialize Bayesian Optimization (BO, left) and Differential Evolution (DE, right). A detailed description is given in Sec. H.4.3. `MetaTuneRecentering` performs best as an initialization method. In the case of DE, methods different from the traditional DE remain the best on this testcase: when we compare DE with a given initialization and DE initialized with `MetaTuneRecentering`, `MetaTuneRecentering` performs best in almost all cases.

I On averaging the best samples in evolutionary computation: the sphere function case

Choosing the right selection rate is a long standing issue in evolutionary computation. In the continuous unconstrained case, we prove mathematically that a single parent $\mu = 1$ leads to a sub-optimal simple regret in the case of the sphere function. We provide a theoretically-based selection rate μ/λ that leads to better progress rates. With our choice of selection rate, we get a provable regret of order $O(\lambda^{-1})$ which has to be compared with $O(\lambda^{-2/d})$ in the case where $\mu = 1$. We complete our study with experiments to confirm our theoretical claims.

I.1 Introduction

In evolutionary computation, the selected population size often depends linearly on the total population size, with a ratio between $1/4$ and $1/2$: 0.270 is proposed in Beyer and Schwefel [2002a], Hansen and Ostermeier [2003], Beyer and Sendhoff [2008] suggest $1/4$ and $1/2$. However, some sources Escalante and Reyes [2013] recommend a lower value $1/7$. Experimental results in Teytaud [2010] and theory in Fournier and Teytaud [2010] together suggest a ratio $\min(d, \lambda/4)$ with d the dimension, i.e. keep a population size at most the dimension. Jebalia and Auger [2010] suggests to keep increasing μ besides that limit, but slowly enough so that that rule $\mu = \min(d, \lambda/4)$ would be still nearly optimal. Weighted recombination is common **?**, but not with a clear gap when compared to truncation ratios **?**, except in the case of large population size **?**. There is, overall, limited theory around the optimal choice of μ for optimization in the continuous setting. In the present paper, we focus on a simple case (sphere function and single epoch), but prove exact theorems. We point out that the single epoch case is important by itself - this is fully parallel optimization Niederreiter [1992], McKay et al. [1979a], **?**], Bousquet et al. [2017]. Experimental results with a publicly available platform support the approach.

I.2 Theory

We consider the case of a single batch of evaluated points. We generate λ points according to some probability distribution. We then select the μ best and average them. The result is our approximation of the optimum. This is therefore an extreme case of evolutionary algorithm, with a single population; this is commonly used for e.g. hyperparameter search in machine learning Bergstra and Bengio [2012], Bousquet et al. [2017], though in most cases with the simplest case $\mu = 1$.

I.2.1 Outline

We consider the optimization of the simple function $x \mapsto \|x-y\|^2$ for an unknown $y \in \mathcal{B}(0,r)$. In Section I.2.2 we introduce notations. In Section I.2.3 we analyze the case of random search uniformly in a ball of radius h centered on y. We can, therefore, exploit the knowledge of the optimum's position and assume that $y = 0$. We then extend the results to random search in a ball of radius r centered on 0, provided that $r > \|y\|$ and show that results are essentially the same up to an exponentially decreasing term (Section I.2.4).

I.2.2 Notations

We are interested in minimizing the function $f : x \in \mathbb{R}^d \mapsto \|x-y\|^2$ for a fixed unknown y in parallel one-shot black box optimization, i.e. we sample λ points $X_1, \dots, X_\lambda$ from some distribution $\mathcal{D}$ and we search for $x^\star = \arg\min_x f(x)$. In what follows we will study the sampling from $\mathcal{B}(0,r)$, the uniform distribution on the ℓ_2-ball of radius r; w.l.o.g. $\mathcal{B}(y,r)$ will also denote the ℓ_2-ball centered in y and of radius r.
We are interested in comparing the strategy "μ-best" vs "1-best". We denote $X_{(1)}, \dots, X_{(\lambda)}$, the sorted values of X_i i.e. $(1),\dots,(\lambda)$ are such that $f(X_{(1)}) \leq \dots \leq f(X_{(\lambda)})$. The "$\mu$-best" strategy is to return $\bar{X}_{(\mu)} = \frac{1}{\mu}\sum_{i=1}^{\mu} X_{(i)}$ as an estimate of the optimum and the "1-best" is to return $X_{(1)}$. We will hence compare : $\mathbb{E}\left[f\left(\bar{X}_{(\mu)}\right)\right]$ and $\mathbb{E}\left[f\left(X_{(1)}\right)\right]$. We recall the definition of the gamma function Γ: $\forall z > 0, \Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$, as well as the property $\Gamma(z+1) = z\Gamma(z)$.

I.2.3 When the center of the distribution is also the optimum

In this section we assume that $y = 0$ (i.e. $f(x) = \|x\|^2$) and consider sampling in $\mathcal{B}(0,r) \subset \mathbb{R}^d$. In this simple case, we show that keeping the best $\mu > 1$ sampled points is asymptotically a better strategy than selecting a single best point. The choice of μ will be discussed in Section I.2.4.

Theorem 29. *For all $\lambda > \mu \geq 2$ and $d \geq 2$, $r > 0$, for $f(x) = \|x\|^2$,*

$$\mathbb{E}_{X_1,\dots,X_\lambda \sim \mathcal{B}(0,r)}\left[f\left(\bar{X}_{(\mu)}\right)\right] < \mathbb{E}_{X_1,\dots,X_\lambda \sim \mathcal{B}(0,r)}\left[f\left(X_{(1)}\right)\right].$$

To prove this result, we will compute the value of $\mathbb{E}\left[f\left(\bar{X}_{(\mu)}\right)\right]$ for all λ and μ. The following lemma gives a simple way of computing the expectation of a function depending only on the norm of its argument.

Lemma 14. *Let $d \in \mathbb{N}^*$. Let X be drawn uniformly in $\mathcal{B}(0,r)$ the d-dimensional ball of radius r. Then for any measurable function $g : \mathbb{R} \to \mathbb{R}$, we have*

$$\mathbb{E}_{X \sim \mathcal{B}(0,r)}[g(\|X\|)] = \frac{d}{r^d}\int_0^r g(\alpha)\alpha^{d-1}d\alpha.$$

In particular, we have $\mathbb{E}_{X \sim \mathcal{B}(0,r)}\left[\|X\|^2\right] = \frac{d}{d+2} \times r^2$.

Proof. Let $V(r,d)$ be the volume of a ball of radius r in $\mathbb{R}^d$ and $S(r,d)$ be the surface of a sphere of radius r in $\mathbb{R}^d$. Then $\forall r > 0$, $V(r,d) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} r^d$ and $S(r,d-1) = \frac{2\pi^{d/2}}{\Gamma(\frac{d}{2})} r^{d-1}$. Let $g : \mathbb{R} \to \mathbb{R}$ be a continuous function. Then:

$$
\begin{aligned}
\mathbb{E}_{X\sim\mathcal{B}(0,r)}[g(\|X\|)] &= \frac{1}{V(r,d)} \int_{x:\|x\|\le r} g(\|x\|)dx \\
&= \frac{1}{V(r,d)} \int_{\alpha=0}^{r} \int_{\theta:\|\theta\|=\alpha} g(\alpha)d\theta d\alpha \\
&= \frac{1}{V(r,d)} \int_{\alpha=0}^{r} g(\alpha)S(\alpha,d-1)d\alpha \\
&= \frac{S(1,d-1)}{V(r,d)} \int_{\alpha=0}^{r} g(\alpha)\alpha^{d-1}d\alpha = \frac{d}{r^d}\int_{\alpha=0}^{r} g(\alpha)\alpha^{d-1}d\alpha. \\
\text{So, } \mathbb{E}_{X\sim\mathcal{B}(r)}\left[\|X\|^2\right] &= \frac{d}{r^d}\int_{\alpha=0}^{r} \alpha^2\alpha^{d-1}d\alpha \\
&= \frac{d}{r^d}\left[\frac{\alpha^{d+2}}{d+2}\right]_0^r = \frac{d}{d+2}r^2.
\end{aligned}
$$

□

We now use the previous lemma to compute the expected regretBubeck et al. [2009] of the average of the μ best points conditionally to the value of $f(X_{(\mu+1)})$. The trick of the proof is that, conditionally to $f(X_{(\mu+1)})$, the order of $X_{(1)}, \ldots, X_{(\mu)}$ has no influence over the average. Computing the expected regret conditionally to $f(X_{(\mu+1)})$ thus becomes straightforward.

Lemma 15. *For all $d > 0$, $r^2 > h > 0$ and $\lambda > \mu \geq 1$, for $f(x) = \|x\|^2$,*

$$\mathbb{E}_{X_1,\ldots,X_\lambda\sim\mathcal{B}(y,r)}\left[f\left(\bar{X}_{(\mu)}\right) \mid f(X_{(\mu+1)}) = h\right] = \frac{h}{\mu} \times \frac{d}{d+2}.$$

Proof. Let us first compute $\mathbb{E}\left[f\left(\bar{X}_{(\mu)}\right) \mid f(X_{(\mu+1)}) = h\right]$. Note that for any function $g : \mathbb{R}^d \to \mathbb{R}$ and distribution $\mathcal{D}$, we have

$$
\begin{aligned}
&\mathbb{E}_{X_1\ldots X_\lambda\sim\mathcal{D}}\left[g(\bar{X}_{(\mu)}) \mid f(X_{(\mu+1)}) = h\right] \\
&\qquad = \mathbb{E}_{X_1\ldots X_\mu\sim\mathcal{D}}\left[g\left(\frac{1}{\mu}\sum_{i=1}^{\mu} X_i\right) \mid X_1\ldots X_\mu \in \{x : f(x) \le h\}\right] \\
&\qquad = \mathbb{E}_{X_1\ldots X_\mu\sim\mathcal{D}_h}\left[g\left(\frac{1}{\mu}\sum_{i=1}^{\mu} X_i\right)\right],
\end{aligned}
$$

where $\mathcal{D}_h$ is the restriction of $\mathcal{D}$ to the level set $\{x : f(x) \leq h\}$. In our setting, we have $\mathcal{D} = \mathcal{B}(0, r)$ and $\mathcal{D}_h = \mathcal{B}(0, \sqrt{h})$. Therefore,

$$\begin{aligned}
&\mathbb{E}_{X_1,\dots,X_\lambda \sim \mathcal{B}(0,r)}\left[f\left(\bar{X}_{(\mu)}\right) \mid f(X_{(\mu+1)}) = h\right] \\
&\qquad = \mathbb{E}_{X_1,\dots,X_\lambda \sim \mathcal{B}(0,r)}\left[\|\bar{X}_{(\mu)}\|^2 \mid f(X_{(\mu+1)}) = h\right] \\
&\qquad = \mathbb{E}_{X_1\dots X_\mu \sim \mathcal{B}(0,\sqrt{h})}\left[\|\frac{1}{\mu}\sum_{i=1}^{\mu} X_i\|^2\right] \\
&\qquad = \frac{1}{\mu^2}\mathbb{E}_{X_1\dots X_\mu \sim \mathcal{B}(0,\sqrt{h})}\left[\sum_{i,j=1}^{\mu} X_i^T X_j\right] \\
&\qquad = \frac{1}{\mu^2}\sum_{i,j=1,i\neq j}^{\mu} \mathbb{E}_{X_i\dots X_j \sim \mathcal{B}(0,\sqrt{h})}\left[X_i^T X_j\right] \\
&\qquad + \frac{1}{\mu^2}\sum_{i=1}^{\mu}\mathbb{E}_{X_i \sim \mathcal{B}(0,\sqrt{h})}\left[\|X_i\|^2\right] = \frac{1}{\mu}\mathbb{E}_{X \sim \mathcal{B}(0,\sqrt{h})}\left[\|X\|^2\right].
\end{aligned}$$

By Lemma 14, we have: $\mathbb{E}_{X \sim \mathcal{B}(0,\sqrt{h})}\left[\|X\|^2\right] = \frac{d}{d+2}h$. Hence $\mathbb{E}_{X_1,\dots,X_\lambda \sim \mathcal{B}(0,r)}\left[f\left(\bar{X}_{(\mu)}\right) \mid f(X_{(\mu+1)}) = h\right] = \frac{d}{d+2}\frac{h}{\mu}$. □

The result of Lemma 15 shows that $\mathbb{E}\left[f\left(\bar{X}_{(\mu)}\right) \mid f(X_{(\mu+1)}) = h\right]$ depends linearly on h. We now establish a similar dependency for $\mathbb{E}\left[f\left(X_{(1)}\right) \mid f(X_{(\mu+1)}) = h\right]$.

Lemma 16. *For* $d > 0,\ h > 0,\ \lambda > \mu \geq 1$, *and* $f(x) = \|x\|^2$,

$$\mathbb{E}_{X_1,\dots,X_\lambda \sim \mathcal{B}(0,r)}\left[f\left(X_{(1)}\right) \mid f(X_{(\mu+1)}) = h\right] = h\frac{\Gamma(\frac{d+2}{d})\Gamma(\mu+1)}{\Gamma(\mu+1+2/d)}.$$

Proof. First note that using the same argument as in Lemma 15, $\forall \beta \in (0, h]$:

$$\begin{aligned}
&\mathbb{P}_{X_1\dots X_\lambda \sim \mathcal{B}(0,\sqrt{h})}\left[f\left(X_{(1)}\right) > \beta \mid f(X_{(\mu+1)}) = h\right] \\
&\qquad = \mathbb{P}_{X_1\dots X_\mu \sim \mathcal{B}(0,\sqrt{h})}[f(X_1) > \beta, \dots, f(X_\mu) > \beta] \\
&\qquad = \mathbb{P}_{X \sim \mathcal{B}(0,\sqrt{h})}[f(X) > \beta]^\mu.
\end{aligned}$$

Recall that the volume of a d-dimensional ball of radius r is proportional to r^d. Thus, we get:

$$\mathbb{P}_{X \sim \mathcal{B}(0,\sqrt{h})}[f(X) < \beta] = \frac{\sqrt{\beta}^d}{\sqrt{h}^d} = \left(\frac{\beta}{h}\right)^{\frac{d}{2}}.$$

It is known that for every positive random variable $X, \mathbb{E}(X) = \int_0^\infty \mathbb{P}(X > \beta) d\beta$. Therefore:

$$\begin{aligned}
\mathbb{E}_S\left[f\big(X_{(1)}\big) \mid f(X_{(\mu+1)}) = h\right] &= \int_0^h \mathbb{P}\left[f\big(X_{(1)}\big) > \beta \mid f(X_{(\mu+1)}) = h\right] d\beta \\
&= \int_0^h \left(1 - \left(\frac{\beta}{h}\right)^{\frac{d}{2}}\right)^{\mu} d\beta \\
&= h \int_0^1 \left(1 - u^{\frac{d}{2}}\right)^{\mu} du \\
&= h \frac{2}{d} \int_0^1 (1-t)^{\mu} t^{2/d-1} dt = h \frac{\Gamma(\frac{d+2}{d})\Gamma(\mu+1)}{\Gamma(\mu+1+2/d)}.
\end{aligned}$$

To obtain the last equality, we identify the integral with the beta function of parameters $\mu+1$ and $\frac{2}{d}$. □

We now directly compute $\mathbb{E}_{X_1,\ldots,X_\lambda \sim \mathcal{B}(0,r)}\left[f(X_{(1)})\right]$.

Lemma 17. *For all* $d > 0$, $\lambda > 0$ *and* $r > 0$*:*

$$\mathbb{E}_{X_1,\ldots,X_\lambda \sim \mathcal{B}(0,r)}\left[f(X_{(1)})\right] = r^2 \frac{\Gamma(\frac{d+2}{d})\Gamma(\lambda+1)}{\Gamma(\lambda+1+2/d)}.$$

Proof. As in Lemma 16, we have for any $\beta \in (0, r^2]$:

$$\begin{aligned}
\mathbb{P}_{X_1 \ldots X_\lambda \sim \mathcal{B}(0,r)}\left[f\big(X_{(1)}\big) > \beta\right] &= \mathbb{P}_{X_1 \ldots X_\lambda \sim \mathcal{B}(0,r)}[f(X_1) > \beta, \ldots, f(X_\lambda) > \beta] \\
&= \mathbb{P}_{X \sim \mathcal{B}(0,r)}[f(X) > \beta]^{\lambda} \\
&= \left(\frac{\sqrt{\beta}}{r}\right)^{d}.
\end{aligned}$$

The result then follows by reasoning as in the proof of Lemma 16. □

By combining the results above, we obtain the exact formula for $\mathbb{E}\left[f(\bar{X}_{(\mu)})\right]$.

Theorem 30. *For all* $d > 0$, $r > 0$ *and* $\lambda > \mu \geq 1$*:*

$$\mathbb{E}_{X_1 \ldots X_\lambda \sim \mathcal{B}(0,r)}\left[f(\bar{X}_{(\mu)})\right] = \frac{r^2 d \times \Gamma(\lambda+1)\Gamma(\mu+1+2/d)}{\mu(d+2)\Gamma(\mu+1)\Gamma(\lambda+1+2/d)}.$$

Proof. The proof follows by applying our various lemmas and integrating over all possible values for h. We have:

$$\mathbb{E}_{X_1 \ldots X_\lambda \sim \mathcal{B}(0,r)}\left[f(\bar{X}_{(\mu)})\right]$$

$$
\begin{aligned}
&= \mathbb{E}\big[\mathbb{E}\big[f(\bar{X}_{(\mu)}) \mid f\big(X_{(\mu+1)}\big)\big]\big] \\
&= \frac{1}{\mu}\frac{d}{d+2}\mathbb{E}\big[f\big(X_{(\mu+1)}\big)\big] \text{ by Lemma 15} \\
&= \frac{1}{\mu}\frac{d}{d+2}\frac{\Gamma(\mu+1+2/d)}{\Gamma(\mu+1)\Gamma(\frac{d+2}{d})}\mathbb{E}\big[\mathbb{E}\big[f(X_{(1)}) \mid f\big(X_{(\mu+1)}\big)\big]\big] \text{ by Lemma 16} \\
&= \frac{1}{\mu}\frac{d}{d+2}\frac{\Gamma(\mu+1+2/d)}{\Gamma(\mu+1)\Gamma(\frac{d+2}{d})}\mathbb{E}\big[f(X_{(1)})\big] \\
&= \frac{r^2 d \times \Gamma(\lambda+1)\Gamma(\mu+1+2/d)}{\mu(d+2)\Gamma(\mu+1)\Gamma(\lambda+1+2/d)} \text{ by Lemma 17.}
\end{aligned}
$$

□

We have checked experimentally the result of Theorem 31 (see Figure I.1): the result of Theorem 29 follows from Theorem 31 since for $d \geq 2$, λ and r fixed, $\mathbb{E}\big[f(\bar{X}_{(\mu)})\big]$ is strictly decreasing in μ. In addition, we can obtain asymptotic progress rates:

Corollary 5. *Consider $d > 0$. When $\lambda \to \infty$, we have*

$$\mathbb{E}_{X_1 \dots X_\lambda \sim \mathcal{B}(0,r)}\big[f(\bar{X}_{(\mu)})\big] \sim \lambda^{-\frac{2}{d}}\frac{r^2 d \times \Gamma(\mu+1+2/d)}{\mu(d+2)\Gamma(\mu+1)},$$

while if $\lambda \to \infty$ and $\mu(\lambda) \to \infty$, $\mathbb{E}_{X_1 \dots X_\lambda \sim \mathcal{B}(0,r)}\big[f(\bar{X}_{(\mu(\lambda))})\big] \sim r^2 \frac{d}{d+2}\frac{\mu(\lambda)^{\frac{2}{d}-1}}{\lambda^{\frac{2}{d}}}.$

As a result, $\forall c \in (0,1)$, $\mathbb{E}\big(f(\bar{X}_{(\lfloor c\lambda \rfloor)})\big) \in \Theta\big(\frac{1}{\lambda}\big)$ and $\mathbb{E}\big(f(X_{(1)})\big) \in \Theta\Big(\frac{1}{\lambda^{2/d}}\Big)$.

Proof. We recall the Stirling equivalent formula for the gamma function: when $z \to \infty$,

$$\Gamma(z) = \sqrt{\frac{2\pi}{z}}\left(\frac{z}{e}\right)^z\left(1 + O\left(\frac{1}{z}\right)\right).$$

Using this approximation, we get the expected results. □

This result shows that by keeping a single parent, we lose more than a constant factor: the progress rate is significantly impacted. Therefore it is preferable to use more than one parent.

I.2.4 Convergence when the sampling is not centered on the optimum

So far we treated the case where the center of the distribution and the optimum are the same. We now assume that we sample from the distribution $\mathcal{B}(0,r)$ and that the function f is $f(x) = \|x-y\|^2$ with $\|y\| \leq r$. We define $\epsilon = \frac{\|y\|}{r}$.

Lemma 18. *Let $r > 0$, $d > 0$, $\lambda > \mu \geq 1$, we have:*

$$\mathbb{P}_{X_1 \dots X_\lambda \sim \mathcal{B}(0,r)}\big(f(X_{(\mu+1)}) > (1-\epsilon)^2 r^2\big) = \mathbb{P}_{U \sim B\left(\lambda,(1-\epsilon)^d\right)}(U \leq \mu),$$

where $B(\lambda, p)$ is a binomial law of parameters λ and p.

Proof. We have $f(X_{(\mu+1)}) > (1-\epsilon)r \iff \sum_{i=1}^{\lambda} \mathbb{1}_{\{f(X_i)\leq(1-\epsilon)^2r^2\}} \leq \mu$ since $\mathbb{1}_{\{f(X_i)\leq(1-\epsilon)^2r^2\}}$ are independent Bernoulli variables of parameter $(1-\epsilon)^d$, hence the result. □

Using Lemma 18, we now get lower and upper bounds on $\mathbb{E}\left[f\left(X_{(\mu+1)}\right)\right]$:

Theorem 31. *Consider $d > 0$, $r > 0$, $\lambda > \mu \geq 1$. The expected value of $f(\bar{X}_{(\mu)})$ satisfies both*

$$\begin{aligned}\mathbb{E}_{X_1\ldots X_\lambda\sim\mathcal{B}(0,r)}\left[f(\bar{X}_{(\mu)})\right] \leq & 4r^2\mathbb{P}_{U\sim B\left(\lambda,(1-\epsilon)^d\right)}(U\leq\mu) \\ & + \frac{r^2d\times\Gamma(\lambda+1)\Gamma(\mu+1+2/d)}{\mu(d+2)\Gamma(\mu+1)\Gamma(\lambda+1+2/d)}\end{aligned}$$

$$\text{and}\quad \mathbb{E}_{X_1\ldots X_\lambda\sim\mathcal{B}(0,r)}\left[f(\bar{X}_{(\mu)})\right] \geq \frac{r^2d\times\Gamma(\lambda+1)\Gamma(\mu+1+2/d)}{\mu(d+2)\Gamma(\mu+1)\Gamma(\lambda+1+2/d)}.$$

Proof.

$$\begin{aligned}\mathbb{E}\left[f(\bar{X}_{(\mu)})\right] = & \mathbb{E}\left(f(\bar{X}_{(\mu)})|f(X_{(\mu+1)})\geq(1-\epsilon)^2r^2\right)\mathbb{P}\left(f(X_{(\mu+1)})\geq(1-\epsilon)^2r^2\right) \\ & + \mathbb{E}\left(f(\bar{X}_{(\mu)})|f(X_{(\mu+1)})<(1-\epsilon)^2r^2\right)\mathbb{P}\left(f(X_{(\mu+1)})<(1-\epsilon)^2r^2\right).\end{aligned}$$

In this Bayes decomposition, we can bound the various terms as follows:

$$\begin{aligned}\mathbb{E}\left(f(\bar{X}_{(\mu)})|f(X_{(\mu+1)})\geq(1-\epsilon)^2r^2\right) & \leq 4r^2, \\ \mathbb{P}\left(f(X_{(\mu+1)})\geq(1-\epsilon)^2r^2\right) & \leq 1, \\ \mathbb{E}\left[f(\bar{X}_{(\mu)})|f(X_{(\mu+1)})<(1-\epsilon)^2r^2\right] & \leq \frac{r^2d\times\Gamma(\lambda+1)\Gamma(\mu+1+2/d)}{\mu(d+2)\Gamma(\mu+1)\Gamma(\lambda+1+2/d)}.\end{aligned}$$

Combining these equations yields the first (upper) bound. The second (lower) bound is deduced from the centered case (i.e. when the distribution is centered on the optimum) as in the previous section. □

Figure I.2 gives an illustration of the bounds. Until $\mu \simeq (1-\epsilon)^d\lambda$, the centered and non centered case coincide when $\lambda \to \infty$: in this case, we can have a more precise asymptotic result for the choice of μ.

Theorem 32. *Consider $d > 0$, $r > 0$ and $y \in \mathbb{R}^d$. Let $\epsilon = \frac{\|y\|}{r} \in [0,1)$ and $f(x) = \|x-y\|^2$. When using $\mu = \lfloor c\lambda \rfloor$ with $0 < c < (1-\epsilon)^d$, we get as $\lambda \to \infty$, for a fixed d,*

$$\mathbb{E}_{X_1\ldots X_\lambda\sim\mathcal{B}(0,r)}\left[f(\bar{X}_{(\mu)})\right] = \frac{dr^2c^{2/d-1}}{(d+2)\lambda} + o\left(\frac{1}{\lambda}\right).$$

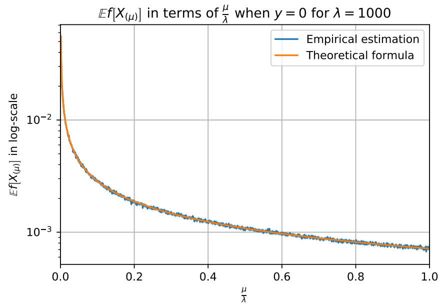


Figure I.1: Centered case: validation of the theoretical formula for $\mathbb{E}_{X_1 \ldots X_\lambda \sim \mathcal{B}(0,r)}\left[f(\bar{X}_{(\mu)})\right]$ when $y = 0$ from Theorem 30 for $d = 5$, $\lambda = 1000$ and $R = 1$. 1000 samples have been drawn to estimate the expectation. The two curves overlap, showing agreement between theory and practice.

Proof. Let $\mu_\lambda = \lfloor c\lambda \rfloor$ with $0 < c < (1-\epsilon)^d$. We immediately have from Hoeffding's concentration inequality:

$$\mathbb{P}_{U \sim B\left(\lambda,(1-\epsilon)^d\right)}(U \leq \mu_\lambda) \in o(\frac{1}{\lambda})$$

when $\lambda \to \infty$. From Corollary 5, we also get:

$$\frac{r^2 d \times \Gamma(\lambda+1)\Gamma(\mu_\lambda + 1 + 2/d)}{\mu_\lambda (d+2)\Gamma(\mu_\lambda+1)\Gamma(\lambda+1+2/d)} \sim \frac{d\, r^2 c^{2/d-1}}{(d+2)\lambda}.$$

Using the inequalities of Theorem 31, we obtain the desired result. □

The result of Theorem 32 shows that a convergence rate $O(\lambda^{-1})$ can be attained for the μ-best approach with $\mu > 1$. The rate for $\mu = 1$ is $\Theta(\lambda^{-2/d})$, proving that the μ-best approach leads asymptotically to a better estimation of the optimum. If we consider the problem $\min_\mu \max_{y:\|y\| \leq \epsilon r} \mathbb{E}\left[f_y(\bar{X}_{(\mu)})\right]$ with f_y the objective function $x \mapsto \|x-y\|^2$, then $\mu = \lfloor c\lambda \rfloor$ with $0 < c < (1-\epsilon)^d$ achieves the $O\left(\lambda^{-1}\right)$ progress rate.

All the results we proved in this section are easily extendable to strongly convex quadratic functions. For larger class of functions, it is less immediate, and left as future work.

I.2.5 Using quasi-convexity

The method above was designed for the sphere function, yet its adaptation to other quadratic convex functions is straightforward. On the other hand, our reasoning might break down when applied to multimodal functions. We thus consider an adaptive strategy to define μ. A desirable property to a μ-best approach is that the level-sets of the functions are convex. A simple workaround is to choose μ maximal such that there is a quasi-convex function which is identical to f on $\{X_{(1)}, \ldots, X_{(\mu)}\}$. If the objective function is quasi-convex on the convex hull of

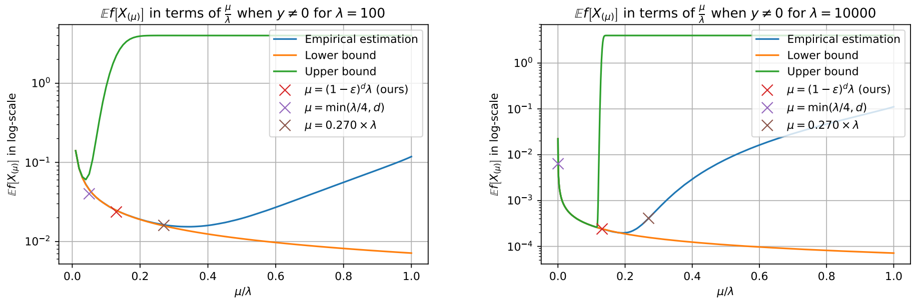


Figure I.2: Non centered case: validation of the theoretical bounds for $\mathbb{E}_{X_1 \ldots X_\lambda \sim \mathcal{B}(0,r)}\left[f(\bar{X}_{(\mu)})\right]$ when $\|y\| = \frac{R}{3}$ (i.e. $\epsilon = \frac{1}{3}$) from Theorem 31 for $d = 5$ and $R = 1$. We implemented $\lambda = 100$ and $\lambda = 10000$. 10000 samples have been drawn to estimate the expectation. We see that such a value for μ is a good approximation of the minimum of the empirical values: we can thus recommend $\mu = \lfloor \lambda(1-\epsilon)^d \rfloor$ when $\lambda \to \infty$. We also added some classical choices of values for μ from literature: when $\lambda \to \infty$, our method performs the best.

$\{X_{(1)}, \ldots, X_{(\tilde{\mu})}\}$ with $\tilde{\mu} \leq \lambda$, then: for any $i \leq \tilde{\mu}$, $X_{(i)}$ is on the frontier (denoted ∂) of the convex hull of $\{X_{(1)}, \ldots, X_{(i)}\}$ and the value

$$h = \max\left\{i \in [1, \lambda], \forall j \leq i, X_{(j)} \in \partial\left[\texttt{ConvexHull}(X_{(1)}, \ldots, X_{(j)})\right]\right\}$$

verifies $h \geq \tilde{\mu}$ so that $\mu = \min(h, \tilde{\mu})$ is actually equal to $\tilde{\mu}$. As a result:

- in the case of the sphere function, or any quasi-convex function, if we set $\tilde{\mu} = \lfloor \lambda(1-\epsilon)^d \rfloor$, using $\mu = \min(h, \tilde{\mu})$ leads to the same value of $\mu = \tilde{\mu} = \lfloor \lambda(1-\epsilon)^d \rfloor$. In particular, we preserve the theoretical guarantees of the previous sections for the sphere function $x \mapsto \|x - y\|^2$.
- if the objective function is not quasi-convex, we can still compute the quantity h defined above, but we might get a μ smaller than $\tilde{\mu}$. However, this strategy remains meaningful at it prevents from keeping too many points when the function is "highly" non-quasi-convex.

I.3 Experiments

To validate our theoretical findings, we first compare the formulas obtained in Theorems 30 and 31 with their empirical estimates. We then perform larger scale experiments in a one-shot optimization setting.

I.3.1 Experimental validation of theoretical formulas

Figure I.1 compares the theoretical formula from Theorem 30 and its empirical estimation: we note that the results coincide and validate our formula. Moreover, the plot confirms that taking the μ-best points leads to a lower regret than the 1-best approach.

We also compare in Figure I.2 the theoretical bounds from Theorem 31 with their empirical estimates. We remark that for $\mu \leq (1-\epsilon)^d\lambda$ the convergence of the two bounds to $\mathbb{E}(f(\bar{X}_{(\mu)}))$

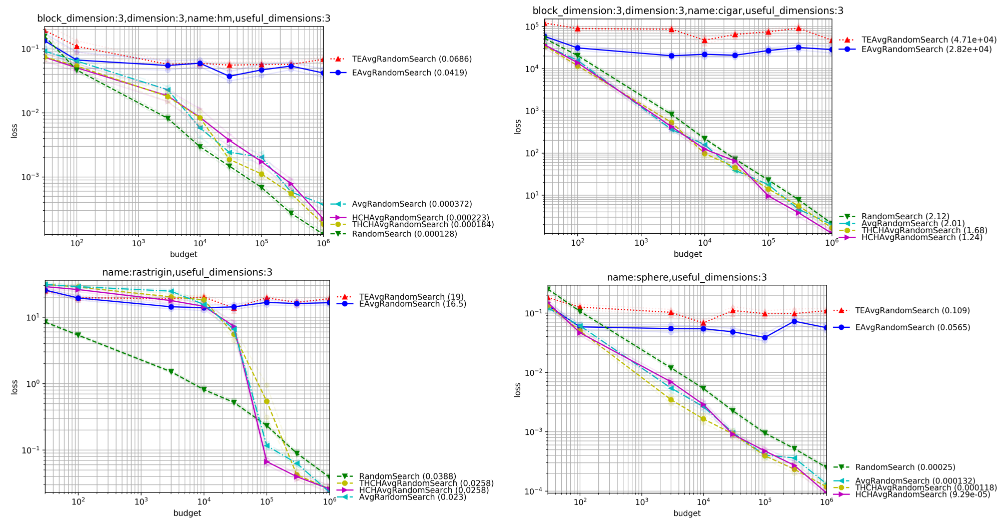


Figure I.3: Experimental curves comparing various methods for choosing μ as a function of λ in dimension 3. Standard deviations are shown by lighter lines (close to the average lines). Each x-axis value is computed independently. Our proposed formulas `HCHAvg` and `THCHAvg` perform well overall. See Fig. I.4 for results in dimension 25.

is fast. There exists a transition phase around $\mu \simeq (1-\epsilon)^d \lambda$ on which the regret is reaching a minimum: thus, one needs to choose μ both small enough to reduce bias and large enough to reduce variance. We compared to other empirically estimated values for μ from Beyer and Schwefel [2002a], Hansen and Ostermeier [2003], Beyer and Sendhoff [2008]. It turns out that if the population is large, our formula for μ leads to a smaller regret. Note that our strategy assumes that ϵ is known, which is not the case in practice. It is interesting to note that if the center of the distribution and the optimum are close (i.e. ϵ is small), one can choose a larger μ to get a lower variance on the estimator of the optimum.

I.3.2 One-shot optimization in Nevergrad

In this section we test different formulas and variants for the choice of μ for a larger scale of experiments in the one-shot setting. Equations I.1-I.6 present the different formulas for μ used in our comparison.

$$\mu = 1 \qquad \text{No prefix} \tag{I.1}$$

$$\mu = \texttt{clip}\left(1, d, \frac{\lambda}{4}\right) \qquad \text{Prefix: } \texttt{Avg} \text{ (averaging)} \tag{I.2}$$

$$\mu = \texttt{clip}\left(1, \infty, \frac{\lambda}{1.1^d}\right) \qquad \text{Prefix: } \texttt{EAvg} \text{ (Exp. Averaging)} \tag{I.3}$$

$$\mu = \texttt{clip}\left(1, \min\left(h, \frac{\lambda}{4}\right), d + \frac{\lambda}{1.1^d}\right) \qquad \text{Prefix: } \texttt{HCHAvg} \text{ (h from Convex Hull)} \tag{I.4}$$

$$\mu = \texttt{clip}\left(1, \infty, \frac{\lambda}{1.01^d}\right) \qquad \text{Prefix: } \texttt{TEAvg} \text{ (Tuned Exp. Avg)} \tag{I.5}$$

$$\mu = \texttt{clip}\left(1, \min\left(h, \frac{\lambda}{4}\right), d + \frac{\lambda}{1.01^d}\right) \qquad \text{Prefix: } \texttt{THCHAvg} \text{ (Tuned HCH Avg)} \tag{I.6}$$

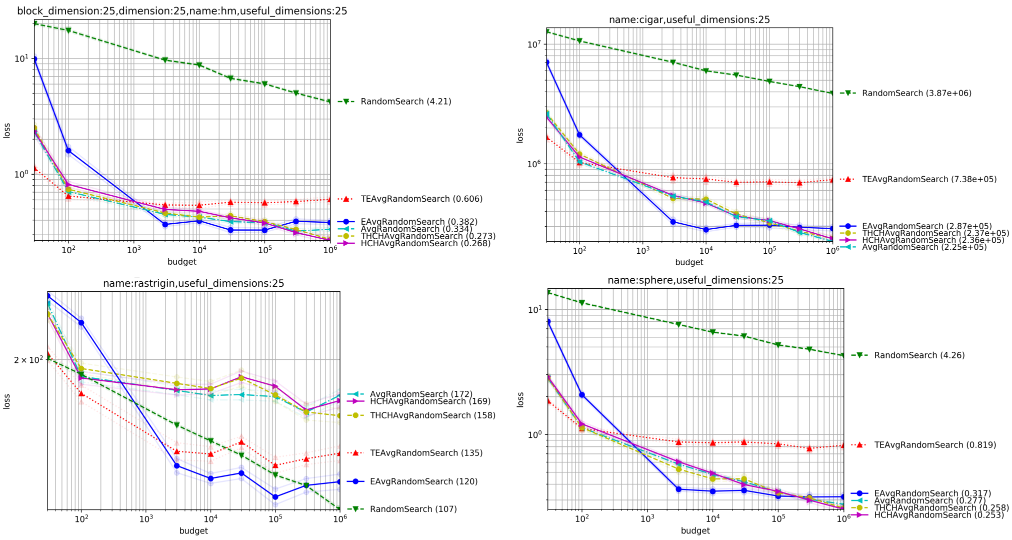


Figure I.4: Experimental curves comparing various methods for choosing μ as a function of λ in dimension 25 (Fig. I.3, continued for dimension 25; see Fig. I.5 for dimension 200). Our proposals lead to good results but we notice that they are outperformed by `TEAvg` and `EAvg` for Rastrigin: it is better to not take into account non-quasi-convexity because the overall shape is more meaningful that local ruggedness. This phenomenon does not happen for the more rugged HM (Highly Multimodal) function. It also does not happen in dimension 3 or dimension 200 (previous and next figures): in those cases, THCH performed best. Confidence intervals shown in lighter color (they are quite small, and therefore they are difficult to notice).

where $\texttt{clip}(a, b, c) = \max(a, \min(b, c))$ is the projection of c in $[a, b]$ and h is the maximum i such that, for all $j \leq i$, $X_{(j)}$ is on the frontier of the convex hull of $\{X_{(1)}, \ldots, X_{(j)}\}$ (Sect. I.2.5). Equation I.1 is the naive recommendation “pick up the best so far”. Equation I.2 existed before the present work: it was, until now, the best rule Teytaud [2010] , overall, in the Nevergrad platform. Equations I.3 and I.5 are the proposals we deduced from Theorem 32: asymptotically on the sphere, they should have a better rate than Equation I.1. Equations I.4 and I.6 are counterparts of Equations I.3 and I.5 that combine the latter formulas with ideas from Teytaud [2010]. Theorem 32 remains true if we add to μ some constant depending on d so we fine tune our theoretical equation (Eq. I.3) with the one provided by Teytaud [2010], so that μ is close to the value in Eq. I.2 for moderate values of λ. We perform experiments in the open source platform Nevergrad Rapin and Teytaud [2018].

While previous experiments (Figures I.1 and I.2) were performed in a controlled ad hoc environment, we work here with more realistic conditions: the sampling is Gaussian (i.e. not uniform in a ball), the objective functions are not all sphere-like, and budgets vary but are not asymptotic. Figures I.3, I.4, I.5 present our results in dimension 3, 25 and 200 respectively. The objective functions are randomly translated using $\mathcal{N}(0, 0.2I_d)$. The objective functions are defined as $f_{Sphere}(x) = \|x\|^2$, $f_{Cigar}(x) = 10^6 \sum_{i=2}^{d} x_i^2 + x_1^2$, $f_{HM}(x) = \sum_{i=1}^{d} x_i^2 \times (1.1 + \cos(1/x_i))$, $f_{Rastrigin}(x) = 10d + f_{sphere}(x) - 10 \sum_i \cos(2\pi x_i)$. Our proposed equations `TEAvg` and `EAvg` are unstable: they sometimes perform excellently (e.g. everything in dimension

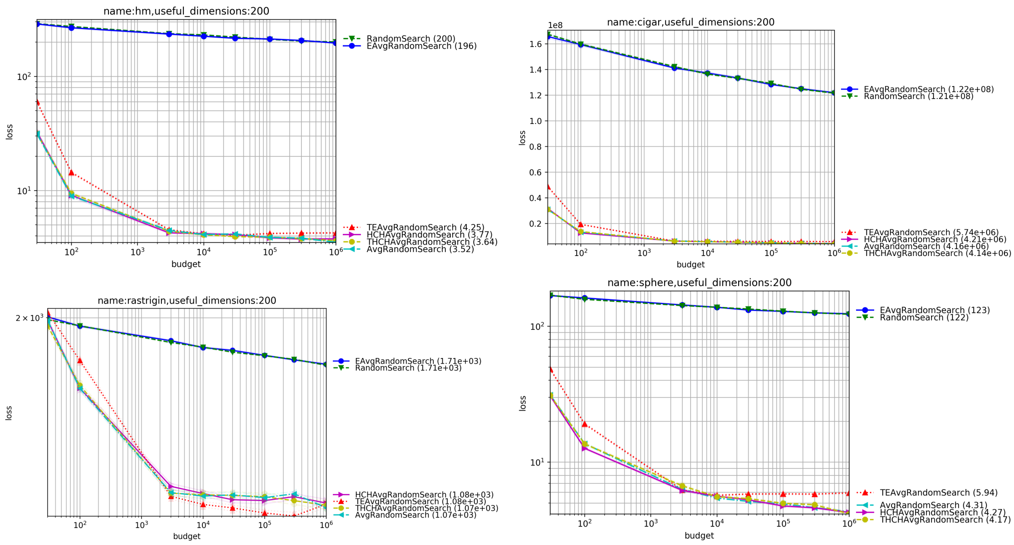


Figure I.5: Experimental curves comparing various methods for choosing μ as a function of λ in dimension 200 (Figures I.3 and I.4, continued for dimension 200). Confidence intervals shown in lighter color (they are quite small, and therefore they are difficult to notice). Our proposed methods `THCHAvg` and `HCHAvg` perform well overall.

25, Figure I.4), but they can also fail dramatically (e.g. dimension 3, Figure I.3). Our combinations `THCHAvg` and `HCHAvg` perform well: in most settings, `THCHAvg` performs best. But the gap with the previously proposed `Avg` is not that big. The use of quasi-convexity as described in Section I.2.5 was usually beneficial: however, in dimension 25 for the Rastrigin function, it prevented the averaging from benefiting from the overall "approximate" convexity of Rastrigin. This phenomenon did not happen for the "more" multimodal function HM, or in other dimensions for the Rastrigin function.

I.4 Conclusion

We have proved formally that the average of the μ best is better than the single best in the case of the sphere function (simple regret $O(1/\lambda)$ instead of $O(1/\lambda^{2/d})$) with uniform sampling. We suggested a value $\mu = \lfloor c\lambda \rfloor$ with $0 < c < (1-\epsilon)^d$. Even better results can be obtained in practice using quasi-convexity, without losing the theoretical guarantees of the convex case on the sphere function. Our results have been successfully implemented in Rapin and Teytaud [2018]. The improvement compared to the state of the art, albeit moderate, is obtained without any computational overhead in our method, and supported by a theoretical result.

J Asymptotic convergence rates for averaging strategies

J.1 Introduction

Finding the minimum of a function from a set of λ points $(x_i)_{i\leq\lambda}$ and their images $(f(x_i))_{i\leq\lambda}$ is a standard task used for instance in hyper-parameter tuning **?**, or control problems. While random search estimate of the optimum consists in returning $\arg\min f(x_i)_{i\leq\lambda}$, in this paper we focus on the similar strategy that consists in averaging the μ best samples, i.e. returning $\frac{1}{\mu}\sum_{i=1}^{\mu} x_{(i)}$ where $f(x_{(1)}) \leq \ldots \leq f(x_{(\lambda)})$.

These kinds of strategies are used in many evolutionary algorithms such as CMA-ES. Although experiments show that these methods perform well, it is not still understood why taking the average of best points actually leads to a lower regret. In Meunier et al. [2020a], it is proved in the case of quadratic functions that the regret is indeed lower for the averaging strategy than for pure random search. In this paper, we extend the result of this paper by proving convergence rates for a wide class of functions including three times continuously differentiable functions with unique optima.

J.1.1 Related Work

Better than picking up the best

Given a finite number of samples λ equipped with their fitness values, we can simply pick up the best, or average the "best ones" Beyer [1995], Meunier et al. [2020a], or apply a surrogate model Gupta et al. [2021], Sudret [2012], Dushatskiy et al. [2021], Auger et al. [2005a], Rudi et al. [2020]. Overall, the best is quite robust, but the surrogate or the averaging usually provides better convergence rates. Using surrogate modeling is fast when the dimension is moderate and the objective function is smooth (simple regret in $O(\lambda^{-m/d})$ for λ points in dimension d with m times differentiability, leading to superlinear rates in evolutionary computation Auger et al. [2005a]). In this paper, we are interested in the rates obtained by averaging the best samples for a wide class of functions. We extend the results of Meunier et al. [2020a] which only hold for the sphere function.

Weighted averaging

Among the various forms of averaging, it has been proposed to take into account the fact that the sampling is not uniform (evolutionary algorithms in continuous domains typically use Gaussian sampling) in Teytaud and Teytaud [2009]: we here simplify the analysis by considering a uniform sampling in a ball, though we acknowledge that this introduces the constraint that the optimum is indeed in the ball. Arnold et al. [2009], Auger et al. [2011] have proposed weights depending on the fitness value, though they acknowledge a moderate impact: we here consider equal weights for the μ best.

Choosing the selection rate

The choice of the selection rate μ/λ is quite debated in evolutionary computation: one can find $\mu = \lambda/7$ Escalante and Reyes [2013], $\mu = \lambda/2$ Beyer and Sendhoff [2008], $\mu = 0.27\lambda$ Beyer and Schwefel [2002a], $\mu = \lambda/4$ Hansen and Ostermeier [2003], $\mu = \min(d, \lambda/4)$ Teytaud [2007], Fournier and Teytaud [2010] and still others in Beyer [1995], Jebalia and Auger [2010]. In this paper, we focus on the selection rate when the number of samples λ is very large in the case of parallel optimization. In this case, the selection ratio would tend to 0. We carefully analyze this ratio and derive convergence rates using this selection ratio.

Taking into account many basins

While averaging the best samples, the non-uniqueness of an optimum might lead to averaging points coming from different basins. Thus we consider at first the case of a unique optimum and hence a unique basin. Then we aim to tackle the case where there are possibly different basins. Island models Skolicki [2007] have also been proposed for taking into account different basins. Meunier et al. [2020a] has proposed a tool for adapting μ depending on the (non) quasi-convexity. In the present work, we extend the methodology proposed in Meunier et al. [2020a].

J.1.2 Outline

In the present paper, we first introduce, in Section J.2, the large class of functions we will study, and study some useful properties of these functions in Section J.3. Then, in Section J.4, we prove upper and lower convergence rates for random search for these functions. In Section J.5, we extend Meunier et al. [2020a] by showing that asymptotically in the number of samples λ, the handled functions satisfy a better convergence rate than random search. We then extend our results on wider classes of functions in Section J.6. Finally we validate experimentally our theoretical findings and compare with other parallel optimization methods.

J.2 Beyond quadratic functions

In the present section, we present the assumptions to extend the results from Meunier et al. [2020a] to the non-quadratic case. We will denote $B(0, r)$ the closed ball centered at 0 of radius r in $\mathbb{R}^d$ endowed with its canonical Euclidean norm denoted by $\|\cdot\|$. We will also denote by $\overset{\circ}{B}(0, r)$ the corresponding *open* ball. All other balls intervening in what follows will also follow that notation. For any subset $S \subset B(0, r)$, we will denote $U(S)$ the uniform law on S.

Let $f : B(0, r) \to \mathbb{R}$ be a continuous function for which we would like to find an optimum point x^*. The existence of such an optimum point is guaranteed by continuity on a compact set. For the sake of simplicity, we assume that $f(x^\star) = 0$. We define the h-level sets of f as follows.

Definition 33. *Let $f : B(0,r) \to \mathbb{R}$ be a continuous function. The closed sublevel set of f of level h is defined as:*

$$S_h := \{x \in B(0,r) \mid f(x) \leq h\}.$$

We now describe the assumptions we will make on the function f that we optimize.

Assumption 7. *$f : B(0,r) \to \mathbb{R}$ is a continuous function and admits a unique optimum point $x^\star$ such that $\|x^\star\| < r$. Moreover we assume that f can be written:*

$$f(x) = (x - x^\star)^T \mathbf{H}(x - x^\star) + \Big((x - x^\star)^T \mathbf{H}(x - x^\star)\Big)^{\alpha/2} \varepsilon(x - x^\star)$$

for some bounded function ε (there exists $M > 0$ such that for all x, $|\varepsilon(x)| \leq M$), $\mathbf{H}$ a symmetric positive definite matrix and $\alpha > 2$ a real number.

Note that H is uniquely defined by the previous relation. In the following we will denote by $e_1(\mathbf{H})$ and $e_d(\mathbf{H})$ respectively the smallest and the largest eigenvalue of $\mathbf{H}$. As $\mathbf{H}$ is positive definite, we have $0 < e_1(\mathbf{H}) \leq e_d(\mathbf{H})$. We will also set $\|x\|_{\mathbf{H}} = \sqrt{x^T \mathbf{H} x}$, which is a norm (*the* $\mathbf{H}$*-norm*) on $\mathbb{R}^d$ as $\mathbf{H}$ is symmetric positive definite. We then have $f(x) = \|x - x^\star\|_{\mathbf{H}}^2 + \|x - x^\star\|_{\mathbf{H}}^{\alpha} \varepsilon(x - x^\star)$

Remark 17 (Why a unique optimum ?). *The uniqueness of the optimum is an hypothesis required to avoid that chosen samples come from two or more wells for f. In this case the averaging strategy would lead to a mistaken point because points from the different wells would be averaged. Nonetheless, multimodal functions can be tackled using our non-quasiconvexity trick (Section J.6.2).*

Remark 18 (Which functions f satisfy Assumption 7?). *One may wonder if Assumption 7 is restrictive or not. We can remark that three times continuously differentiable functions satisfy the assumption with $\alpha = 3$, as long as the unique optimum satisfies a strict second order stationary condition. Also, we will see in Section J.6.1 that results are immediately valid for strictly increasing transformations of any f for which Assumption 7 holds, so that we indirectly include all piecewise linear functions as well as long as they have a unique optimum. So the class of functions is very large, and in particular allows non symmetric functions to be treated, which might seem counter intuitive at first.*

The aim of this paper is to study a parallel optimization problem as follows. We sample $X_1, \cdots, X_\lambda$ from the uniform distribution on $B(0,r)$. Let $X_{(1)}, \cdots, X_{(\lambda)}$ denote the ordered random variables, where the order is given by the objective function

$$f(X_{(1)}) \leq \cdots \leq f(X_{(\lambda)}).$$

We then introduce the μ-best average

$$\overline{X}_{(\mu)} = \frac{1}{\mu} \sum_{i=1}^{\mu} X_{(i)}$$

In the following of the paper, we will compare the standard random search algorithm (i.e. $\mu = 1$) with the algorithm that consists in returning the average of the μ best points. To this end, we will study the expected simple regret for functions satisfying the assumption:

$$\mathbb{E}\big[f(\overline{X}_{(\mu)})\big]$$

J.3 Technical lemmas

In this section, we prove two technical lemmas on f that will be useful to study the convergence of the algorithm. The first one shows that f can be upper bounded and lower bounded by two spherical functions.

Lemma 19. *Under Assumption 7, there exist two real numbers* $0 < l \leq L$, *such that, for all* $x \in B(0, r)$*:*

$$l\|x - x^\star\|^2 \leq f(x) \leq L\|x - x^\star\|^2. \tag{J.1}$$

Moreover such l *and* L *must satisfy* $0 < l \leq e_1(\mathbf{H}) \leq e_d(\mathbf{H}) \leq L$.

Proof. As $\mathbf{H}$ is symmetric positive definite, we have the following classical inequality for the $\mathbf{H}$-norm

$$e_1(\mathbf{H})\|x - x^\star\|^2 \leq \|x - x^\star\|^2_{\mathbf{H}} \leq e_d(\mathbf{H})\|x - x^\star\|^2 \tag{J.2}$$

Now set for $x \in B(0, r) \setminus \{x^\star\}$

$$\phi(x) := \frac{f(x) - f(x^\star)}{\|x - x^\star\|^2} = \frac{\|x - x^\star\|^2_{\mathbf{H}}}{\|x - x^\star\|^2}(1 + \|x - x^\star\|^{\alpha-2}_{\mathbf{H}}\varepsilon(x - x^\star)).$$

By the above inequalities, we have

$$e_1(\mathbf{H})^{(\alpha-2)/2}\|x - x^\star\|^{\alpha-2} \leq \|x - x^\star\|^{\alpha-2}_{\mathbf{H}} \leq e_d(\mathbf{H})^{(\alpha-2)/2}\|x - x^\star\|^{\alpha-2}.$$

Thus, as $\alpha > 2$, we obtain $\|x - x^\star\|^{\alpha-2}_{\mathbf{H}} \to_{x \to x^\star} 0$. By assumption, the function ε is also bounded as $x \to x^\star$.
We thus conclude that there exists $\delta > 0$ such that, for all $x \in \overset{\circ}{B}(x^\star, \delta)$

$$\frac{1}{2}e_1(\mathbf{H}) \leq \phi(x) \leq 2e_d(\mathbf{H}).$$

Now notice that $B(0, r) \setminus \overset{\circ}{B}(x^\star, \delta)$ is a closed subset of the compact set $B(0, r)$ hence it is also compact. Moreover, by assumption f is continuous on $B(0, r)$ and $f(x) > 0 = f(x^\star)$ for all $x \neq x^\star$. Hence ϕ is continuous and positive on this compact set. Thus it attains its minimum and maximum on this set and its minimum is positive. In particular, we can write, on this set, for some $l_0, L_0 > 0$

$$l_0 \leq \phi(x) \leq L_0.$$

We now set $l = \min\{l_0, \frac{1}{2}e_1(\mathbf{H})\}$. Note that $l > 0$ because $l_0 > 0$ and $e_1(\mathbf{H}) > 0$ (as $\mathbf{H}$ is positive definite). We also set $L = \max\{L_0, 2e_1(\mathbf{H})\}$ which is also positive. These are global bounds for ϕ which gives the first part of the result.
For the second part, let $\mathbf{u}_1$ be a normalized eigenvector respectively associated to $e_1(\mathbf{H})$. Then

$$\frac{f(x^\star + \epsilon \mathbf{u}_1)}{\|\epsilon \mathbf{u}_1\|^2} = e_1(\mathbf{H}) + \epsilon^{\alpha-2}\varepsilon(\epsilon \mathbf{u}_1)$$

Taking the limit as $\epsilon \to 0$. we get that, if l satisfies (J.1), then $l \leq e_1(\mathbf{H})$. Similarly, we can prove that $L \geq e_d(\mathbf{H})$. □

Secondly, we frame S_h into two ellipsoids as $h \to 0$. This lemma is a consequence of the assumptions we make on f.

Lemma 20. *Under Assumption 7, there exists $h_0 \geq 0$ such that for $h \leq h_0$, we have $A_h \subset S_h \subset B_h$ where:*

$$A_h := \{x \mid \|x - x^\star\|_{\mathbf{H}} \leq \phi_-(h)\}$$
$$B_h := \{x \mid \|x - x^\star\|_{\mathbf{H}} \leq \phi_+(h)\}$$

with $\phi_-(h)$ and $\phi_+(h)$ two functions satisfying

$$\begin{aligned} \phi_-(h) &= \sqrt{h} - \frac{M}{2}h^{(\alpha-1)/2} + o(h^{(\alpha-1)/2}) \\ \textit{and } \phi_+(h) &= \sqrt{h} + \frac{m}{2}h^{(\alpha-1)/2} + o(h^{(\alpha-1)/2}) \end{aligned}$$

when $h \to 0$ for some constants $m > 0$ and $M > 0$ which are respectively a (specific) lower and upper bound for ε.

Proof. By assumption $|\varepsilon| \leq M$, hence we have:

$$\{x \in B(0,r) \mid \|x - x^\star\|^2_{\mathbf{H}} + M\|x - x^\star\|^\alpha_{\mathbf{H}} \leq h\} \subset S_h$$

Let $g\colon u \mapsto u^2 + Mu^\alpha$. This is a continuous, strictly increasing function on $[0, +\infty)$. By a classical consequence of the intermediate value theorem, this implies that g admits a continuous, strictly increasing inverse function. Note that $g(0) = 0$ hence $g^{-1}(0) = 0$.

Thus we can write $\{u \geq 0 | u^2 + Mu^\alpha \leq h\} = [0, g^{-1}(h)]$. We now denote g^{-1} by ϕ_-. As ϕ_- is non-decreasing, we get

$$\{x \in B(0,r) \mid \|x - x^\star\|_{\mathbf{H}}^2 + M\|x - x^\star\|_{\mathbf{H}}^\alpha \leq h\} = A_h \cap B(0,r)$$

Now observe that for h sufficiently small

$$\{x \in B(0,r) \mid \|x - x^\star\|_{\mathbf{H}}^2 + M\|x - x^\star\|_{\mathbf{H}}^\alpha \leq h\} = A_h.$$

Indeed, if $x \in A_h$, we have by the triangle inequality and (J.2)

$$\begin{aligned}
\|x\| &\leq \|x^\star\| + \|x - x^\star\| \\
&\leq \|x^\star\| + e_1(\mathbf{H})^{-1/2}\|x - x^\star\|_{\mathbf{H}} \\
&\leq \|x^\star\| + e_1(\mathbf{H})^{-1/2}\phi_-(h)
\end{aligned}$$

Recall that by assumption $\|x^\star\| < r$ and let $\delta = r - \|x^\star\|$. As $\phi_-(h) \to_{h\to 0} 0$, for h sufficiently small, we have $e_1(\mathbf{H})^{-1/2}\phi_-(h) \leq \delta$ hence $\|x\| \leq r$ for h sufficiently small, which gives the inclusion $A_h \subset S_h$.
For the asymptotics of ϕ_-, as we have by definition $\phi_-(h)^2(1 + M\phi_-(h)^{\alpha-2}) = h$, and as $\phi_-(h) \to_{h\to 0} 0$ we deduce that $\phi_-(h) \sim_0 \sqrt{h}$. Let us define $u(h) = \phi_-(h) - \sqrt{h}$. We have $u(h) \in o(\sqrt{h})$. We then compute:

$$(\sqrt{h} + u(h))^2 + M(\sqrt{h} + u(h))^\alpha = h$$

This gives

$$\begin{aligned}
u(h)(u(h) + 2\sqrt{h}) &= -Mh^{\alpha/2}(1 + \frac{u(h)}{\sqrt{h}})^\alpha \\
u(h)(\frac{u(h)}{2\sqrt{h}} + 1) &= -\frac{M}{2}h^{(\alpha-1)/2}(1 + \frac{u(h)}{\sqrt{h}})^\alpha
\end{aligned}$$

As $u(h) \in o(\sqrt{h})$ for $h \to 0$, we obtain

$$u(h) \sim -\frac{M}{2}h^{(\alpha-1)/2}.$$

which concludes for ϕ_-.

On the other side, we recall that $f(x) > 0$ for all $x \neq x^\star$ as $x^\star$ is the unique minimum of f on $B(0,r)$. Write

$$0 < \|x - x^\star\|_{\mathbf{H}}^2(1 + \|x - x^\star\|_{\mathbf{H}}^{\alpha-2}\varepsilon(x - x^\star)).$$

Now observe that, as $\|x^\star\| < r$, we have for $x \in B(0,r)$, by the triangle inequality, $\|x - x^\star\| < 2r$. Hence, by the classical inequality for the $\mathbf{H}$-norm (J.2), we get

$$\varepsilon(x - x^\star) > -\frac{1}{\|x - x^\star\|_{\mathbf{H}}^{\alpha-2}} \geq -\left(\sqrt{e_d(\mathbf{H})}2r\right)^{-(\alpha-2)} =: -m$$

So we have:

$$S_h \subset \{x \in B(0,r) \mid \|x - x^\star\|_{\mathbf{H}}^2 - m\|x - x^\star\|_{\mathbf{H}}^\alpha \leq h\}$$

The function $g\colon u \mapsto u^2 - mu^\alpha$ is differentiable. A study of the derivative shows that g is continuous, strictly increasing on $[0, r_0]$ and continuous, strictly decreasing on $[r_0, +\infty[$ where $r_0 = (\frac{2}{\alpha m})^{1/(\alpha-2)}$. Hence $g_{|[0,r_0]}$ admits a continuous strictly increasing inverse ϕ_+ and $g_{|[r_0,+\infty[}$ a continuous strictly decreasing inverse $\tilde{\phi}$. We thus write

$$\{u \geq 0 | u^2 - mu^\alpha \leq h\} = [0, \phi_+(h)] \cup [\tilde{\phi}(h), +\infty).$$

Hence

$$\begin{aligned}\{x \in B(0,r) \mid& \|x - x^\star\|_{\mathbf{H}}^2 - m\|x - x^\star\|_{\mathbf{H}}^\alpha \leq h\} \\ &= \big(B_h \cap B(0,r)\big) \cup \big(B(0,r) \cap V_h\big)\end{aligned}$$

with $V_h = \{x \in \mathbb{R}^d | \; \|x - x^\star\|_{\mathbf{H}} > \tilde{\phi}(h)\}$. We now show that for h sufficiently small

$$\{x \in B(0,r) \mid \|x - x^\star\|_{\mathbf{H}}^2 - m\|x - x^\star\|_{\mathbf{H}}^\alpha \leq h\} = B_h.$$

Indeed, note first that if $x \in B(0,r)$, we obtain by (J.2)

$$\|x - x^\star\|_{\mathbf{H}}^2 \leq e_d(\mathbf{H})\|x - x^\star\|^2 < 4e_d(\mathbf{H})r^2.$$

where we have used that, as $\|x\| < r$, the triangle inequality gives $\|x - x^\star\| < 2r$. Hence $B(0,r) \subset \{x \in \mathbb{R}^d | \; \|x - x^\star\|_{\mathbf{H}}^2 < 4e_d(\mathbf{H})r^2\}$. We now show that $B(0,r) \subset \{x \in \mathbb{R}^d | \; \|x - x^\star\|_{\mathbf{H}} \leq \tilde{\phi}(h)\}$. Indeed, at $h = 0, 0 = \phi_+(0) < \tilde{\phi}(0)$ are by definition, the two roots of

$$u^2 - mu^\alpha = 0.$$

Hence $\tilde{\phi}(0) = \sqrt{e_d(\mathbf{H})2r}$. By continuity of $\tilde{\phi}(h)$ at $h = 0$, we obtain that $B(0,r) \subset \{x \in \mathbb{R}^d | \; \|x - x^\star\|_{\mathbf{H}} \leq \tilde{\phi}(h)\}$ for h sufficiently small. As $\phi_+(h) \leq \tilde{\phi}(h)$, we thus obtain that, for h sufficiently small, $V_h \cap B(0,r) = \emptyset$. Next, the same line of reasoning as the one for ϕ_-, using that $\phi_+(h) \to_{h\to 0} 0$ and $\|x^\star\| < r$, shows that $B_h \cap B(0,r) = B_h$ for h sufficiently small.

Hence, for h small enough we have

$$\{x \in B(0,r) \mid \|x - x^\star\|_{\mathbf{H}}^2 - m\|x - x^\star\|_{\mathbf{H}}^\alpha \leq h\} = B_h.$$

This gives $S_h \subset B_h$.
Finally, similarly to ϕ_-, we can show that $\phi_+(h) = \sqrt{h} + \frac{m}{2} h^{(\alpha-1)/2} + o(h^{(\alpha-1)/2})$, which concludes the proof of this lemma. □

J.4 Bounds for random search

In this section we provide upper bounds and lower bounds for the random search algorithm for functions satisfying Assumption 7. These bounds will also be useful for analyzing the convergence of the μ-best approach.

J.4.1 Upper Bound

First, we prove an upper bound for functions satisfying Assumption 7.

Lemma 21 (Upper bound for random search algorithm)**.** *Let f be a function satisfying Assumption 7. There exists a constant $C_0 > 0$ and an integer $\lambda_0 \in \mathbb{N}$ such that for all integers $\lambda \geq \lambda_0$:*

$$\mathbb{E}_{X_1,\cdots,X_\lambda \sim U(B(0,r))}\left[f\left(X_{(1)}\right)\right] \leq C_0 \lambda^{-\frac{2}{d}} \quad .$$

Proof. Let us first recall the following classical property about the expectation of a positive valued random variable:

$$\mathbb{E}_{X_1,\ldots,X_\lambda \sim U(B(0,r))}\left[f\left(X_{(1)}\right)\right] = \int_0^\infty \mathbb{P}\left[f\left(X_{(1)}\right) \geq t\right] dt$$

By independence of the samples we have:

$$\int_0^\infty \mathbb{P}\left[f\left(X_{(1)}\right) \geq t\right] dt = \int_0^\infty \mathbb{P}_{X \sim U(B(0,r))}[f(X) \geq t]^\lambda dt$$

Then thanks to Lemma 19:

$$\begin{aligned}
\int_0^\infty & \mathbb{P}_{X \sim U(B(0,r))}[f(X) \geq t]^\lambda dt \\
& \leq \int_0^\infty \mathbb{P}_{X \sim U(B(0,r))}\left[L\|X - x^\star\|^2 \geq t\right]^\lambda dt \\
& = \int_0^{L(r+\|x^\star\|)^2} \mathbb{P}\left[\|X - x^\star\| \geq \sqrt{\frac{t}{L}}\right]^\lambda dt
\end{aligned}$$

where the second equality follows because $\|X - x^\star\| \le r$ almost surely. Then, by definition of the uniform law as well as the non-increasing character of $t \mapsto \mathbb{P}_{X\sim U(B(0,r))}\left[\|X - x^\star\| \ge \sqrt{\frac{t}{L}}\right]$, we obtain

$$\begin{aligned}
&\int_0^{L(r+\|x^\star\|)^2} \mathbb{P}_{X\sim U(B(0,r))}\left[\|X - x^\star\| \ge \sqrt{\frac{t}{L}}\right]^\lambda dt \\
&= \int_0^{L(r-\|x^\star\|)^2} \mathbb{P}_{X\sim U(B(0,r))}\left[\|X - x^\star\| \ge \sqrt{\frac{t}{L}}\right]^\lambda dt \\
&+ \int_{L(r-\|x^\star\|)^2}^{L(r+\|x^\star\|)^2} \mathbb{P}_{X\sim U(B(0,r))}\left[\|X - x^\star\| \ge \sqrt{\frac{t}{L}}\right]^\lambda dt \\
&\le \int_0^{L(r-\|x^\star\|)^2} \left[1 - \left(\sqrt{\frac{t}{Lr^2}}\right)^d\right]^\lambda dt \\
&+ L\Big((r+\|x^\star\|)^2 - (r-\|x^\star\|)^2\Big)\mathbb{P}[\|X - x^\star\| \ge r - \|x^\star\|]^\lambda \\
&\le \int_0^{Lr^2} \left[1 - \left(\frac{t}{Lr^2}\right)^{\frac{d}{2}}\right]^\lambda dt + 4Lr\|x^\star\|\mathbb{P}[\|X - x^\star\| \ge r - \|x^\star\|]^\lambda \\
&= Lr^2 \int_0^1 \left[1 - u^{\frac{d}{2}}\right]^\lambda du + 4Lr\|x^\star\|\mathbb{P}[\|X - x^\star\| \ge r - \|x^\star\|]^\lambda
\end{aligned}$$

Note that $\mathbb{P}[\|X - x^\star\| < r - \|x^\star\|] < 1$. Thus the second term in the last equality satisfies $\mathbb{P}[\|X - x^\star\| < r - \|x^\star\|]^\lambda \in o(\lambda^{-2/d})$. The first term has a closed form given in Meunier et al. [2020a]:

$$\int_0^1 \left[1 - u^{\frac{d}{2}}\right]^\lambda du = \frac{\Gamma(\frac{d+2}{d})\Gamma(\lambda+1)}{\Gamma(\lambda+1+2/d)}$$

Finally thanks to the Stirling approximation, we conclude:

$$\mathbb{E}_{X_1,\ldots,X_\lambda\sim U(B(0,r))}\left[f\left(X_{(1)}\right)\right] \le C_1\lambda^{-2/d} + o(\lambda^{-2/d})$$

where $C_1 > 0$ is a constant independent from λ. □

This lemma proves that the strategy consisting in returning the best sample (i.e. random search) has an upper rate of convergence of order $\lambda^{-2/d}$, which depends on dimension of the space. It also worth noting this result is common in the literature Rudi et al. [2020], Bergstra and Bengio [2012]

J.4.2 Lower Bound

We now give a lower bound for the convergence of the random search algorithm. We also prove a conditional expectation bound that will be useful for the analysis of the μ-best averaging approach.

Lemma 22 (Lower bound for random search algorithm)**.** *Let f be a function satisfying Assumption 7. There exist a constant $C_1 > 0$ and $\lambda_1 \in \mathbb{N}$ such that for all integers $\lambda \geq \lambda_1$, we have the following lower bound for random search:*

$$\mathbb{E}_{X_1,\ldots,X_\lambda \sim U(B(0,r))}\left[f\left(X_{(1)}\right)\right] \geq C_1 \lambda^{-2/d} \quad .$$

Moreover, let $(\mu_\lambda)_{\lambda \in \mathbb{N}}$ be a sequence of integers such that $\forall \lambda \geq 2, 1 \leq \mu_\lambda \leq \lambda - 1$ and $\mu_\lambda \to \infty$. Then, there exist a constant $C_2 > 0$ and $\lambda_2 \in \mathbb{N}$ such that for all $h \in [0, \max f]$ and $\lambda \geq \lambda_2$, we have the following lower bound when the sampling is conditioned:

$$\mathbb{E}_{X_1,\ldots,X_\lambda \sim U(B(0,r))}\left[f\left(X_{(1)}\right) \mid f(X_{(\mu_\lambda+1)}) = h\right] \geq C_2 h \mu_\lambda^{-2/d} \quad .$$

Proof. The proof is very similar to the previous one. Let us first show the unconditional inequality. We use the identity for the expectation of a positive random variable

$$\begin{aligned}
&\mathbb{E}_{X_1,\ldots,X_\lambda \sim U(B(0,r))}\left[f\left(X_{(1)}\right)\right] \\
&\quad = \int_0^\infty \mathbb{P}_{X_1,\ldots,X_\lambda \sim U(B(0,r))}\left[f\left(X_{(1)}\right) \geq t\right] dt
\end{aligned}$$

Since the samples are independent, we have

$$\begin{aligned}
&\int_0^\infty \mathbb{P}_{X_1,\ldots,X_\lambda \sim U(B(0,r))}\left[f\left(X_{(1)}\right) \geq t\right] dt \\
&\quad = \int_0^\infty \mathbb{P}_{X \sim U(B(0,r))}[f(X) \geq t]^\lambda dt
\end{aligned}$$

Using Lemma 19, we get:

$$\begin{aligned}
&\int_0^\infty \mathbb{P}_{X \sim U(B(0,r))}[f(X) \geq t]^\lambda dt \\
&\quad \geq \int_0^\infty \mathbb{P}_{X \sim U(B(0,r))}\left[l\|X - x^\star\|^2 \geq t\right]^\lambda dt \\
&\quad \geq \int_0^{l(r-\|x^\star\|)^2} \mathbb{P}_{X \sim U(B(0,r))}\left[l\|X - x^\star\|^2 \geq t\right]^\lambda dt \\
&\quad = \int_0^{l(r-\|x^\star\|)^2} \left[1 - \left(\sqrt{\frac{t}{lr^2}}\right)^d\right]^\lambda dt
\end{aligned}$$

We can decompose the integral to obtain:

$$\begin{aligned}
&\int_0^{l(r-\|x^\star\|)^2}\left[1-\left(\sqrt{\frac{t}{lr^2}}\right)^d\right]^\lambda dt\\
&=\int_0^{lr^2}\left[1-\left(\sqrt{\frac{t}{lr^2}}\right)^d\right]^\lambda-\int_{l(r-\|x^\star\|)^2}^{lr^2}\left[1-\left(\sqrt{\frac{t}{lr^2}}\right)^d\right]^\lambda dt\\
&\geq lr^2\frac{\Gamma(\frac{d+2}{d})\Gamma(\lambda+1)}{\Gamma(\lambda+1+\frac{2}{d})}-l(r^2-(r-\|x^\star\|)^2)\left[1-\left(\frac{r-\|x^\star\|}{r}\right)^d\right]^\lambda\\
&\geq\frac{1}{2}lr^2\Gamma(\frac{d+2}{d})\lambda^{-2/d}\text{ for }\lambda\text{ sufficiently large.}
\end{aligned}$$

where the last inequality follows by Stirling's approximation applied to the first term and because the second term is $o(\lambda^{-2/d})$ as in previous proof.
This concludes the proof of the first part of the lemma. Let us now treat the case of the conditional inequality. Using the same first identity as above we have

$$\begin{aligned}
&\mathbb{E}_{X_1,\ldots,X_\lambda\sim U(B(0,r))}\left[f\left(X_{(1)}\right)\mid f(X_{(\mu_\lambda+1)})=h\right]\\
&=\int_0^\infty\mathbb{P}_{X_1,\ldots,X_\lambda\sim U(B(0,r))}\left[f\left(X_{(1)}\right)\geq t\mid f(X_{(\mu_\lambda+1)})=h\right]dt
\end{aligned}$$

Remark 19. *Note that if we sample λ independent variables $X_1\ldots X_\lambda$ while conditioning on $f(X_{(\mu+1)})=h$ and keep only the μ-best variables X_i such that $f(X_i)\leq h$, this is exactly equivalent to sampling directly $X_1\ldots X_\mu$ from the h-level set. This result was justified and used in Meunier et al. [2020a] in their proofs.*

Hence we obtain

$$\begin{aligned}
&\int_0^\infty\mathbb{P}_{X_1,\ldots,X_\lambda\sim U(B(0,r))}\left[f\left(X_{(1)}\right)\geq t\mid f(X_{(\mu_\lambda+1)})=h\right]dt\\
&=\int_0^\infty\mathbb{P}_{X\sim U(S_h)}[f(X)\geq t]^{\mu_\lambda}dt
\end{aligned}$$

Using Lemma 19, we get:

$$\begin{aligned}
&\int_0^\infty\mathbb{P}_{X\sim U(S_h)}[f(X)\geq t]^{\mu_\lambda}dt\\
&\geq\int_0^\infty\mathbb{P}_{X\sim U(S_h)}\left[l\|X-x^\star\|^2\geq t\right]^{\mu_\lambda}dt\\
&\geq\int_0^\infty\mathbb{P}_{X\sim U(B(x^\star,\sqrt{\frac{h}{l}}))}\left[l\|X-x^\star\|^2\geq t\right]^{\mu_\lambda}dt
\end{aligned}$$

where the last inequality follows from the inclusion $S_h \subset B(x^\star, \sqrt{\frac{h}{l}})$, which is also a consequence of Lemma 19. We then get

$$
\begin{aligned}
&\int_0^\infty \mathbb{P}_{X\sim U(B(x^\star,\sqrt{\frac{h}{l}}))}\left[l\|X-x^\star\|^2 \geq t\right]^{\mu_\lambda} dt \\
&\quad = \int_0^h \mathbb{P}_{X\sim U(B(x^\star,\sqrt{\frac{h}{l}}))}\left[l\|X-x^\star\|^2 \geq t\right]^{\mu_\lambda} dt \\
&\quad = \int_0^h \left[1-\left(\sqrt{\frac{t}{h}}\right)^d\right]^{\mu_\lambda} dt \\
&\quad = h\frac{\Gamma(\frac{d+2}{d})\Gamma(\mu_\lambda+1)}{\Gamma(\mu_\lambda+1+2/d)} \\
&\quad \geq \frac{1}{2}h\Gamma(\frac{d+2}{d})\mu_\lambda^{-2/d} \text{ for } \lambda \text{ sufficiently large.}
\end{aligned}
$$

□

This lemma, along with Lemma 21, proves that for any function satisfying Assumption 7, its rate of convergence is exponentially dependent on the dimension and of order $\lambda^{-2/d}$ where λ is the number of points sampled to estimate the optimum.

Remark 20 (Convergence of the distance to the optimum)**.** *It is worth noting that, thanks to Lemma 19, the convergence rates are also valid for the square distance to the optimum $x^\star$.*

J.5 Convergence rates for the μ-best averaging approach

In the next section we focus on the case where we average the μ best samples among the λ samples. We first prove a lemma when the sampling is conditional on the $(\mu+1)$-th value.

Lemma 23. *Let f be a function satisfying Assumption 7. There exists a constant $C_3 > 0$ such that for all $h \in [0, \max f]$ and λ and μ two integers such that $1 \leq \mu \leq \lambda - 1$, we have the following conditional upper bound:*

$$
\mathbb{E}_{X_1,\ldots X_\lambda\sim U(B(0,r))}\left[f(\bar{X}_{(\mu)})|f(X_{(\mu+1)})=h\right] \leq C_3\left(\frac{h}{\mu}+h^{\alpha-1}\right).
$$

Proof. We first decompose the expectation as follows.

$$
\begin{aligned}
&\mathbb{E}_{X_1,\ldots X_\lambda\sim U(B(0,r))}\left[f(\bar{X}_{(\mu)})|f(X_{(\mu+1)}))=h\right] \\
&\quad = \mathbb{E}_{X_1,\ldots X_\mu\sim U(S_h)}\left[f(\bar{X}_\mu)\right] \\
&\quad = \mathbb{E}_{X_1,\cdots,X_\mu\sim U(S_h)}\left[\|\bar{X}_\mu-x^\star\|^2_{\mathbf{H}}\right] \qquad \text{(J.3)} \\
&\quad + \mathbb{E}_{X_1,\cdots,X_\mu\sim U(S_h)}\left[\|\bar{X}_\mu-x^\star\|^\alpha_{\mathbf{H}}\varepsilon(\bar{X}_\mu-x^\star)\right] \qquad \text{(J.4)}
\end{aligned}
$$

where we have use the same argument as in Remark 19 in the first equality. We will treat the terms (J.3) and (J.4) independently. We first look at (J.3). We have the following "bias-variance" decomposition.

$$\begin{aligned}\mathbb{E}_{X_1,\cdots,X_\mu\sim U(S_h)}\|\bar{X}_\mu - x^\star\|^2_{\mathbf{H}} =&(1-\frac{1}{\mu})\|\mathbb{E}_{X\sim U(S_h)}X - x^\star\|^2_{\mathbf{H}}\\ &+\frac{1}{\mu}\mathbb{E}_{X\sim U(S_h)}\|X-x^\star\|^2_{\mathbf{H}}\end{aligned}$$

We will use Lemma 20. We have $A_h \subset S_h \subset B_h$. Hence for the variance term

$$\frac{1}{\mu}\mathbb{E}_{X\sim U(S_h)}\|X-x^\star\|^2_{\mathbf{H}} \leq \frac{1}{\mu}\mathbb{E}_{X\sim U(S_h)}\phi_+(h)^2 \leq \frac{\phi_+(h)^2}{\mu} \sim_0 \frac{h}{\mu}.$$

where $\sim_0$ means "is equivalent to ... when $h \to 0$, in other words, $u(h) \sim_0 v(h)$ iff $\frac{u(h)}{v(h)} \to 0$ as $h \to 0$. For the bias term, recall that

$$\mathbb{E}_{X\sim U(S_h)}[X-x^\star] = \frac{1}{\mathrm{vol}(S_h)}\int_{S_h}(x-x^\star)dx.$$

We then have by inclusion of sets

$$\mathrm{vol}(A_h) \leq \mathrm{vol}(S_h) \leq \mathrm{vol}(B_h)$$

Note that the volume of the d-dimensional ellipsoid B_h satisfies $\mathrm{vol}(B_h) = \phi_+(h)^d\frac{\omega_d}{\det(\mathbf{H})}$ with $\omega_d = \mathrm{vol}(B(0,1))$ and similarly for A_h. From this we deduce by the squeeze theorem that

$$\mathrm{vol}(S_h) \sim \frac{\omega_d h^{d/2}}{\det(\mathbf{H})}.$$

We now decompose the integral

$$\begin{aligned}\int_{S_h}(x-x^\star)dx &= \int_{A_h}(x-x^\star)dx + \int_{S_h\setminus A_h}(x-x^\star)dx\\ &= \int_{S_h\setminus A_h}(x-x^\star)dx\end{aligned}$$

(because A_h is an ellipsoid centered at $x^\star$ hence the integral of $x - x^\star$ over it is 0). We then upper-bound using the triangle inequality for the $\mathbf{H}-$norm:

$$\begin{aligned}\|\int_{S_h\setminus A_h}(x-x^\star)dx\|_{\mathbf{H}} &\leq \int_{S_h\setminus A_h}\|x-x^\star\|_{\mathbf{H}}dx\\ &\leq \phi_+(h)\mathrm{vol}(S_h\setminus A_h)\\ &= \phi_+(h)(\mathrm{vol}(S_h)-\mathrm{vol}(A_h))\\ &\leq \phi_+(h)(\mathrm{vol}(B_h)-\mathrm{vol}(A_h))\end{aligned}$$

$$\sim d\frac{\omega_d}{\det(\mathbf{H})}\frac{m+M}{2}h^{d/2}h^{(\alpha-1)/2}$$

For the last equivalent, we used a Taylor expansion for the volume of A_h and B_h. We conclude that there exist $h_1 > 0$ and a constant $C > 0$ not depending on λ and μ such that for $h \leq h_1$,

$$\|\mathbb{E}_{X\sim U(S_h)}[X] - x^\star\|_{\mathbf{H}}^2 \leq Ch^{\alpha-1}$$

Since h is upper bounded by $\max f$, the previous inequality can be extended to $h \in [0, \max f]$, with a possibly larger constant still not depending on λ and μ. Let us now upper bound the remainder term (J.4). As $\varepsilon \leq M$ by assumption, we can write

$$\begin{aligned}\mathbb{E}_{X_1,\cdots,X_\mu\sim U(S_h)}&\left[\|\bar{X}_\mu - x^\star\|_{\mathbf{H}}^\alpha \varepsilon(\bar{X}_\mu - x^\star)\right]\\&\leq M\mathbb{E}_{X_1,\cdots,X_\mu\sim U(S_h)}\left[\|\bar{X}_\mu - x^\star\|_{\mathbf{H}}^\alpha\right]\end{aligned}$$

We have $X_1, \cdots, X_\mu \in S_h \subset B_h$ hence by the convexity of B_h (which is a ball for the $\mathbf{H}$-norm) we also have $\bar{X}_\mu \in B_h$ and thus, for h sufficiently small, we have:

$$\|\bar{X}_\mu - x^\star\|_{\mathbf{H}} \leq \phi_+(h).$$

Note that $\phi_+(h) \sim_0 \sqrt{h}$ thus, for h sufficiently small, $\|\bar{X}_\mu - x^\star\|_{\mathbf{H}} \leq 1$ almost surely, hence, as $\alpha > 2$

$$\|\bar{X}_\mu - x^\star\|_{\mathbf{H}}^\alpha \leq \|\bar{X}_\mu - x^\star\|_{\mathbf{H}}^2$$

almost surely. Since h is upper bounded, we have the existence of a constant $C' > 0$ not depending on λ and μ, such that for all $h \in [0, \max f]$,

$$\|\bar{X}_\mu - x^\star\|_{\mathbf{H}}^\alpha \leq C'\|\bar{X}_\mu - x^\star\|_{\mathbf{H}}^2$$

Thus we can upper bound the remainder with the same bounds as the one for the main term (up to constants), for any $h \in [0, \max f]$. We now group the "main" term and remainder term to get the existence of a constant $C_3 > 0$ not depending on λ and μ such that for all $h \in [0, \max f]$,

$$\mathbb{E}_{X_1,\ldots X_\lambda\sim U(B(0,r))}\left[f(\bar{X}_{(\mu)})|f(X_{(\mu+1)}) = h\right] \leq C_3\left(\frac{h}{\mu} + h^{\alpha-1}\right) \quad .$$

□

We are now set to prove our main result, which is an upper convergence rate for the μ-best approach. This is the main result of the paper.

Theorem 33. *Let f be a function satisfying Assumption 7. Let $(\mu_\lambda)_{\lambda\in\mathbb{N}}$ be a sequence of integers such that $\forall\lambda \geq 2$, $1 \leq \mu_\lambda \leq \lambda - 1$ and $\mu_\lambda \to \infty$. Then, there exist two constants $C, C' > 0$ and $\tilde{\lambda} \in \mathbb{N}$ such that for $\lambda \geq \tilde{\lambda}$, we have the upper bound:*

$$\mathbb{E}_{X_1,\ldots,X_\lambda \sim U(B(0,r))}\left[f(\bar{X}_{(\mu_\lambda)})\right] \leq C\frac{\mu_\lambda^{\frac{2(\alpha-1)}{d}}}{\lambda^{\frac{2(\alpha-1)}{d}}} + C'\frac{\mu_\lambda^{\frac{2}{d}-1}}{\lambda^{\frac{2}{d}}} \quad .$$

In particular if $\mu_\lambda \sim C''\lambda^{\frac{2(\alpha-2)}{d+2(\alpha-2)}}$ for some $C'' > 0$, we obtain:

$$\mathbb{E}_{X_1,\ldots,X_\lambda \sim U(B(0,r))}\left[f(\bar{X}_{(\mu_\lambda)})\right] \leq C'''\lambda^{-\frac{2(\alpha-1)}{d+2(\alpha-2)}}$$

for some $C''' > 0$ independent of λ.

We note that $\frac{\mu}{\lambda} \to 0$ as $\lambda \to 0$. This makes sense intuitively: we average points in a sublevel set, which makes sense only if, asymptotically in λ, this sublevel set shrinks to a neighborhood of the optimum.

Proof. The random variable $f(X_{(\mu_\lambda+1)})$ takes its values in $[0, \max f]$ almost surely. As such, thanks to Lemma 23, there exists a constant $C_3 > 0$ such that for all $\lambda \geq 1$:

$$\begin{aligned}
\mathbb{E}\left[f(\bar{X}_{(\mu_\lambda)})\right] &= \mathbb{E}\left[\mathbb{E}\left[f(\bar{X}_{(\mu_\lambda)}) \mid f(X_{(\mu_\lambda+1)})\right]\right] \\
&\leq \mathbb{E}\left[C_3\left(\frac{1}{\mu_\lambda}f(X_{(\mu_\lambda+1)}) + f(X_{(\mu_\lambda+1)})^{\alpha-1}\right)\right] \\
&= C_3\left(\frac{1}{\mu_\lambda}\mathbb{E}\left[f(X_{(\mu_\lambda+1)})\right] + \mathbb{E}\left[f(X_{(\mu_\lambda+1)})^{\alpha-1}\right]\right)
\end{aligned}$$

Let us first bound $\mathbb{E}\left[f(X_{(\mu_\lambda+1)})\right]$. Thanks to Lemma 22, there exist a constant $C_2 > 0$ and $\lambda_2 \in \mathbb{N}$ such that:

$$\begin{aligned}
\mathbb{E}\left[f(X_{(\mu_\lambda+1)})\right] &\leq \frac{\mu_\lambda^{2/d}}{C_2}\mathbb{E}\left[\mathbb{E}\left[f(X_{(1)}) \mid f(X_{(\mu_\lambda+1)})\right]\right] \\
&= \frac{\mu_\lambda^{2/d}}{C_2}\mathbb{E}\left[f(X_{(1)})\right]
\end{aligned}$$

Thanks to Lemma 21, there exists a constant $C_0 > 0$ and an integer $\lambda_0 \in \mathbb{N}$ such that for all integers $\lambda \geq \lambda_0$:

$$\mathbb{E}_{X_1,\cdots,X_\lambda \sim U(B(0,r))}\left[f\left(X_{(1)}\right)\right] \leq C_0\lambda^{-\frac{2}{d}} \quad .$$

Then finally for $\lambda \geq \max(\lambda_0, \lambda_2)$

$$\mathbb{E}\left[f(X_{(\mu_\lambda+1)})\right] \leq \frac{C_0}{C_2}\frac{\mu_\lambda^{2/d}}{\lambda^{2/d}} \quad .$$

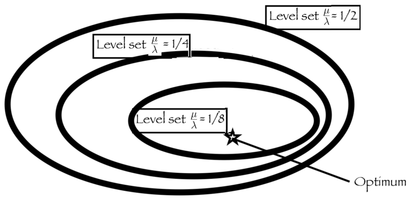


Figure J.1: Assume that we consider a fixed ratio μ/λ and that λ goes to ∞. The average of selected points, in an unweighted setting and with uniform sampling, converges to the center of the area corresponding to the ratio μ/λ: we will not converge to the optimum if that optimum is not the middle of the sublevel. This explains why we need $\mu/\lambda \to 0$ as $\lambda \to \infty$: we do not want to stay at a fixed sublevel.

For the term $\mathbb{E}\left[f(X_{(\mu_\lambda+1)})^{\alpha-1}\right]$, we write thanks to Lemma 22

$$\mathbb{E}\left[f(X_{(\mu_\lambda+1)})^{\alpha-1}\right] \leq \frac{\mu_\lambda^{2(\alpha-1)/d}}{C_2^{\alpha-1}}\mathbb{E}\Big[\mathbb{E}\left[f(X_{(1)}) \mid f(X_{(\mu_\lambda+1)})\right]^{\alpha-1}\Big].$$

Then, by Jensen's inequality for the conditional expectation, we get

$$\mathbb{E}\left[f(X_{(\mu_\lambda+1)})^{\alpha-1}\right] \leq \frac{\mu_\lambda^{2(\alpha-1)/d}}{C_2^{\alpha-1}}\mathbb{E}\left[f(X_{(1)})^{\alpha-1}\right].$$

Similarly to Lemma 21, by replacing $\|X - x^\star\|^2$ by $\|X - x^\star\|^{2(\alpha-1)}$, one can show $\mathbb{E}\left[f(X_{(1)})^{\alpha-1}\right] \leq C_3'\lambda^{-2(\alpha-1)/d}$ for some $C_3' > 0$ independent of λ. We thus get $\mathbb{E}\left[f(X_{(\mu_\lambda+1)})^{\alpha-1}\right] \leq C\frac{\mu_\lambda^{2(\alpha-1)/d}}{\lambda^{2(\alpha-1)/d}}$ for some $C > 0$ independent of λ, which, combined with the above bound on $\mathbb{E}\left[f(X_{(\mu_\lambda+1)})\right]$, concludes the proof of the main bound.
To conclude for the final bound, it suffices to notice that this choice of μ_λ ensures that the two terms in the upper bound are of the same order. $\square$

This theorem gives an asymptotic upper rate of convergence for the algorithm that consists in averaging the best samples to optimize a function with parallel evaluations. The proof of the optimality of the rate is left as further work. We also remark that the selection ratio depends on the dimension and goes to 0 as $\lambda \to \infty$. It sounds natural since the level sets might be assymetric and then keeping a constant selection rate would give a biased estimate of the optimum (see Figure J.1). However, the choice proposed for μ is the best one can make with regards to the upper bound we obtained. We make two important remarks about the theorem.

Remark 21 (Comparison with random search)**.** *The asymptotic rate obtained for the μ-best averaging approach is of order $\lambda^{-\frac{2(\alpha-1)}{d+2(\alpha-2)}}$, which is strictly better than the $\lambda^{-2/d}$ rate obtained with random search, as soon as $d > 2$ (because $\alpha > 2$). This theorem then proves our claim on a wide range of functions.*

Remark 22 (Comparison with Meunier et al. [2020a])**.** *Meunier et al. [2020a] obtained a rate of order λ^{-1} for the sphere function. This rate is better than the one described in Theorem 33. This* *comes from the bias term in Lemma 23. Indeed for the sphere function, sublevel sets are symmetric, hence the bias term equals* 0*, which is not the case in general for functions satisfying Assumption 7. In this paper we are able to deal with potentially non symmetric functions. One can remark, that if the sublevel sets are symmetric the bias term vanishes and we recover the rate of Meunier et al. [2020a].*

J.6 Handling wider classes of functions

The results we proved are valid for functions satisfying Assumption 7. In particular, the functions are supposed to be regular and have a unique optimum point. In this section, we propose to extend our results to wider classes of functions.

J.6.1 Invariance by Composition with Non-Decreasing Functions

Mathematical results are typically proved under some smoothness assumptions: however, algorithms enjoying some invariance to monotonic transformations of the objective functions do converge on wider spaces of functions as well Akimoto et al. [2020]. Since the method is based on comparison between the samples, the rank is invariant when the function f is composed with a strictly increasing function g. Let f be a function satisfying Assumption 7 and g be a strictly increasing function. Consider $h = g \circ f$. Then h admits a unique minimum $x^\star$ coinciding with the one of f. As such, the expectation $\mathbb{E}_{X_1,\ldots X_\lambda \sim U(B(0,r))}\left[\|X_{(\mu)} - x^\star\|^2\right]$ satisfies the same rates than Theorem 33. This an immediate consequence of Lemma 19. In particular, using the square distance criteria, the rate are preserved even for potentially non regular functions. For example, our theorem can be adapted to convex piecewise-linear functions, compositions of quadratic functions with non-differentiable increasing functions, and many others. Results based on surrogate models are not applicable here.

J.6.2 Beyond Unique Optima: the Convex Hull trick, Revisited

One of the drawbacks of averaging strategies is that they do not work when there are two basins of optima. For instance, if the two best points $x_{(1)}$ and $x_{(2)}$ have objective values close to those of two distinct optima $x^\star, y^\star$ respectively then averaging $x_{(1)}$ and $x_{(2)}$ may result in a point whose objective value is close to neither. However, in the presence of quasi-convexity this can be countered. It thus makes sense to take into account the possible obstructions to the quasi-convexity of the function and try to counter these, while still maintaining the same basic algorithm as in the case of a unique optimum. Meunier et al. [2020a] proposed to take into account contradictions to quasi-convexity by restricting the number μ of points used in the averaging. Based on their ideas, we propose the following heuristic.

Let us fix the number of initially selected points equal to $\mu_{\max}$. Let $x_{(1)}, \ldots, x_{(\mu_{\max})}$ be these points ranked from best to worst. Define $S_i = (x_{(1)}, \ldots, x_{(i)})$ and C_i the interior of the convex hull of S_i. Assume that there is no tie in fitness values, that is no $i \neq j$ such that $f(x_i) = f(x_j)$. Given $\mu_{\max}$, choose μ maximal such that

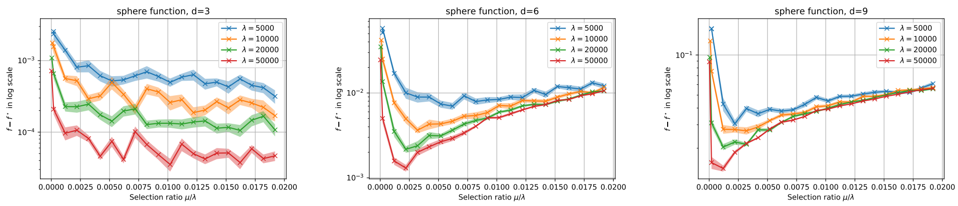
$$\forall i \leq \mu, x_{(i)} \notin C_i. \tag{J.5}$$

One can remark that $x_{(\mu)} \in C_\mu \Rightarrow f$ is not quasi-convex on C_μ. However, this may not detect all cases in which f is not quasi-convex on C_μ. More generally,

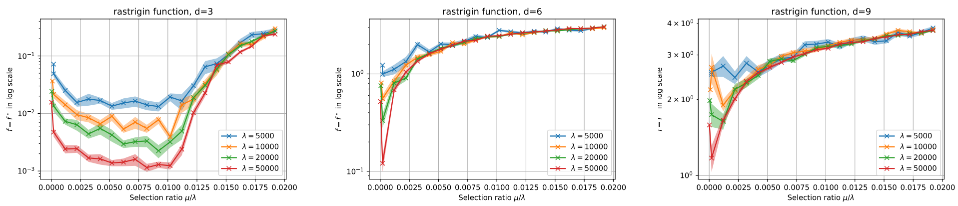
$$\exists j > \mu - 1,\ x_{(j)} \in C_\mu \Rightarrow f \text{ is not quasi-convex on } C_\mu. \tag{J.6}$$

If such a j is not μ, Eq. (J.5) does not detect the non-quasiconvexity: therefore, (J.6) detects more non-quasiconvexities than Eq. (J.5).

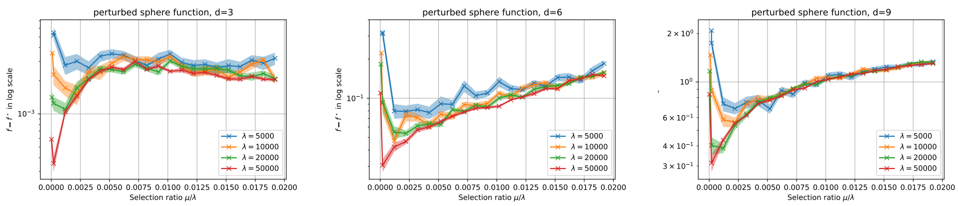
Therefore we choose μ maximal such that for all $i < \mu, j > i, x_{(j)} \notin C_i$. This heuristic leads to a choice of average which is "consistent" with the existence of multiple basins.



Sphere function



Rastrigin function



Perturbed sphere function

Figure J.2: Average regret $f(\bar{X}_{(\mu)}) - f(x^\star)$ in logarithmic scale in function of the selection ratio μ/λ for different values of $\lambda \in \{5000, 10000, 20000, 50000\}$. The experiments are run on Sphere, Rastrigin and Perturbed Sphere function for different dimensions $d \in \{3, 6, 9\}$. All results are averaged over 30 independent runs. We observe, consistently with our theoretical results and intuition, that (i) the optimal $r = \frac{\mu}{\lambda}$ decreases as d increases (ii) we need a smaller r when the function is multimodal (Rastrigin) (iii) we need a smaller r in case of dissymmetry at the optimum (perturbed sphere).

J.7 Experiments

We divide the experimental section in two parts. In a first part, we focus on validating theoretical findings, then we compare with existing optimization methods.

J.7.1 Validation of Theoretical Findings

In this section, we will assume that $r = 1$ and that the optimum x^* will be sampled uniformly in the ball of radius 0.9. We compare results on the following functions:

1. Sphere function:

$$f(x) = \sum_{i=1}^{d} (x_i - x_i^\star)^2$$

2. Rastrigin function:

$$f(x) = \sum_{i=1}^{d} (x_i - x_i^\star)^2 + 1 - \cos\left(2\pi(x_i - x_i^\star)\right)$$

3. Perturbed sphere function:

$$f(x) = \sum_{i=1}^{d} (x_i - x_i^\star)^2 + \left(\sum_{i=1}^{d} g(x_i - x_i^\star)\right)^3$$

   with $g(x) = x$ if $x > 0$ and $-2x$ otherwise. This function has highly non symmetric sublevel sets, but still satisfies Assumption 7.

We plotted in Figure J.2 the regret $f(\bar{X}_{(\mu)}) - f(x^\star)$ as a function of μ/λ for different dimensions d and number of samples λ. The experiments are averaged over 30 runs. We remark for instance on the Rastrigin function that for the μ-best averaging approach to be better than random search, we need a very large number of samples as the dimension increases. Overall, these plots validate our theoretical findings that averaging a few best points leads to a better regret than only taking the best one.

J.7.2 Comparison with Other Methods

In this section, we compare averaging strategies with other standard strategies, using the Nevergrad library Rapin and Teytaud [2018]. Figure J.3 presents experimental results based on Nevergrad. Instead of the uniform sampling used in the theoretical results and the previous experimental validation, we use Gaussian sampling in this set of experiments. Following the notation from Meunier et al. [2020a], we consider distinct averaging prefixes:

- `AvgXX` = method `XX`, plus averaging of the $\mu = \lambda/(1.1^d)$ best points in dimension d.
- `HAvgXX` = = method `XX`, plus averaging of the $\lambda/(1.1^d)$ best points, restricted by the convex hull trick (Section J.6.2).

Many other methods are included: we refer to Rapin and Teytaud [2018] for more information. Recently, Cauwet et al. [2019], Meunier et al. [2020c] pointed out that when the optimum is randomly drawn from a standard normal distribution, we should use rescaling methods for focusing closer to the center in high dimensional setting. Several such methods have been proposed:

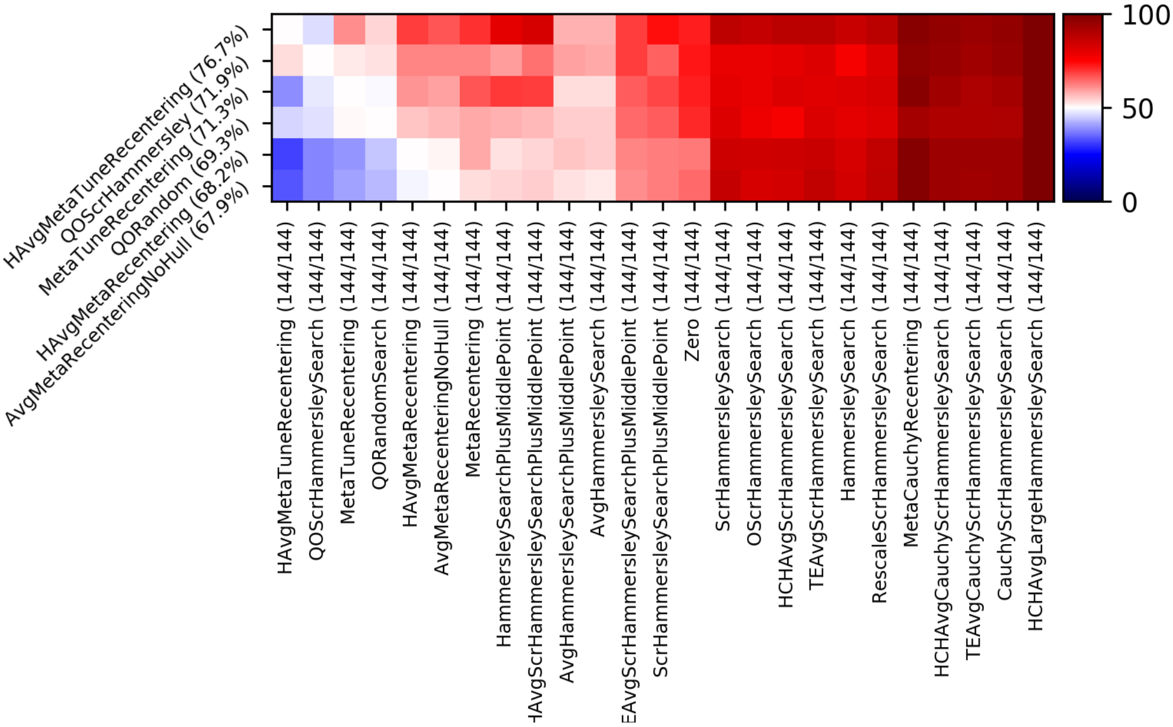


Figure J.3: Experimental results: row A and col B presents the frequency (over all 144 test cases) at which A outperforms B in terms of average loss. Then rows are sorted per average winning rate and we keep the 6 best ones. Zero is a naive method just choosing zero: we see that, consistently with Cauwet et al. [2019], many methods are worse than that when the dimension is huge compared to the budget.

- `QOXX` = method `XX`, plus quasi-opposite sampling Rahnamayan et al. [2007], i.e. each time we draw x with $\mathcal{N}$, we also use $-rx$ where r is uniformly independently drawn in $(0, 1)$.
- `XXPlusMiddlePoint` = method `XX`, except that there is one point forced at the center of the domain.
- `MetaRecentering` Cauwet et al. [2019]: rescaling $\sigma = (1 + \log(n))/(4\log(d))$, i.e. we randomly draw with $\sigma \times \mathcal{N}(0, I_d)$ instead of $\mathcal{N}(0, I_d)$.
- `MetaTuneRecentering` Meunier et al. [2020c]: rescaling $\sigma = \sqrt{\log(\lambda)/d}$, i.e. we randomly draw with $\sigma \times \mathcal{N}(0, I_d)$ instead of $\mathcal{N}(0, I_d)$.

Experimental setup. We measure the simple regret and compare methods by average frequency of win against other methods. For each test case, we randomly draw the optimum as $\mathcal{N}(0, I_d)$ (multivariate standard Gaussian), with different budgets λ in $\{30, 100, 300, 1000, 3000, 10000, 100000\}$ and dimensions d in $\{3, 10, 30, 100, 300, 1000, 3000\}$. Due to their time of evaluation, we did not run the cases with both $d = 3000$ and $\lambda = 100000$. We evaluated on 3 different functions: the sphere function, the Griewank function, and the Highly Multimodal function. Previous results Bousquet et al. [2017] from the literature have already shown that replacing random sampling by scrambled Hammersley sampling (i.e. modern low discrepancy sequences compatible with high dimension) leads to better results.

Analysis of results. Analyzing the table results from Figure J.3, we observe that

- Averaging performs well overall: `AvgXX` is better than `XX`;
- The quasi-convex trick from Section J.6.2 does work: `HAvgXX` is better than `AvgXX`;
- The rescaling strategy from Meunier et al. [2020c] outperforms the ones in Cauwet et al. [2019] (`MetaTuneRecentering` better than `MetaRecentering` or than `PlusMiddlePoint`) which are already better than standard quasi-random sampling. Quasi-Opposite sampling is also competitive.

We also include various methods present in the platform, including those which are based on Cauchy or Hammersley without scrambling (Hammersley in the name without "Scr" prefix), or sophisticated uses of convex hulls for estimating the location of the optimum (HCH in the name).

J.8 Conclusion

We proved that averaging $\mu > 1$ points rather than picking up the best works even for non quadratic functions, in the sense that the convergence rate is better than the one obtained just by picking up the best point. We also proved faster rates than methods based on meta-models (such as Rudi et al. [2020]) unless the objective function is very smooth and low dimensional. We also show that our results cover a wider family of functions (Section J.6.1). We also propose a rule for choosing μ, depending on λ and the dimension. This shows that the optimal μ/λ ratio decreases to 0 as the dimension goes to infinity, which is confirmed by Fig. J.2. We also note, by comparing with Meunier et al. [2020a], that the optimal ratio should be smaller (Fig. J.1), which is confirmed by our experiments on the perturbed sphere (Fig. J.2). We also propose a method for adapting this μ, by automatically detecting non-quasi-convexity and reducing it: and prove that it detects more non-quasiconvexities than the method proposed in Meunier et al. [2020a]. Finally, we validate the approach on a reproducible open-sourced platform (Fig. J.3).

Further Work

Using density-dependent weights as in Teytaud and Teytaud [2009] should allow us to get rid of the constraint $||x^*|| < r$ using a Gaussian sampling instead of a uniform sampling. Better rates might be obtained with rank-dependent weights as in Arnold et al. [2009]. We also leave as further work the proof of the optimality of the rate for this strategy. Moreover, we also believe better rates can be obtained for smoother functions, and leave this study for further work. The case of noisy objective functions Arnold and Beyer [2006] is critical. The study is harder, and good evolutionary algorithms use large populations, making the overall algorithm closer to a small number of one-shot optimization algorithms: actually, some fast algorithms use mainly learning Astete-Morales et al. [2015], Coulom [2011], Audet et al. [2018]. Population controlHellwig and Beyer [2016] is successful and its last stage looks exactly like a one-shot optimization method.

K Black-Box Optimization Revisited: Improving Algorithm Selection Wizards through Massive Benchmarking

Existing studies in black-box optimization suffer from low generalizability, caused by a typically selective choice of problem instances used for training and testing of different optimization algorithms. Among other issues, this practice promotes overfitting and poor-performing user guidelines. We address this shortcoming by introducing in this work a general-purpose algorithm selection wizard that was designed and tested on a previously unseen breadth of black-box optimization problems, ranging from academic benchmarks to real-world applications, from discrete over numerical to mixed-integer problems, from small to very large-scale problems, from noisy over dynamic to static problems, etc. Not only did we use the already very extensive benchmark environment available in Nevergrad, but we also extended it significantly by adding a number of additional benchmark suites, including Pyomo, Photonics, LSGO, and MuJoCo. Our wizard achieves competitive performance on all benchmark suites. It significantly outperforms previous state-of-the-art algorithms on some of the suites, including YABBOB and LSGO. Its excellent performance is obtained without any task-specific parametrization.

The algorithm selection wizard, all of its base solvers, as well as the benchmark suites are available for reproducible research in the open-source Nevergrad platform.

K.1 Introduction: State of the Art

Many real-world optimization challenges are effectively black-box problems; i.e., the main source of information when solving them is the evaluation of solution candidates. These evaluations often require simulations or even physical experiments. Black-box optimization methods are thus widely applied in practice, with a particularly growing impact in machine learning Salimans et al. [2016], Wang et al. [2020b], to the point that they are considered a key research area of artificial intelligence. Black-box optimization algorithms are typically easy to implement and easy to adjust to different problem types. To achieve peak performance, however, proper algorithm selection and configuration are key, since black-box optimization algorithms have complementary strengths and weaknesses Rice [1976], Kerschke et al. [2019]. But while automated algorithm selection has become standard in SAT solving Xu et al. [2008] and AI planning Vallati et al. [2015], a manual selection and configuration of the algorithms—often entirely based on users previous experience and not necessarily on broader performance data—is still predominant in the broader black-box optimization context. To reduce the bias inherent to such manual choices, and to support the automation of algorithm selection and configuration, sound comparisons of the different black-box optimization approaches are needed. Existing benchmarking suites, however, are rather selective in the problems they cover. This leads to specialized algorithm frameworks whose performance suffer from poor generalizability. We address this flaw in black-box optimization by presenting a high-performing algorithm selection wizard, ABBO (Automated Black-Box Optimization). ABBO uses very basic information about the problem and the available computational resources to select one or several algorithms, which are run for the allocated budget of function evaluations. The wizard was developed within the Nevergrad platform Rapin and Teytaud [2018], which we have significantly extended for this work to obtain an even broader set of benchmark problems.

In summary, our key contributions are as follows: **(1) Algorithm Selection Wizard ABBO:** Our algorithm selection technique, ABBO can be seen as an extension of the Shiwa wizard presented in Liu et al. [2020]. The wizard takes as input information on the *problem* (the dimension of the search domain, the type and range of each variable, and their order), the *presence of noise* in the evaluation (but not its intensity), and the *computational resources* that are available to solve the problem (the budget and the degree of parallelism, i.e., number of solution candidates that can be evaluated simultaneously). Based on this input, the wizard outputs one or several algorithms that it suggests to execute on the given problem. ABBO uses three types of selection techniques: *passive algorithm selection* (choosing an algorithm as a function of a priori available features Liu et al. [2020]), *active algorithm selection* (a bet-and-run strategy which runs several algorithms for some time and stops all but the strongest after a predefined number of evaluations Mersmann et al. [2011], Pitzer and Affenzeller [2012], Fischetti and Monaci [2014], Malan and Engelbrecht [2013], Muñoz et al. [2015], Cauwet et al. [2016], Kerschke et al. [2019]), and *chaining* (running several algorithms in turn, in an a priori defined order Molina et al. [2009]).

Our wizard selects from and combines a very large number of base algorithms, among them algorithms suggested in Virtanen et al. [2019], Hansen and Ostermeier [2001], Storn and Price [1997], Powell [1964, 1994], Liu et al. [2020], Hellwig and Beyer [2016], Artelys [2015], Doerr et al. [2017, 2021]. We compare the performance of ABBO to all these algorithms, as well as to its predecessor Shiwa, and to all other algorithms available in Nevergrad.

(2) Benchmark Collection: By integrating a number of additional benchmark suites into the Nevergrad platform, we obtain a huge benchmark collection that covers a previously unseen breadth of black-box optimization problems, ranging from academic benchmarks to real-world applications, from discrete over numerical to mixed-integer problems, from small to very large-scale problems, from noisy over dynamic to static problems, etc.

Structure of the Paper: We motivate in Sec. K.2 why we have chosen to develop ABBO within the Nevergrad benchmarking environment and how we have extended it. Sec. K.3 summarizes ABBO and discusses differences to previous versions. Experimental results can be found in Sec. K.4. Sec. K.5 concludes our paper with an outlook to future work.

Availability of Data and Code: Our algorithm selection wizard ABBO, its base solvers, and the benchmark collection have all been merged to the **main Nevergrad master**, where they are available for reproducible, open-source research. Nevergrad periodically reruns all algorithms and makes all data available on the **public dashboard** Rapin and Teytaud [2020]. Note that ABBO is called **NGOpt8** within the Nevergrad environment, to allow for better version control in the latter: NGOpt is the name of the latest version of the optimization wizard, the current version (July 2021) is NGOpt14. Since Nevergrad is developing at fast pace, we have saved a **frozen version of its code base**. Together with the **performance data** used for this paper and the **videos** illustrating the performance of the policies learned for MuJoCo, this code is available on zenodo Meunier et al. [2021d].

Table K.1: Properties of selected benchmark collections (details in the main text). "+" means that the feature is present, "-" that the feature is missing, and "NA" means that it is not applicable.

Testbed	BBOB	MuJoCo	LSGO	BBComp	Nevergrad
Large scale	-	NA	+	-	+
Translations	+	NA	+	+	+
Symmetrizations / rotations	+	NA	+	-	+
One-line reproducibility	-	-	-	-	+
Periodically automated dashboard	NA	NA	NA	NA	+
Complex or real-world	-	+	-	+	+
Available open source and no licensing issues	+	-	+	+	+
Ask/tell/recommend framework	-	NA	+	+	+
Human excluded / client-server	-	-	-	+	-

K.2 Sound Black-Box Optimization Benchmarking

We discuss in this section which features we consider desirable for sound benchmarking, and how different suites address these. This discussion motivates our decision to design and to evaluate ABBO within the Nevergrad environment Rapin and Teytaud [2018]. Tab. K.1 summarizes how some of the common benchmarking environments address the properties discussed below.

K.2.1 Desirable Properties for Sound Benchmarking

Generalization: The most obvious issue in terms of generalization is the statistical one: we need sufficiently many experiments for conducting valid statistical tests and for evaluating the robustness of algorithms' performance. However, this is probably not the main issue. A biased benchmark, excluding large parts of the real world needs, leads to biased conclusions, no matter how many experiments we perform. Inspired by Recht et al. [2018] in the case of image classification, and similar to the spirit of cross-validation for supervised learning, we use a much broader collection of benchmark problems for evaluating algorithms in an unbiased manner. Another subtle issue in terms of generalization is the case of instance-based choices of (hyper-)parameters: an experimenter modifying the algorithm or its parameters specifically for each instance can easily achieve considerable performance improvements. In this paper, we consider that only the following problem properties are known in advance (and can hence be used for algorithm selection and configuration): the dimension of the domain, the type and range of each variable, their order, the presence of noise (but not its intensity), the budget, and the degree of parallelism (i.e., number of solution candidates that can be evaluated simultaneously). To mitigate the common risk of over-tuning, we evaluate algorithms on a broad range of problems, from academic benchmark problems to real-world applications. Each algorithm runs on all benchmarks without any change or task-specific tuning.

Large scale: Since practical problems can reach very large dimensions, we consider it desirable to include benchmark suites that comprise such problems. A "+" in Table K.1 indicates that the collection provides benchmark problems in dimension $\geq 1\,000$.

Translations: The search point zero frequently plays a special role in optimization. For example, complexity penalization often "pushes" towards zero. Also, large values in a neural network lead to saturation Glorot and Bengio [2010]: then, we get a plateau and cannot learn from the samples. In artificial experiments, several classical test functions have their optimum at $(0, \ldots, 0)$. To avoid misleading conclusions, it is now a standard procedure, advocated in particular in Hansen et al. [2012], to randomly translate the objective functions. Concretely, we consider that there is translation when optima are randomly translated by a $\mathcal{N}(0, \sigma^2)$ shift. This property is mainly interesting for artificially created benchmarks, but is unfortunately not always applied.

Symmetrizations / Rotations: Some optimization methods may perform well on separable objective functions but degrade significantly when optimizing non-separable functions. If the dimension of a separable objective function is d, these methods can reduce the objective function into d one-dimensional optimization processes Salomon [1996]. Therefore, Hansen et al. [2012, 2011] proposed that objective functions should be rotated to generate more difficult non-separable objective functions. In Bousquet et al. [2017], the importance of dummy variables, which are not invariant under rotation, was pointed out. Several references in the genetic algorithms literature, including Holland [1973], argue that rotations may not always be the right approach, in particular when the order of the variables carries a meaning. Nevergrad uses rotations, but separates the rotated and non-rotated cases in its evaluation, allowing users to focus on the setting of their choice. Assuming an optimum at 0 up to a translation step, we consider *rotation* as the replacement of the function $x \mapsto f(x)$ by $x \mapsto f(M(x))$ for a randomly selected rotation matrix M. We speak of *symmetrization* when $x \mapsto f(x)$ is replaced by $x \mapsto f(S(x))$, where S is a randomly chosen diagonal matrix whose entries are either 1 or -1.

One-line reproducibility: Where reproducibility requires significant coding, it is unlikely to be of great use outside of a very small set of specialists. One-line reproducibility is given when the effort to reproduce an entire experiment does not require more than the execution of a single line. This is possible in Nevergrad, as an example <*python -m nevergrad.benchmark yabbob –plot*> will reproduce YABBOB results on 30 cores.

Periodically automated dashboard: Some platforms do not collect the algorithms, which severely limits their reproducibility, as their implementations may not be available for public comparison. An automated and periodically rerun dashboard mitigates this risk. It is also convenient because new problems can be added "on the go" without causing problems, as all algorithms will be executed on all these new problem instances.

Complex or real-world: Benchmarks that contain real-world optimization problems, or at least complex simulators are desirable to evaluate our methods in realistic environments. MuJoCo is an example of a complex simulator.

Open sourced / no license: Another important aspect of benchmarking environments is whether or not algorithms, problems, and data are available under an open source agreement. BBOB does not collect algorithms, MuJoCo requires a license, BBComp is no longer maintained. As part of our work we integrated MuJoCo into Nevergrad, making it available to a broad public, since users can upload their algorithms in Nevergrad and they will be run on all benchmarks, including MuJoCo.

Ask/tell/recommend framework: Formalizing the concept of numerical optimization is typically made through the formalism of oracles or parallel oracles Rogers Jr [1987]. A recent trend is the adoption of the ask-and-tell format developed in Breitkopf and Coelho [2013]. The bandit literature pointed out that we should distinguish *ask*, *tell*, and *recommend*: the way we choose a point for gathering more information ("ask") is not necessarily close to the way we choose an approximation of the optimum ("recommend"), see Bubeck et al. [2011], Coulom [2012], Decock and Teytaud [2013] for detailed discussions. The difference is particularly important in noisy optimization, where an algorithm that just happens to do one lucky evaluation should not be able to get credit unless it would actively recommend that solution. A closely related issue is that a run with budget T is not necessarily close to the truncation of a run in budget $10T$.

Human excluded / client-server: Whether or not the problem instances are truly black-box. In the proper black-box setting, algorithms can only suggest points and observe function values, but neither the algorithm nor its designer have access to any other information about the problem apart from the number of variables, their type, ranges, and order. It is impossible to repeat experiments for tuning hyperparameters without "paying" the budget of the tuning. Nevergrad does not reproduce the extreme black-box nature of BBComp Škvorc et al. [2019], where the objective function is evaluated on a server and the algorithms really only perform function evaluations over the internet without having access to any other source of information about the problem at hand. Still, by integrating a wide range of benchmarks in a single open-source framework, which, in addition, is periodically re-run, we nevertheless conclude that Nevergrad provides the right environment for the development and the evaluation of ABBO.

K.2.2 Benchmarking Suites (Now) Available in Nevergrad

As a result of our work, Nevergrad now includes PBT (a small scale version of Population-Based Training Jaderberg et al. [2017]), Pyomo Hart et al. [2017], Photonics (problems related to optical properties and nanometric materials), YABBOB and variants, LSGO Li et al. [2013], MLDA Gallagher and Saleem [2018], PowerSystems, FastGames, 007, Rocket, SimpleTSP, Realworld Rapin and Teytaud [2018], Liu et al. [2020], MuJoCo Todorov et al. [2012], and others, including a (currently small) benchmark of hyperparameters of Scikit-Learn Pedregosa et al. [2011], and Keras-tuning. In this list, underlined means that the benchmark is either new (i.e., created by us), or, in the case of PowerSystems and SimpleTSP, significantly modified compared to previous works, or, in the case of Pyomo, LSGO, and MuJoCo, included for the first time inside Nevergrad. For MuJoCo, we believe that interfacing with Nevergrad is particularly useful, to ensure fair comparisons, which rely very much on the precise setup of MuJoCo. Some more details about the suites will be given in Sec. K.4 when we discuss results for selected benchmark collections.

K.3 The ABBO Algorithm Selection Wizard

Base Solvers: Black-box optimization problems are often tackled using evolutionary computation. Evolution strategies Beyer and Schwefel [2002b], Beyer [2001], Rechenberg [1989] have been particularly dominant in the continuous case, in experimental comparisons based on the

Algorithm 15: High-level overview of ABBO. Selection rules are followed in this order, first match applied. d = dimension, budget b = number of evaluations. Details of ABBO and the configuration of its base solvers are available in the Nevergrad platform Rapin and Teytaud [2018], where ABBO is listed as NGOpt8.

Case	Choice
A: Discrete decision variables only, noise-free case	
Noisy optimization with categorical variables	Genetic algorithm mixed with bandits Heidrich-Meisner and Igel [2009], Liu et al. [2020].
alphabets of size < 5, sequential evaluations	$(1+1)$-Evolutionary Alg. with linearly decreasing stepsize
alphabets of size < 5, parallel case	Adaptive $(1+1)$-Evolutionary Alg. Doerr et al. [2021].
Other discrete cases with finite alphabets	Convert to the continuous case using SoftMax as in Liu et al. [2020] and apply CMandAS2 Rapin et al. [2019]
Presence of infinite discrete domains	FastGA Doerr et al. [2017]
B: Numerical decision variables only, evaluations are subject to noise	
$d > 100$	progressive optimization as in Berthier [2016].
$d \leq 30$	TBPSA Hellwig and Beyer [2016]
$b > 100$	sequential quadratic programming
Other cases	TBPSA Hellwig and Beyer [2016]
C: Numerical decision variables only, high degree of parallelism, noise-free	
Parallelism $> b/2$ or $b < d$	MetaTuneRecentering Meunier et al. [2020c]
Parallelism $> b/5$, $d < 5$, and $b < 100$	DiagonalCMA-ES Ros and Hansen [2008]
Parallelism $> b/5$, $d < 5$, and $b < 500$	Chaining of DiagonalCMA-ES (100 asks), then CMA-ES+meta-model Auger et al. [2005b]
Parallelism $> b/5$, other cases	NaiveTBPSA as in Cauwet and Teytaud [2020]
D: Numerical decision variables only, sequential evaluations, noise-free	
$b > 6\,000$ and $d > 7$	Chaining of CMA-ES and Powell, half budget each.
$b < 30d$ and $d > 30$	$(1+1)$-Evol. Strategy w/ 1/5-th rule Rechenberg [1989]
$d < 5$ and $b < 30d$	CMA-ES + meta-model Auger et al. [2005b]
$b < 30d$	Cobyla Powell [1994]
E: other cases. Noisy discrete cases: progressive methods. Other continuous noise-free cases than C and E apply DE or CMA depending on the dimension (see code and Liu et al. [2020]).	

Black-Box Optimization Benchmark BBOB Hansen et al. [2012] or variants thereof. Parallelization advantages Salimans et al. [2016] are particularly appreciated in the machine learning context. Differential Evolution Storn and Price [1997] is a key component of most winning algorithms in competitions based on variants of Large Scale Global Optimization (LSGO Li et al. [2013]). LSGO is more based on correctly identifying a partial decomposition and scaling to $\geq 1\,000$ variables, whereas BBOB focuses (mostly, except Varelas et al. [2020]) on ≤ 40 variables. Mathematical programming techniques Powell [1964, 1994], Nelder and Mead [1965], Artelys [2015] are rarely used in the evolutionary computation world, but they have won competitions Artelys [2015] and significantly improved evolution strategies through memetic methods Radcliffe and Surry [1994]. Methods focused on MuJoCo Mania et al. [2018], Sener and Koltun [2020a] have rarely been tested on other benchmarks such as BBOB or LSGO. Reproducibility in MuJoCo is problematic as its results can depend on very small details in the implementation. Closer to machine learning, efficient global optimization Jones et al. [1998] is widely used, although it suffers from the curse of dimensionality more than other methods Snoek et al. [2012]. The LAMTCS algorithm presented in Wang et al. [2020b] applies black-box optimization on MuJoCo, i.e., for the control of various realistic robots Todorov et al. [2012].

Algorithm Selection Wizards: As mentioned, ABBO combines the various base algorithms available in Nevergrad in three different ways (see Sec. K.1). Its high-level structure is summarized in Algorithm 15 for selected optimization scenarios. We cannot replicate the full set of case distinctions here. All details are accessible via the implementation of ABBO in Nevergrad. Written in Python, this implementation is comparatively easy to navigate even for users with limited programming experience.

The most relevant predecessor of ABBO is the Shiwa algorithm presented in Liu et al. [2020]. Shiwa was also developed within Nevergrad and was shown to outperform each of the base algorithms when averaged over diverse benchmark problems. Also our ABBO entirely relies on the base algorithms as available in Nevergrad; that is, we did not modify the configuration of any method. We have, however, added a number of different algorithms for the development of ABBO, almost exclusively taken from the research literature (see below for details). We therefore acknowledge that the efficiency of ABBO heavily relies on the quality of these base components, which is based on cumulative effort of numerous research teams.

From a high-level perspective, ABBO extends Shiwa by the following features:

(1) Better use of chaining Molina et al. [2009] and more intensive use of mathematical programming techniques for the last part of the optimization run, i.e., the local convergence, thanks to Meta-Models (simple quadratic forms trained on the best points, used in the parallel case) and more time spent in Powell's method Powell [1964] (in the sequential case). This explains the improvement visible in Sec. K.4.1.

(2) Better performance in discrete optimization, achieved, in particular, by adaptive mutation rates (i.e., step size distributions).

(3) Better segmentation of the different cases of continuous optimization.

More concretely, and for the parts of the ABBO that are detailed in Algorithm 15, the main differences between ABBO and Shiwa are as follows. (A) We have added several evolutionary algorithms with variable mutation rates, 36 for the parallel cases and one for the sequential case (using a linearly decreasing mutation rate). We also introduced active algorithm selection ("bet-and-run") with CMandAS2, which—depending on the budget b—races three copies of CMA-ES or two copies of CMA-ES and a (1+1) ES for $b/10$ steps. (B) We use progressive methods (i.e., progressively adding variables in the optimization run, starting at a small set and then growing to the entire set of variables, as in Berthier [2016]) for high-dimensional cases, and we use Sequential Quadratic Programming (SQP) Artelys [2015] when the budget is sufficient for training a quadratic model. (C) We make use of the space filling design MetaTuneRecentering proposed in Meunier et al. [2020c]: we use it in the highly parallel case, but also in the sequential setting if the budget is smaller than the dimension. (D) We also use meta-models for some small dimensional cases. Overall, meta-models are helping our algorithms in the continuous setting except for sequential or high-dimensional cases.

Table K.2: Rank of ABBO, Shiwa, and CMA-ES on Selected Benchmark Suites. See main text for an explanation of symbols.

Benchmark	**Use for ABBO**	**# of configs**	**ranking**			**ABBO best competitor**
			ABBO	Shiwa	CMA-ES	
HDBO	Designing	24	2/21	$1^\dagger$	2	Shiwa
PARAMULTIMODAL	Designing	112	**1/27**	$3^\dagger$	6	DiagonalCMA-ES Ros and Hansen [2008]
Realworld	Designing	486	**1/6**	$2^\dagger$	3	Shiwa
Illcondi	Designing	12	**1/24**	$3^\dagger$	8	Cobyla
Illcondipara	Designing	12	5/28	$7^\dagger$	3	DiagonalCMA-ES
YABBOB	Designing*	630	**1/8**	2	5	Shiwa
YAPARABBOB	Designing*	630	**1/6**	4	5	MetaModel
YAHDBBOB	Designing*	378	2/19	3	18	$(1+1)$-ES
YANOISYBBOB	Designing*	630	2/11	6	10	SQP
YAHDNOISYBBOB	Designing*	630	4/24	2	13	SQP
YASMALLBBOB	Designing*	378	**1/8**	2	7	Shiwa
HdMultimodal	Validation	42	**1/14**	$2^\dagger$	4	Shiwa
Noisy	Validation	96	16/28	$19^\dagger$	NA	RecombiningOptimisticNoisyDiscrete$(1+1)$
RankNoisy	Validation	72	4/25	NA	19	ProgD13
AllDEs	Validation	60	**1/28**	$2^\dagger$	3	Shiwa
Pyomo	Evaluating	104	**1/19**	$3^\dagger$	10	Shiwa
Rocket	Evaluating	13	5/18	$4^\dagger$	3	DiagonalCMA-ES Ros and Hansen [2008]
SimpleTSP	Evaluating	52	3/15	$2^\dagger$	7	PortfolioDiscrete$(1+1)$
Seq. Fastgames	Evaluating	20	3/28	$4^\dagger$	23	OptimisticDiscrete$(1+1)$
LSGO	Evaluating	45	**1/6**	$4^\dagger$	6	MiniLHSDE
Powersystems	Evaluating	48	10/26	NA	25	$(1+1)$-ES

K.4 Experimental Results

When presenting results on a single benchmark function, we present the average objective function value for different budget values. When a collection comprises multiple benchmark problems, we present the aggregated experimental results with two distinct types of plots:

(1) Loss: normalized average (over all runs) objective value for each budget, averaged over all problems. The normalized objective value is the average objective value linearly rescaled to $[0, 1]$: then we normalize over different problems.

(2) Heatmaps, showing for each pair (x, y) of optimization algorithms the frequency at which Algorithm x outperforms Algorithm y. Algorithms are ranked by average winning frequency. For instance, in Figure K.1, ABBO wins in 63.8% of the cases against other algorithms, whereas CMA wins 50.4%.

High-Level Overview: Tab. K.2 summarizes the rank of ABBO on some of the benchmark suites. The rank is based on the winning rate in Nevergrad's dashboard Rapin and Teytaud [2020]. Each of the suites listed in Tab. K.2 comprises several problems and different settings with respect to budget, objective function, possibly dimension, and noise level. We separate benchmarks that were used for designing ABBO from those used for its validation, and those only used for testing.

The "*" symbol marks suites that were used for designing ABBO's predecessor Shiwa. Some of our modifications also improve the performance of Shiwa compared to the version published

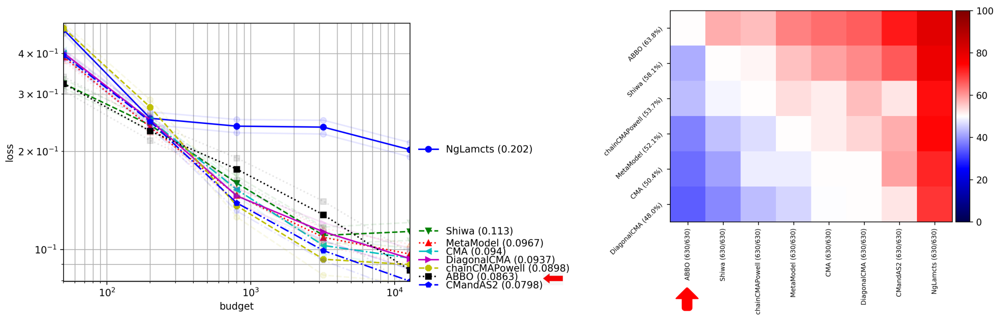


Figure K.1: Average normalized loss and heatmap for YABBOB. Additional plots for High-dimensional (HD), NoisyHD, and Large budgets are available in Fig. K.4. Other variants include parallel, differences of budgets, and combinations of those variants, with excellent results for ABBO.

in Liu et al. [2020]; for example, our chaining implies that the $(k+1)$-st code starts from the best point obtained by the k-th algorithm, which significantly improves in particular the chaining CMA-ES+Powell or CMA-ES+SQP. Experiments with "†" in the ranking of Shiwa correspond to this improved version of Shiwa.

Since the submission of this paper, several variants of bandit-based algorithms have been added for high-dimensional noisy optimization. They outperform ABBO, hence its poor rank for these cases.

K.4.1 Suites Used for Designing and Validating ABBO

YABBOB (Yet Another Black-Box Optimization Benchmark Rapin et al. [2019]), is an adaptation of BBOB Hansen et al. [2012], with extensions such as parallelism and noise management. It contains many variants, including noise, parallelism, high-dimension (prior to Varelas et al. [2020] BBOB was limited to dimension < 50). Results are available in Figures K.1 and K.4. The high-dimensional suite is inspired by Li et al. [2013], the noisy one is related to the noisy counterpart of BBOB but implements the difference between ask and recommend as discussed in Sec. K.2. The parallel one generalizes YABBOB to settings in which several evaluations can be executed in parallel. Results on PARAMULTIMODAL are presented in Fig. K.6 (left). In addition, ABBO was run on ILLCONDI & ILLCONDIPARA (ill conditioned functions), HDMULTIMODAL (a multimodal case focusing on high-dimension), NOISY & RANKNOISY (two noisy continuous testbeds), YAWIDEBBOB (a broad range of functions including discrete cases and cases with constraints).

AllDEs and Hdbo are benchmark collections specifically designed to compare DE variants (AllDEs) and high-dimensional Bayesian Optimization (Hdbo), respectively Rapin and Teytaud [2018]. These benchmark functions are similar to the ones used in YABBOB. Many variants of DE (resp. BO) are considered. Results are presented in Fig. K.5. They show that the performance of ABBO, relatively to DE or BO, is consistent over a wide range of parametrizations of DE or BO, at least in their most classical variants, which are all available in Nevergrad for empirical comparisons.

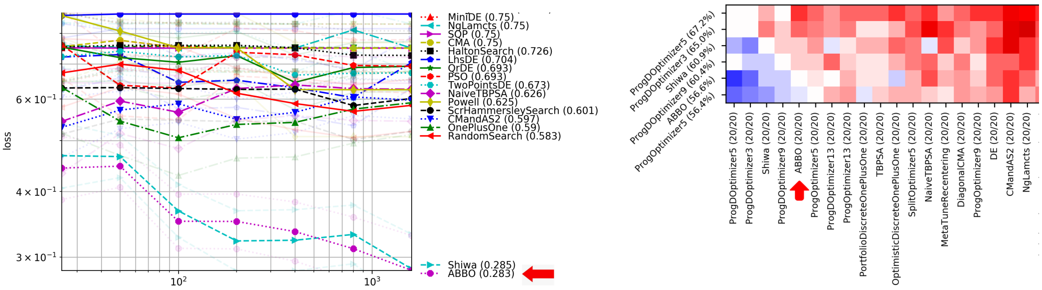


Figure K.2: Additional problems: Pyomo (left figure, covering Knapsack, P-median and others) and SequentialFastgames (on the right, presented as heatmaps due to the high noise. Subsumes Guess-Who, War, Batawaf, Flip). Rockets, SimpleTSP, PowerSystems, and LSGO plots are available in Figs. K.7, and K.8. Pyomo and SimpleTSP include discrete variables. Pyomo includes constraints. Rocket, PowerSystems, SequentialFastGames are based on open source simulators.

Realworld: A test of ABBO is performed on the Realworld optimization benchmark suite proposed in Rapin and Teytaud [2018]. This suite includes testbeds from MLDA Gallagher and Saleem [2018] and from Liu et al. [2020]. Results for this suite, presented in Fig. K.6, confirm that ABBO performs well also on benchmarks that were not explicitly used for its design. However, this benchmark was used for designing Shiwa, which was the starting point for the design of ABBO. A rigorous cross-validation, on benchmarks totally independent from the design of Shiwa, is provided in the next sections.

K.4.2 New Benchmark Suites Used Only for Evaluating ABBO

Pyomo is a modeling language in Python for optimization problems Hart et al. [2017]. It has been adopted for formulating large models for complex and real-world systems, including energy systems and network resource systems. We implemented an interface to Pyomo for Nevergrad. Experimental results are summarized in Fig. K.2. They show that ABBO also performs decently in discrete settings and in constrained cases.

Additional new artificial and real-world functions: LSGO Li et al. [2013] combines various functions into an aggregated testbed including composite highly multimodal functions. Correctly decomposing the problem is essential. Various implementations of LSGO exist; in particular, the Octave and C++ versions do not match exactly for F3/F6/F10. We match the C++ version, which is the one used in Li et al. [2013]. For F7, there is a difference between the code and the paper and we match the code rather than the paper. Following Li et al. [2013], our implementation comprises functions with subcomponents (i.e., groups of decision variables) having non-uniform sizes and non-uniform, even conflicting, contributions to the objective function. We also present experimental results on **SequentialFastgames** from the Nevergrad benchmarks, and three newly introduced benchmarks, namely **Rocket**, **Simple TSP** (a set of traveling salesman problems, where a vector $x \in \mathbb{R}^d$ is converted into a permutation σ by letting $\sigma(i)$ be the index of the i-th largest element in x (ties broken at random)), and **power systems** (unit commitment problems Padhy [2004]). Experimental results are presented in Figs. K.2, K.7, and K.8,

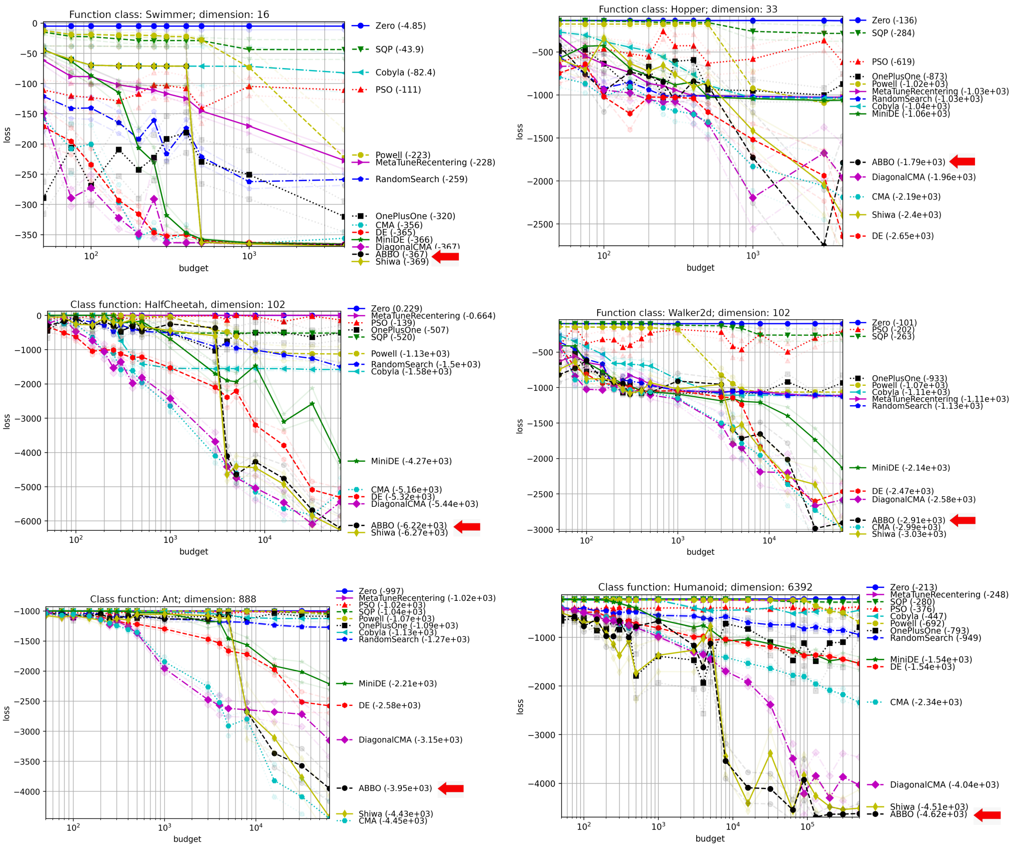


Figure K.3: Results on the MuJoCo testbeds. Dashed lines show the standard deviation. Compared to the state of the art in Wang et al. [2020b], with an algorithm adapted manually for the different tasks, we get overall better results for Humanoid, Ant, and Walker. We get worse results for Swimmer (could match if we had modified our code for the three easier tasks as done in Wang et al. [2020b]), similar for Hopper and Cheetah: we reach the target for 5 of the 6 problems (see main text). Runs of Shiwa correspond to the improvement of Shiwa due to chaining, as explained in Sec. K.3.

respectively. They show that ABBO performs well on new benchmarks, never used for its design nor for that of the low-level heuristics used inside ABBO.

MuJoCo: Some articles Sener and Koltun [2020b], Wang et al. [2020b] studied the MuJoCo testbeds Todorov et al. [2012] in the black-box setting. MuJoCo tasks correspond to control problems. Defined in Wang et al. [2020b], Mania et al. [2018], the objective is to learn a linear mapping from states to actions. It turned out that the scaling of the variables is critical Mania et al. [2018]: following the recommendation from Meunier et al. [2020c] to sample close to zero in the high dimensional setting, we chose to initialize all the variables of the problem with a variance decaying with the dimension, for all methods run in Fig. K.3. We remark that ABBO and Shiwa perform well, even when compared to gradient-based methods, while having the advantage of being applicable to settings in which gradients are not available. In comparison to gradient-based methods,

Table K.3: Results on MuJoCo for a linear policy in the black-box setting from Wang et al. [2020b] and references therein. We compare various published results to results from ABBO. Two last columns = average reward for the maximum budget tested in Wang et al. [2020b], namely 1k, 4k, 4k, 40k, 30k, 40k, respectively. "ioa" = iterations on average for reaching the target. "iter" = iterations for target reached for median run. "*" refers to problems for which the target was not reached by Wang et al. [2020b]: then BR means "best result in 10 runs". ABBO reaches the target for Humanoid and Ant whereas previous (black-box) papers did not; we get nearly the same ioa for Hopper and HalfCheetah (Nevergrad computed the expected value instead of computing the ioa, so we cannot compare exactly; see Fig. K.3 for curves). ABBO is slower than LA-MCTS on Swimmer. Note that we keep the same method for all benchmarks whereas LA-MCTS modified the algorithm for 3 rows. On HDMULTIMODAL, ABBO performs better than LA-MCTS, as detailed in the text, and as confirmed in Wang et al. [2020b], which acknowledges the poor results of LA-MCTS for high-dimensional Ackley and Rosenbrock.

Task	**Target**	**LA-MCTS results**	**ABBO result**	**LA-MCTS avg reward**	**ABBO avg reward**
Swimmer-v2	325	132 ioa	around 450 iter	358	**365**
Hopper-v2	3120	2897 ioa	around 3 000 iter	**3292**	1787
HalfCheetah-v2	3430	3877 ioa	around 4 000 iter	3227	**4730**
Walker2d-v2*	4390	BR: 3314 (not reached)	BR: **4398**, budget $<$ 64 000	2769	**2949**
Ant-v2*	3580	BR: 2791 (not reached)	BR: **5325**, budget $<$ 32 000	2511	**3532**
Humanoid-v2*	6 000	BR: 3384 (not reached)	BR (budget 5 00000): **4870**	2511	**4620**

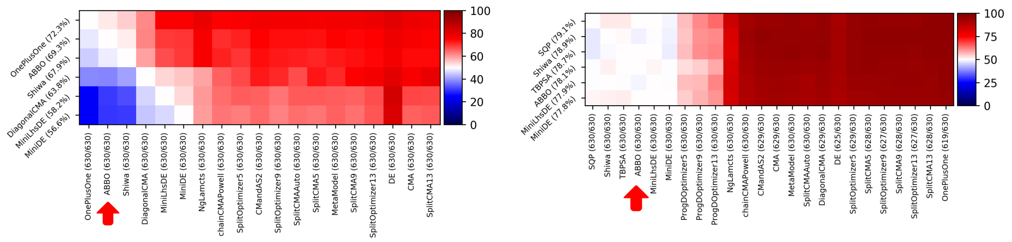


Figure K.4: YAHDBBOB (dimension $\geq$ 50) and YANOISYHDBBOB (noisy + dimension $\geq$ 50) heatmaps.

black-box methods do not require computation of the gradient, and hence, save computational time.

We use the same experimental setup as Wang et al. [2020b] (linear policy, offline whitening of states). We get better results than LA-MCTS, in a setting without using any expensive surrogate model (Tab. K.3). Our runs with CMA-ES and Shiwa are better than those in Wang et al. [2020b]. We acknowledge that LMRS Sener and Koltun [2020b] outperforms our method on all MuJoCo tasks, using a deep network as a surrogate model: however, we point out that a part of their code is not open sourced, making the experiments not reproducible. In addition, when rerunning their repository without the closed source part, it solved Half-Cheetah within budget 56k, which is larger than ours. For Humanoid, the target was reached at 768k, which is again larger than our budget. The results from ABBO are comparable to, and are usually better than (for the 3 hardest problems) the results from LA-MCTS, while ABBO is entirely reproducible. In addition, it runs the same method for all benchmarks and it is not optimized for each task specifically as in Sener

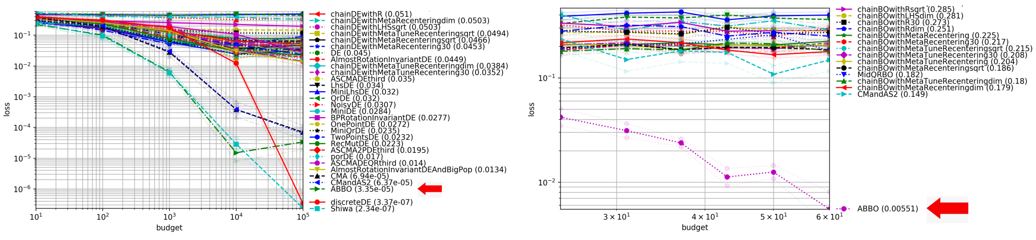


Figure K.5: ABBO vs specific families of optimization algorithms (DE on the left in dimension 5, 20 and 100; and BO in dimension 20 on the right) on Cigar, Hm, Ellipsoid, Sphere functions. Not all run algorithms are mentioned, for short. Bayesian optimization (Nevergrad uses Nogueira [2014–]), often exploring boundaries first, is outperformed in high dimension Wang et al. [2020b].

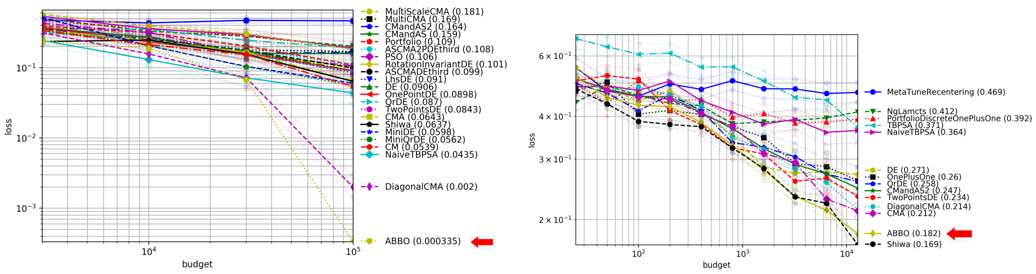


Figure K.6: Left: experiments for the parallel multimodal setting PARAMULTIMODAL. Budget up to 100 000, parallelism 1 000, Ackley+Rosenbrock+DeceptiveMultimodal+Griewank+Lunacek+Hm. Right: Realworld benchmark from Nevergrad: games, Sammon mappings, clustering, small traveling salesman instance, small power systems.

and Koltun [2020b], Wang et al. [2020b]. In contrast to ABBO, LA-MCTS Wang et al. [2020b] uses different underlying regression methods and sampling methods depending on the MuJoCo task, and it is not run on other benchmarks except for some of the HDMULTIMODAL ones. On the latter, ABBO performances are significantly better for Ackley and Rosenbrock in dimension 100 (average results around 100 and 10^{-8} after 10k iterations for Rosenbrock and Ackley respectively for ABBO, vs. 500 and 0.5 in Wang et al. [2020b]). From the curves in Wang et al. [2020b] and those presented here in this paper, we expect LA-MCTS to perform well with an adapted choice of parametrization and with a low budget, for tasks related to MuJoCo, whereas ABBO is adapted for wide ranges of tasks and budgets.

As mentioned at the end of the introduction, videos illustrating the performance of the learnt policies are available at Meunier et al. [2021d].

K.5 Conclusions

We have introduced in this paper ABBO, an improved algorithm selection wizard that significantly improves upon its predecessor Shiwa Liu et al. [2020]. For the development and the evaluation of ABBO we have considerably extended the Nevergrad platform by adding several real-

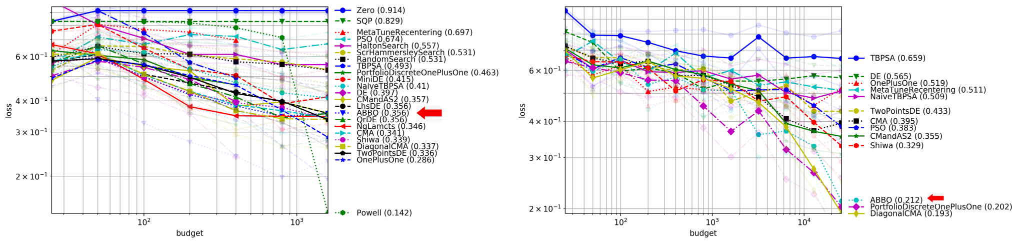


Figure K.7: Additional problems (1): on left, Rocket (26 continuous variables, budget up to 1600, sequential or parallelism 30) and on right, SimpleTSP (10 to 1 000 decision variables).

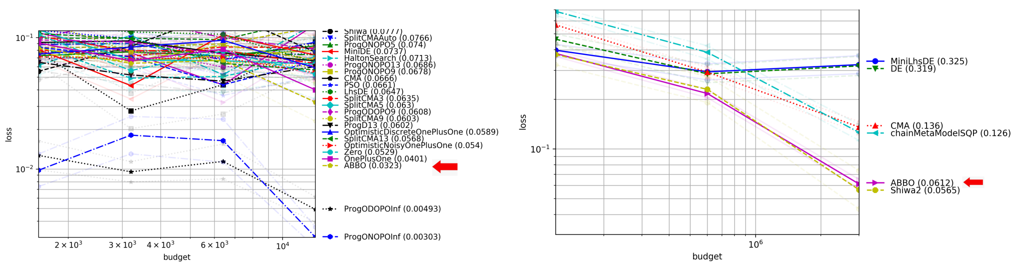


Figure K.8: Additional problems (2): on left, PowerSystems (1806 to 9646 neural decision variables) and on right, LSGO (mix of partially separable, overlapping, shifted cases as in Li et al. [2013]).

world and academic benchmark suites. All our work is available in the master branch of Nevergrad, where it is available for reproducible, open-source research. ABBO is listed as NGOpt8 in Nevergrad.

Despite its simplicity, ABBO shows very promising performance across the whole benchmark suite, often outperforming the previous state-of-the-art, problem-specific solvers. Highlights of our performance comparison include: (a) by solving 5 of the 6 MuJoCo cases without any task-specific hyperparameter tuning, ABBO outperforms LA-MCTS, which was specialized for each single task, (b) ABBO outperforms Shiwa on YABBOB and its variants, which is the benchmark suite that was used to design Shiwa in the first place, (c) ABBO is also among the best methods on LSGO and almost all other benchmarks.

Future work: Nevergrad implements most of the desirable features outlined in Sec. K.2, with one notable exception, the true black-box setting, which other benchmark environments have implemented through a client-server interaction Škvorc et al. [2019]. A possible combination between our platform and such a challenge, using the dashboard to publish the results, could be useful, to offer a meaningful way for cross-validation. Further improving ABBO is on the roadmap. In particular, we are experimenting with the automation of the still hand-crafted selection rules. Note, though, that it is important to us to maintain a high level of interpretability, which we consider key for a wide acceptance of the wizard. Another avenue for future work is a proper configuration of the low-level heuristics subsumed by ABBO. At present, some of them are merely textbook implementations, and significant room for improvement can therefore be expected. Newer variants of CMA-ES Loshchilov [2014], Akimoto and Hansen [2016], Loshchilov et al. [2018], of LMRS Sener and Koltun [2020b], recent Bayesian optimization libraries (e.g. Eriksson et al. [2019]), as well as per-instance algorithm configuration such as Belkhir et al. [2017] are not unlikely to result in important improvements for various benchmarks. We also plan on extending the benchmark collection available through Nevergrad further, both via interfacing existing benchmark collections/problems and by designing new benchmark problems ourselves.

Bibliography

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Marc Abeille, Alessandro Lazaric, et al. Linear thompson sampling revisited. *Electronic Journal of Statistics*, 11(2):5165–5197, 2017.

Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

Youhei Akimoto and Nikolaus Hansen. Projection-based restricted covariance matrix adaptation for high dimension. In *Proc. of Genetic and Evolutionary Computation (GECCO'16)*, pages 197–204, 2016.

Youhei Akimoto, Anne Auger, Tobias Glasmachers, and Daiki Morinaga. Global linear convergence of evolution strategies on more than smooth strongly convex functions, 2020.

Mohamed Akrout. On the adversarial robustness of neural networks without weight transport. *ArXiv*, abs/1908.03560, 2019.

Abdullah Al-Dujaili and Una-May O'Reilly. There are no bit parts for sign bits in black-box attacks, 2019.

Rafael Pinot Alexandre Araujo, Laurent Meunier and Benjamin Negrevergne. Advocating for multiple defense strategies against adversarial examples. *Workshop on Machine Learning for CyberSecurity (MLCS@ECML-PKDD)*, 2020.

David Alvarez-Melis, Stefanie Jegelka, and Tommi S Jaakkola. Towards optimal transport with global invariances. *arXiv preprint 1806.09277*, 2018.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, 2017.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. *arXiv preprint arXiv:1912.00049*, 2019.

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, 2019.

Alexandre Araujo, Rafael Pinot, Benjamin Negrevergne, Laurent Meunier, Yann Chevaleyre, Florian Yger, and Jamal Atif. Robust neural networks using randomized adversarial training. *arXiv preprint arXiv:1903.10219*, 2019.

Alexandre Araujo, Laurent Meunier, Rafael Pinot, and Benjamin Negrevergne. Advocating for multiple defense strategies against adversarial examples. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 165–177. Springer, 2020.

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.

Dirk V Arnold and H-G Beyer. A general noise model and its effects on evolution strategy performance. *IEEE Transactions on Evolutionary Computation*, 10(4):380–391, 2006.

Dirk V. Arnold, Hans-Georg Beyer, and Alexander Melkozerov. On the behaviour of weighted multi-recombination evolution strategies optimising noisy cigar functions. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, page 483–490, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605583259. doi: 10.1145/1569901.1569969. URL https://doi.org/10.1145/1569901.1569969.

SME Artelys, 2015. URL https://www.artelys.com/news/159/16/KNITRO-wins-the-GECCO-2015-Black-Box-Optimization-Competition.

Sandra Astete-Morales, Marie-Liesse Cauwet, and Olivier Teytaud. Evolution strategies with additive noise: A convergence rate lower bound. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, FOGA '15, page 76–84, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334341. doi: 10.1145/2725494.2725500. URL https://doi.org/10.1145/2725494.2725500.

Emanouil I Atanassov. On the discrepancy of the Halton sequences. *Math. Balkanica (NS)*, 18 (1-2):15–32, 2004.

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, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018b. URL https://arxiv.org/abs/1802.00420.

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 2018c. PMLR. URL http://proceedings.mlr.press/v80/athalye18b.html.

Charles Audet, Amina Ihaddadene, Sébastien Le Digabel, and Christophe Tribes. Robust optimization of noisy blackbox problems using the mesh adaptive direct search algorithm. *Optimization Letters*, 12(4):675–689, 2018. doi: 10.1007/s11590-017-1226-6. URL https://scholar.google.com/scholar?cluster=17486798405551999711.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Anne Auger, Marc Schoenauer, and Olivier Teytaud. Local and global order 3/2 convergence of a surrogate evolutionnary algorithm. In Beyer and O'Reilly [2005], page 8. ISBN 1-59593-010-8.

Anne Auger, Marc Schoenauer, and Olivier Teytaud. Local and global order 3/2 convergence of a surrogate evolutionary algorithm. In Beyer and O'Reilly [2005], pages 857–864. ISBN 1-59593-010-8. doi: 10.1145/1068009.1068154. URL https://doi.org/10.1145/1068009.1068154.

Anne Auger, Dimo Brockhoff, and Nikolaus Hansen. Mirrored sampling in evolution strategies with weighted recombination. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, page 861–868, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450305570. doi: 10.1145/2001576.2001694. URL https://doi.org/10.1145/2001576.2001694.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. Adversarial learning guarantees for linear hypotheses and neural networks. *International Conference on Machine Learning*, 2020.

Pranjal Awasthi, Natalie Frank, Anqi Mao, Mehryar Mohri, and Yutao Zhong. Calibration and consistency of adversarial surrogate losses. *arXiv preprint arXiv:2104.09658*, 2021a.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. On the existence of the adversarial bayes classifier. *Advances in Neural Information Processing Systems*, 34, 2021b.

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

Mona Azadkia and Sourav Chatterjee. A simple measure of conditional dependence. *The Annals of Statistics*, 49(6):3070–3102, 2021.

Han Bao, Clay Scott, and Masashi Sugiyama. Calibrated surrogate losses for adversarially robust classification. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 408–451. PMLR, 09–12 Jul 2020. URL http://proceedings.mlr.press/v125/bao20a.html.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, 2019.

Nacim Belkhir, Johann Dréo, Pierre Savéant, and Marc Schoenauer. Per instance algorithm configuration of cma-es with limited budget. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'17)*, page 681–688. ACM, 2017.

Mikhail Belkin, Daniel J Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In *Advances in neural information processing systems*, pages 2300–2311, 2018.

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.

Alexis Bellot and Mihaela van der Schaar. Conditional independence testing using generative adversarial networks. In *Advances in Neural Information Processing Systems 32*, pages 2199–2208, 2019.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

Wicher Pieter Bergsma. *Testing conditional independence for continuous random variables*. Citeseer, 2004.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(Feb):281–305, February 2012.

Thomas B Berrett, Yi Wang, Rina Foygel Barber, and Richard J Samworth. The conditional permutation test for independence while controlling for confounders. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(1):175–197, 2020.

Vincent Berthier. Progressive differential evolution on clustering real world problems. In *Revised Selected Papers of the 12th International Conference on Artificial Evolution - Volume 9554*, pages 71–82, New York, NY, USA, 2016. Springer-Verlag New York, Inc. ISBN 978-3-319-31470-9.

Dimitir P Bertsekas and Steven Shreve. *Stochastic optimal control: the discrete-time case*. 2004.

Dimitris Bertsimas and John N Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

Louis Béthune, Alberto González-Sanz, Franck Mamalet, and Mathieu Serrurier. The many faces of 1-lipschitz neural networks. *arXiv preprint arXiv:2104.05097*, 2021.

H.-G. Beyer. *The Theory of Evolution Strategies*. Natural Computing Series. Springer, Heideberg, Berlin, 2001. URL http://books.google.com/books?id=8tbInLufkTMC.

Hans-Georg Beyer. Toward a theory of evolution strategies: On the benefits of sex—the $(\mu/\mu, \lambda)$ theory. *Evol. Comput.*, 3(1):81–111, March 1995. ISSN 1063-6560. doi: 10.1162/evco.1995.3.1.81. URL https://doi.org/10.1162/evco.1995.3.1.81.

Hans-Georg Beyer and Una-May O'Reilly, editors. *Genetic and Evolutionary Computation Conference, GECCO 2005, Proceedings, Washington DC, USA, June 25-29, 2005*, 2005. ACM. ISBN 1-59593-010-8.

Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies –a comprehensive introduction. *Natural Computing: An International Journal*, 1(1):3–52, May 2002a. URL https://doi.org/10.1023/A:1015059928466.

Hans-Georg Beyer and Bernhard Sendhoff. Covariance matrix adaptation revisited – the cmsa evolution strategy –. In Günter Rudolph, Thomas Jansen, Nicola Beume, Simon Lucas, and Carlo Poloni, editors, *Parallel Problem Solving from Nature – PPSN X*, pages 123–132, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-87700-4.

H.G. Beyer and HP. Schwefel. Evolution strategies - a comprehensive introduction. *Natural Computing*, 1(1):3–52, March 2002b. doi: 10.1023/A:1015059928466.

Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. *Advances in Neural Information Processing Systems*, 32:7496–7508, 2019.

Battista Biggio, Giorgio Fumera, and Fabio Roli. Evade hard multiple classifier systems. In *Applications of Supervised and Unsupervised Ensemble Methods*, pages 15–38. Springer, 2009.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

Åke Björck et al. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 1971.

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

Avishek Joey Bose, Gauthier Gidel, Hugo Berard, Andre Cianflone, Pascal Vincent, Simon Lacoste-Julien, and William L. Hamilton. Adversarial example games, 2021.

Jakob Bossek, Pascal Kerschke, Aneta Neumann, Frank Neumann, and Carola Doerr. One-shot decision-making with and without surrogates. *CoRR*, abs/1912.08956, 2019. URL http://arxiv.org/abs/1912.08956.

Jakob Bossek, Carola Doerr, and Pascal Kerschke. Initial design strategies and their effects on sequential model-based optimization. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO'20)*. ACM, 2020. To appear. Available at https://arxiv.org/abs/2003.13826.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Olivier Bousquet, Sylvain Gelly, Kurach Karol, Olivier Teytaud, and Damien Vincent. Critical hyper-parameters: No random, no cry. Preprint https://arxiv.org/pdf/1706.03200.pdf, 2017.

Stephen Boyd. Subgradient methods. 2003.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

Piotr Breitkopf and Rajan Filomeno Coelho. *Multidisciplinary design optimization in computational mechanics*. John Wiley & Sons, 2013.

Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 547–555, New York, NY, USA, 2011. Association for Computing Machinery. doi: 10.1145/2020408.2020495.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 42–1, 2012.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pages 23–37. Springer, 2009.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theor. Comput. Sci.*, 412(19):1832–1852, 2011.

Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pages 831–840. PMLR, 2019.

Leon Bungert, Nicolás García Trillos, and Ryan Murray. The geometry of adversarial training in binary classification. *arXiv preprint arXiv:2111.13613*, 2021.

Emmanuel Candès, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold:'model-x'knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

A. Cappelli, Ruben Ohana, Julien Launay, Laurent Meunier, Iacopo Poli, and F. Krzakala. Adversarial robustness by design through analog computing and synthetic gradients. *ArXiv*, abs/2101.02115, 2021a.

Alessandro Cappelli, Julien Launay, Laurent Meunier, Ruben Ohana, and Iacopo Poli. Ropust: Improving robustness through fine-tuning with photonic processors and synthetic gradients. *arXiv preprint arXiv:2108.04217*, 2021b.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

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.

Y. Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019a.

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

Marie-Liesse Cauwet and Olivier Teytaud. Population control meets doob's martingale theorems: the noise-free multimodal case. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 321–322, 2020.

Marie-Liesse Cauwet, Jialin Liu, Baptiste Rozière, and Olivier Teytaud. Algorithm portfolios for noisy optimization. *Annals of Mathematics and Artificial Intelligence*, 76(1):143–172, 2016.

Marie-Liesse Cauwet, Camille Couprie, Julien Dehos, Pauline Luc, Jérémy Rapin, Morgane Riviere, Fabien Teytaud, and Olivier Teytaud. Fully parallel hyperparameter search: Reshaped space-filling. *arXiv preprint arXiv:1910.08406. To appear in Proc. of ICML 2020*, 2019.

Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Fast conditional independence test for vector variables with large sample sizes. *arXiv preprint arXiv:1804.02747*, 2018.

Tony F Chan and Selim Esedoglu. Aspects of total variation regularized l 1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

François Chapeau-Blondeau and David Rousseau. Noise-enhanced performance for an optimal bayesian estimator. *IEEE Transactions on Signal Processing*, 52(5):1327–1334, 2004.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence*, 2018a.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018b.

Lénaïc Chizat. Convergence rates of gradient methods for convex optimization in the space of measures. *arXiv preprint arXiv:2105.08368*, 2021a.

Lenaic Chizat. Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming*, pages 1–46, 2021b.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274 (11):3090–3123, 2018.

Alexandre Chotard, Anne Auger, and Nikolaus Hansen. Cumulative step-size adaptation on linear functions. In Carlos A. Coello Coello, Vincenzo Cutello, Kalyanmoy Deb, Stephanie Forrest, Giuseppe Nicosia, and Mario Pavone, editors, *Parallel Problem Solving from Nature - PPSN XII*, pages 72–81, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-32937-1.

Konstantina Christakopoulou and Arindam Banerjee. Adversarial attacks on an oblivious recommender. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 322–330, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362436. doi: 10.1145/3298689.3347031. URL https://doi.org/10.1145/3298689.3347031.

Kacper P Chwialkowski, Aaditya Ramdas, Dino Sejdinovic, and Arthur Gretton. Fast two-sample testing with analytic representations of probability measures. *Advances in Neural Information Processing Systems*, 28:1981–1989, 2015.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 854–863, 2017.

John Cloutier, Kathryn L Nyman, and Francis Edward Su. Two-player envy-free multi-cake division. *Mathematical Social Sciences*, 59(1):26–37, 2010.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, abs/1902.02918, 2019. URL http://arxiv.org/abs/1902.02918.

Patrick L Combettes and Jean-Christophe Pesquet. Lipschitz certificates for layered network structures driven by averaged activation operators. *SIAM Journal on Mathematics of Data Science*, 2020.

Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 1–6, 1987.

Rémi Coulom. CLOP: confident local optimization for noisy black-box parameter tuning. In H. Jaap van den Herik and Aske Plaat, editors, *Advances in Computer Games - 13th International Conference, ACG 2011, Tilburg, The Netherlands, November 20-22, 2011, Revised Selected Papers*, volume 7168 of *Lecture Notes in Computer Science*, pages 146–157. Springer, 2011. doi: 10.1007/978-3-642-31866-5_13. URL https://doi.org/10.1007/978-3-642-31866-5_13.

Rémi Coulom. Clop: Confident local optimization for noisy black-box parameter tuning. In H. Jaap van den Herik and Aske Plaat, editors, *Advances in Computer Games*, pages 146–157, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-31866-5.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9): 1853–1865, 2016.

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

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.

Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of adversaries. *Advances in Neural Information Processing Systems*, 31, 2018.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004.

JJ Daudin. Partial association measures and an application to qualitative regression. *Biometrika*, 67(3):581–590, 1980.

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296, 2010.

Jérémie Decock and Olivier Teytaud. Noisy optimization complexity under locality assumption. In *Proceedings of the Twelfth Workshop on Foundations of Genetic Algorithms XII*, FOGA XII '13, pages 183–190, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1990-4. doi: 10.1145/2460239.2460256. URL http://doi.acm.org/10.1145/2460239.2460256.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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.

Josef Dick and Friedrich Pillichshammer. *Digital Nets and Sequences*. Cambridge University Press, 2010.

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

Gavin Weiguang Ding, Yash Sharma, Kry Yik-Chau Lui, and Ruitong Huang. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. *ArXiv*, abs/1812.02637, 2020.

Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R Nevins, Guang Yao, and Mike West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1): 196–212, 2004.

Benjamin Doerr, Huu Phuoc Le, Régis Makhmara, and Ta Duy Nguyen. Fast genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 777–784, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4920-8.

Benjamin Doerr, Carola Doerr, and Johannes Lengler. Self-adjusting mutation rates with provably optimal success rules. *Algorithmica*, 83(10):3108–3147, 2021.

Elvis Dohmatob. Generalized no free lunch theorem for adversarial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1646–1654, Long Beach, California, USA, 2019. PMLR. URL http://proceedings.mlr.press/v97/dohmatob19a.html.

Carles Domingo-Enrich, Samy Jelassi, Arthur Mensch, Grant Rotskoff, and Joan Bruna. A mean-field analysis of two-player zero-sum games. *Advances in Neural Information Processing Systems*, 33:20215–20226, 2020.

Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE, 2018.

Gary Doran, Krikamol Muandet, Kun Zhang, and Bernhard Schölkopf. A permutation-based kernel conditional independence test. In *UAI*, pages 132–141. Citeseer, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Lester E Dubins and Edwin H Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1P1):1–17, 1961.

John Duchi, Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

Richard M. Dudley. The speed of mean Glivenko-Cantelli convergence. *Annals of Mathematical Statistics*, 40(1):40–50, 1969.

Richard Mansfield Dudley et al. Weak convergence of probabilities on nonseparable metric spaces and empirical measures on euclidean spaces. *Illinois Journal of Mathematics*, 10(1):109–126, 1966.

Paul Dupuis and Richard S Ellis. *A weak convergence approach to the theory of large deviations*, volume 902. John Wiley & Sons, 2011.

Arnaud Dupuy, Alfred Galichon, and Yifei Sun. Estimating matching affinity matrix under low-rank constraints. *arXiv preprint arXiv:1612.09585*, 2016.

Arkadiy Dushatskiy, Tanja Alderliesten, and Peter A. N. Bosman. A novel surrogate-assisted evolutionary algorithm applied to partition-based ensemble learning, 2021.

Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 2017.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811. PMLR, 2019.

M. Ergezer and I. Sikder. Survey of oppositional algorithms. In *14th International Conference on Computer and Information Technology (ICCIT 2011)*, pages 623–628, 2011.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 5496–5507. Curran Associates, 2019.

H.J. Escalante and A. Morales Reyes. Evolution strategies. *CCC-INAOE tutorial*, 2013.

A. Esmailzadeh and S. Rahnamayan. Enhanced differential evolution using center-based sampling. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2641–2648, 2011.

A. Esmailzadeh and S. Rahnamayan. Center-point-based simulated annealing. In *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4, 2012.

Ky Fan. Minimax theorems. *Proceedings of the National Academy of Sciences of the United States of America*, 39(1):42, 1953.

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

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pages 1632–1640, 2016.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *AAAI*, 2015.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-Ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. *arXiv preprint arXiv:1810.08278*, 2018.

Simon Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *J. Mach. Learn. Res.*, 21:205–1, 2020.

Matteo Fischetti and Michele Monaci. Exploiting erraticism in search. *Operations Research*, 62 (1):114–122, 2014.

Hervé Fournier and Olivier Teytaud. Lower Bounds for Comparison Based Evolution Strategies using VC-dimension and Sign Patterns. *Algorithmica*, 2010.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738, 2015.

Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5(Jan): 73–99, 2004.

Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Advances in neural information processing systems*, pages 489–496, 2008.

Marcus Gallagher and Sobia Saleem. Exploratory landscape analysis of the mlda problem set. In *Proceedings of PPSN18 Workshops*, 2018.

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.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences, 2017.

Aude Genevay, Lénaic Chizat, Francis Bach, Marco Cuturi, and Gabriel Peyré. Sample complexity of sinkhorn divergences. *arXiv preprint arXiv:1810.02733*, 2018.

Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International Statistical Review / Revue Internationale de Statistique*, 70(3):419–435, 2002. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/1403865.

G. L. Gilardoni. On pinsker's and vajda's type inequalities for csiszár's f-divergences. *IEEE Transactions on Information Theory*, 56(11):5377–5386, Nov 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2068710.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

Amir Globerson et al. Nightmare at test time: Robust learning by feature deletion. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Morgane Goibert and Elvis Dohmatob. Adversarial robustness via adversarial label-smoothing. *ArXiv*, abs/1906.11567, 2019.

Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

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

Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6 (4):1–19, 2015.

I. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015a.

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*, 2015b.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020a.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy A. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *ArXiv*, abs/2010.03593, 2020b.

Yves Grandvalet, Stéphane Canu, and Stéphane Boucheron. Noise injection: Theoretical prospects. *Neural Computation*, 9(5):1093–1108, 1997.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Samuel J Greydanus, Misko Dzumba, and Jason Yosinski. Hamiltonian neural networks. 2019.

Ziwei Guan, Kaiyi Ji, Donald J Bucci Jr, Timothy Y Hu, Joseph Palombo, Michael Liston, and Yingbin Liang. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. *arXiv preprint arXiv:2002.07214*, 2020.

Benjamin Guedj. A primer on pac-bayesian learning. *arXiv preprint arXiv:1901.05353*, 2019.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. *arXiv preprint arXiv:2104.13733*, 2021.

Anupam Gupta, Tomer Koren, and Kunal Talwar. Better algorithms for stochastic bandits with adversarial corruptions. *arXiv preprint arXiv:1902.08647*, 2019a.

Lipi Gupta, Auralee Edelen, Nicole Neveu, Aashwin Mishra, Christopher Mayes, and Young-Kee Kim. Improving surrogate model accuracy for the lcls-ii injector frontend using convolutional neural networks and transfer learning, 2021.

S. Gupta, R. Gribonval, L. Daudet, and Ivan Dokmanić. Don't take it lightly: Phasing optical random projections with unknown operators. In *NeurIPS*, 2019b.

S. Gupta, R. Gribonval, L. Daudet, and Ivan Dokmani'c. Fast optical system identification by numerical interferometry. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1474–1478, 2020.

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

J.H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960. URL http://eudml.org/doc/131448.

J. M. Hammersley. Monte-carlo methods for solving multivariate problems. *Annals of the New York Academy of Sciences*, 86(3):844–874, 1960.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 11(1), 2003.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

Nikolaus Hansen, Raymond Ros, Nikolas Mauny, Marc Schoenauer, and Anne Auger. Impacts of Invariance in Search: When CMA-ES and PSO Face Ill-Conditioned and Non-Separable Problems. *Applied Soft Computing*, 11:5755–5769, 2011. doi: 10.1016/j.asoc.2011.03.001. URL https://hal.inria.fr/inria-00583669.

Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-parameter black-box optimization benchmarking: Experimental setup. *Orsay, France: Université Paris Sud, Institut National de Recherche en Informatique et en Automatique (INRIA) Futurs, Équipe TAO, Tech. Rep*, 2012.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis) DOI=http://dx.doi.org/10.1145/2827872*, 5 (4):1–19, 2015.

William E Hart, Carl D Laird, Jean-Paul Watson, David L Woodruff, Gabriel A Hackebeil, Bethany L Nicholson, John D Siirola, et al. *Pyomo-optimization modeling in python*, volume 67. Springer, 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, X. Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *ArXiv*, abs/1603.05027, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016b.

Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.

Verena Heidrich-Meisner and Christian Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 401–408, 2009.

Michael Hellwig and Hans-Georg Beyer. Evolution under strong noise: A self-adaptive evolution strategy can reach the lower performance bound - the pccmsa-es. In Julia Handl, Emma Hart, Peter R. Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter, editors, *Parallel Problem Solving from Nature – PPSN XIV*, pages 26–36, Cham, 2016. Springer International Publishing. ISBN 978-3-319-45823-6.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

John H. Holland. Genetic algorithms and the optimal allocation of trials. *SIAM J. Comput.*, 2 (2):88–105, 1973. doi: 10.1137/0202009. URL https://doi.org/10.1137/0202009.

Sara Hooker. The hardware lottery. *ArXiv*, abs/2009.06489, 2020.

Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. Finding mixed nash equilibria of generative adversarial networks. In *International Conference on Machine Learning*, pages 2810–2819. PMLR, 2019.

Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Q Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In *Advances in Neural Information Processing Systems*, pages 1635–1646, 2019.

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.

Zhen Huang, Nabarun Deb, and Bodhisattva Sen. Kernel partial correlation coefficient–a measure of conditional dependence. *arXiv preprint arXiv:2012.14804*, 2020c.

Hisham Husain, Richard Nock, and Robert C Williamson. A primal-dual link between gans and autoencoders. In *Advances in Neural Information Processing Systems*, pages 413–422, 2019.

Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. Targeted attacks on deep reinforcement learning agents through adversarial observations. *arXiv preprint arXiv:1905.12282*, 2019.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, pages 2137–2146, 2018a.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018b.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

Nicole Immorlica, Karthik Abinav Sankararaman, Robert E. Schapire, and Aleksandrs Slivkins. Adversarial bandits with knapsacks. *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 202–219, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks, 2017.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

W. James and Charles Stein. Estimation with quadratic loss. In *Proc. of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 361–379. University of California Press, 1961. URL https://projecteuclid.org/euclid.bsmsp/1200512173.

Hicham Janati, Thomas Bazeille, Bertrand Thirion, Marco Cuturi, and Alexandre Gramfort. Multi-subject meg/eeg source imaging with sparse multi-task regression. *NeuroImage*, page 116847, 2020.

Adel Javanmard and Mohammad Mehrabi. Pearson chi-squared conditional randomization test. *arXiv preprint arXiv:2111.00027*, 2021.

Mohamed Jebalia and Anne Auger. Log-linear Convergence of the Scale-invariant $(\mu/\mu_w, \lambda)$-ES and Optimal μ for Intermediate Recombination for Large Population Sizes. In Robert Schaefer, Carlos Cotta, Joanna Kolodziej, and Günter Rudolph, editors, *Parallel Problem Solving From Nature (PPSN2010)*, Lecture Notes in Computer Science, pages xxxx–xxx, Krakow, Poland, September 2010. Springer. URL https://hal.inria.fr/inria-00494478.

Ahmadreza Jeddi, M. Shafiee, and A. Wong. A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning. *ArXiv*, abs/2012.13628, 2020.

Saumya Jetley, Nicholas A. Lord, and Philip H.S. Torr. With friends like these, who needs adversaries? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 10772–10782, Red Hook, NY, USA, 2018. Curran Associates Inc.

Kui Jia, Dacheng Tao, Shenghua Gao, and Xiangmin Xu. Improving training of deep neural networks via singular value bounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 864–872, 2019.

Ruiwei Jiang and Yongpei Guan. Risk-averse two-stage stochastic program with distributional ambiguity. *Operations Research*, 66(5):1390–1405, 2018.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. arxiv, pages arxiv–1907. 2019.

Wittawat Jitkrittum, Zoltán Szabó, and Arthur Gretton. An adaptive test of independence with analytic kernel embeddings. JMLR, 2017.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, Dec 1998. ISSN 1573-2916.

Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3640–3649, 2018.

Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Geometric robustness of deep networks: analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4441–4449, 2018.

Leonid Kantorovich. On the transfer of masses (in russian). *Doklady Akademii Nauk*, 37(2): 227–229, 1942.

Sayash Kapoor, Kumar Kshitij Patel, and Purushottam Kar. Corruption-tolerant bandit learning. *Machine Learning*, 108(4):687–715, 2019.

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

Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

Carson Kent, Jose Blanchet, and Peter Glynn. Frank-wolfe methods in probability space. *arXiv preprint arXiv:2105.05352*, 2021.

Pascal Kerschke, Holger H Hoos, Frank Neumann, and Heike Trautmann. Automated algorithm selection: Survey and perspectives. *Evolutionary computation*, 27(1):3–45, 2019.

Justin Khim and Po-Ling Loh. Adversarial risk bounds for binary classification via function transformation. *arXiv preprint arXiv:1810.09519*, 2, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

A. Krizhevsky. Learning multiple layers of features from tiny images. In *Learning Multiple Layers of Features from Tiny Images*, 2009.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. In *International Conference on Machine Learning*, 2020a.

Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 69–75. IEEE, 2020b.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Adeline Langlois, Damien Stehlé, and Ron Steinfeld. Gghlite: More efficient multilinear maps from ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 239–256. Springer, 2014.

Tor Lattimore and Csaba Szepesvári. Bandit algorithms. Pre-publication version, 2018. URL `http://downloads.tor-lattimore.com/banditbook/book.pdf`.

Tor Lattimore, Koby Crammer, and Csaba Szepesvári. Linear multi-resource allocation with semi-bandit feedback. In *Advances in Neural Information Processing Systems*, pages 964–972, 2015.

Julien Launay, Iacopo Poli, Franccois Boniface, and F. Krzakala. Direct feedback alignment scales to modern deep learning tasks and architectures. *ArXiv*, abs/2006.12878, 2020.

Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

Mathias Lechner, Ramin Hasani, Radu Grosu, Daniela Rus, and Thomas A Henzinger. Adversarial training is not ready for robot learning. *arXiv preprint arXiv:2103.08187*, 2021.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 727–743, 2018.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

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

Bing Li. *Sufficient dimension reduction: Methods and applications with R*. CRC Press, 2018.

Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in neural information processing systems*, pages 1885–1893, 2016.

Chun Li and Xiaodan Fan. On nonparametric conditional independence tests for continuous variables. *Wiley Interdisciplinary Reviews: Computational Statistics*, 12(3):e1489, 2020.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.500. URL `https://aclanthology.org/2020.emnlp-main.500`.

Mingjie Li, Lingshen He, and Zhouchen Lin. Implicit euler skip connections: Enhancing adversarial robustness via numerical stability. In *International Conference on Machine Learning*, pages 5874–5883. PMLR, 2020b.

Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Joern-Henrik Jacobsen. Preventing gradient attenuation in lipschitz constrained convolutional networks. In *Advances in Neural Information Processing Systems*, 2019b.

Ruilin Li, Xiaojing Ye, Haomin Zhou, and Hongyuan Zha. Learning to match via inverse optimal transport. *J. Mach. Learn. Res.*, 20:80–1, 2019c.

Xiaodong Li, Ke Tang, Mohammad N Omidvar, Zhenyu Yang, Kai Qin, and Hefei China. Benchmark functions for the cec 2013 special session and competition on large-scale global optimization. *gene*, 7(33):8, 2013.

Yingkai Li, Edmund Y Lou, and Liren Shan. Stochastic linear optimization with adversarial corruption. *arXiv preprint arXiv:1909.02109*, 2019d.

T. Lillicrap, D. Cownden, D. Tweed, and C. Akerman. Random feedback weights support learning in deep neural networks. *ArXiv*, abs/1411.0247, 2014.

Chang Liu, Bo Li, Yevgeniy Vorobeychik, and Alina Oprea. Robust linear regression against training data poisoning. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 91–102, 2017.

Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. *arXiv preprint arXiv:1905.06494*, 2019.

Jialin Liu, Antoine Moreau, Mike Preuss, Jeremy Rapin, Baptiste Roziere, Fabien Teytaud, and Olivier Teytaud. Versatile black-box optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 620–628, 2020.

Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. signSGD via zeroth-order oracle. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJe-DsC5Fm.

Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *European Conference on Computer Vision*, pages 381–397. Springer, 2018.

Phil Long and Rocco Servedio. Consistency versus realizable h-consistency for multiclass classification. In *International Conference on Machine Learning*, pages 801–809. PMLR, 2013.

Ilya Loshchilov. A computationally efficient limited memory cma-es for large scale optimization. In *Proc. of Genetic and Evolutionary Computation (GECCO'14)*, pages 397–404, 2014.

Ilya Loshchilov, Tobias Glasmachers, and Hans-Georg Beyer. Large scale black-box optimization by limited-memory matrix adaptation. *IEEE Transactions on Evolutionary Computation*, 23 (2):353–358, 2018.

Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, 2005.

Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 114–122, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355599. doi: 10.1145/3188745.3188918. URL https://doi.org/10.1145/3188745.3188918.

Thodoris Lykouris, Max Simchowitz, Aleksandrs Slivkins, and Weidong Sun. Corruption robust exploration in episodic reinforcement learning. *ArXiv*, abs/1911.08689, 2019.

Yuzhe Ma, Kwang-Sung Jun, Lihong Li, and Xiaojin Zhu. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, pages 186–204. Springer, 2018.

Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. Policy poisoning in batch reinforcement learning and control. In *Advances in Neural Information Processing Systems*, pages 14543–14553, 2019.

H. Maaranen, K. Miettinen, and M.M. Mäkelä. Quasi-random initial population for genetic algorithms. *Computers and Mathematics with Applications*, 47(12):1885–1895, 2004.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. ACL. URL http://www.aclweb.org/anthology/P11-1015.

Erika Mackin and Lirong Xia. Allocating indivisible items in categorized domains. *arXiv preprint arXiv:1504.05932*, 2015.

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*, 2018a.

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*, 2018b.

S. Mahdavi, S. Rahnamayan, and K. Deb. Center-based initialization of cooperative co-evolutionary algorithm for large-scale optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 3557–3565, 2016.

Katherine M Malan and Andries P Engelbrecht. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences*, 241:148–163, 2013.

Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning, 2018.

Florian Markowetz and Rainer Spang. Inferring cellular networks–a review. *BMC bioinformatics*, 8(6):1–17, 2007.

J. Matoušek. *Geometric Discrepancy*. Springer, 2nd edition, 2010.

J. Matyas. Random optimization. *Automation and Remote control*, 26:246–253, 1965.

John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. A proposal for the dartmouth summer research project on artificial intelligence. *AI magazine*, 27(4):12–12, 1955.

M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2): 239–245, 1979a.

Michael D. McKay, Richard J. Beckman, and William J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21:239–245, 1979b. ISSN 00401706.

Bhaskar Mehta and Wolfgang Nejdl. Attack resistant collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 75–82, 2008.

Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. Exploratory landscape analysis. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 829–836, 2011.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*, 2017. URL `https://arxiv.org/abs/1702.04267`.

Laurent Meunier, Jamal Atif, and Olivier Teytaud. Yet another but more efficient black-box adversarial attack: tiling and evolution strategies. *arXiv preprint arXiv:1910.02244*, 2019.

Laurent Meunier, Yann Chevaleyre, Jérémy Rapin, Clément W. Royer, and Olivier Teytaud. On averaging the best samples in evolutionary computation. In Thomas Bäck, Mike Preuss, André H. Deutz, Hao Wang, Carola Doerr, Michael T. M. Emmerich, and Heike Trautmann, editors, *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part II*, volume 12270 of *Lecture Notes in Computer Science*, pages 661–674. Springer, 2020a. doi: 10.1007/978-3-030-58115-2_46. URL `https://doi.org/10.1007/978-3-030-58115-2_46`.

Laurent Meunier, Carola Doerr, Jeremy Rapin, and Olivier Teytaud. Variance reduction for better sampling in continuous domains, 2020b.

Laurent Meunier, Carola Doerr, Jeremy Rapin, and Olivier Teytaud. Variance reduction for better sampling in continuous domains. In *International Conference on Parallel Problem Solving from Nature*, pages 154–168. Springer, 2020c.

Laurent Meunier, Blaise Delattre, Alexandre Araujo, and Alexandre Allauzen. A dynamical system perspective for lipschitz neural networks, 2021a. URL `https://arxiv.org/absw/2110.12690`.

Laurent Meunier, Iskander Legheraba, Yann Chevaleyre, and Olivier Teytaud. Asymptotic convergence rates for averaging strategies. In *Proceedings of the 16th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, pages 1–11, 2021b.

Laurent Meunier, Herilalaina Rakotoarison, Pak Kan Wong, Baptiste Roziere, Jeremy Rapin, Olivier Teytaud, Antoine Moreau, and Carola Doerr. Black-box optimization revisited: Improving algorithm selection wizards through massive benchmarking. *IEEE Transactions on Evolutionary Computation*, 2021c.

Laurent Meunier, Herilalaina Rakotoarison, Pak Kan Wong, Baptiste Roziere, Jérémy Rapin, Olivier Teytaud, Antoine Moreau, and Carola Doerr. Black-Box Optimization Revisited: Improving Algorithm Selection Wizards through Massive Benchmarking, May 2021d. URL https://doi.org/10.5281/zenodo.4744977.

Laurent Meunier, Meyer Scetbon, Rafael B Pinot, Jamal Atif, and Yann Chevaleyre. Mixed nash equilibria in the adversarial examples game. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7677–7687. PMLR, 18–24 Jul 2021e. URL https://proceedings.mlr.press/v139/meunier21a.html.

Sanya Mitaim and Bart Kosko. Adaptive stochastic resonance. *Proceedings of the IEEE*, 86(11): 2152–2183, 1998.

Nikolas Mittag. A nonparametric k-sample test of conditional independence. 2018.

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

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. 2018.

D. Molina, M. Lozano, and F. Herrera. Memetic algorithm with local search chaining for continuous optimization problems: A scalability test. In *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pages 1068–1073, 2009.

Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4636–4645, Long Beach, California, USA, 09–15 Jun 2019. PMLR, PMLR. URL http://proceedings.mlr.press/v97/moon19a.html.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94. Ieee, 2017.

Herve Moulin. *Game theory for the social sciences*. NYU press, 1986.

Hervé Moulin. *Fair division and collective welfare*. MIT press, 2004.

Youssef Mroueh and Tom Sercu. Fisher gan. In *Advances in Neural Information Processing Systems*, pages 2513–2523, 2017.

Ronghui Mu, Wenjie Ruan, Leandro Soriano Marcolino, and Qiang Ni. Sparse adversarial video attacks with spatial transformations. *arXiv preprint arXiv:2111.05468*, 2021.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

Mario A Muñoz, Yuan Sun, Michael Kirley, and Saman K Halgamuge. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences*, 317:224–245, 2015.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29, 2016.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

John A. Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103 (1):127–152, 2005.

Matey Neykov, Sivaraman Balakrishnan, and Larry Wasserman. Minimax optimal conditional independence testing. *The Annals of Statistics*, 49(4):2151–2177, 2021.

Harald Niederreiter. *Random Number Generation and quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992. ISBN 0-89871-295-5.

Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014–. URL https://github.com/fmfn/BayesianOptimization.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *NIPS*, 2016.

Ruben Ohana, Jonas Wacker, Jonathan Dong, Sébastien Marmin, Florent Krzakala, Maurizio Filippone, and Laurent Daudet. Kernel computations from large-scale random features obtained by optical processing units. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9294–9298. IEEE, 2020.

Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *J. Mach. Learn. Res.*, 18: 18:1–18:65, 2017. URL http://jmlr.org/papers/v18/14-467.html.

Narayana Prasad Padhy. Unit commitment-a bibliographical survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205, 2004.

N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 372–387, March 2016. doi: 10.1109/EuroSP.2016.36.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016a.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016b.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016c.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pages 506–519, New York, NY, USA, 2017a. ACM. ISBN 978-1-4503-4944-4. doi: 10.1145/3052973.3053009. URL http://doi.acm.org/10.1145/3052973.3053009.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017b.

Haekyu Park, Jinhong Jung, and U Kang. A comparative study of matrix factorization and random walk with restart in recommender systems. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 756–765. IEEE, 2017.

Junhyung Park and Krikamol Muandet. A measure-theoretic approach to kernel conditional mean embeddings. In *Advances in Neural Information Processing Systems*, 2020.

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.

François-Pierre Paty and Marco Cuturi. Subspace robust wasserstein distances. *arXiv preprint arXiv:1901.08949*, 2019.

Judea Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Juan C. Perdomo and Yaron Singer. Robust attacks against multiple classifiers. *arXiv preprint arXiv:1906.02816*, 2019.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Mathis Petrovich, Chao Liang, Yanbin Liu, Yao-Hung Hubert Tsai, Linchao Zhu, Yi Yang, Ruslan Salakhutdinov, and Makoto Yamada. Feature robust optimal transport for high-dimensional data. *arXiv preprint arXiv:2005.12123*, 2020.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization: the case of the exponential family. *arXiv preprint arXiv:1902.01148*, 2019.

Rafael Pinot, Raphael Ettedgui, Geovani Rizk, Yann Chevaleyre, and Jamal Atif. Randomization matters how to defend against strong adversarial attacks. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Rafael Pinot, Laurent Meunier, Florian Yger, Cédric Gouy-Pailler, Yann Chevaleyre, and Jamal Atif. On the robustness of randomized classifiers to adversarial examples. *arXiv preprint arXiv:2102.10875*, 2021.

Erik Pitzer and Michael Affenzeller. A comprehensive survey on fitness landscape analysis. *Recent advances in intelligent engineering systems*, pages 161–191, 2012.

M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964.

M. J. D. Powell. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, pages 51–67. Springer Netherlands, Dordrecht, 1994. ISBN 978-94-015-8330-5.

Muni Sreenivas Pydi and Varun Jog. Adversarial risk via optimal transport and optimal couplings. *IEEE Transactions on Information Theory*, 67(9):6031–6052, 2021a. doi: 10.1109/TIT.2021.3100107.

Muni Sreenivas Pydi and Varun Jog. The many faces of adversarial risk. *Advances in Neural Information Processing Systems*, 34, 2021b.

N. J. Radcliffe and P. D. Surry. Formal memetic algorithms. In T.C. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, pages 1–16. Springer Verlag LNCS 865, 1994.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAi, 2018.

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

Hamed Rahimian, Güzin Bayraksan, and Tito Homem-de Mello. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. *Mathematical Programming*, 173(1):393–430, 2019.

S. Rahnamayan and G. G. Wang. Center-based sampling for population-based algorithms. In *2009 IEEE Congress on Evolutionary Computation*, pages 933–938, May 2009. doi: 10.1109/CEC.2009.4983045.

S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama. Quasi-oppositional differential evolution. In *2007 IEEE Congress on Evolutionary Computation*, pages 2229–2236, Sep. 2007.

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.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

J. Rapin and O. Teytaud. Nevergrad - A gradient-free optimization platform. `https://GitHub.com/FacebookResearch/Nevergrad`, 2018.

Jeremy Rapin and Olivier Teytaud. Dashboard of results for Nevergrad platform. `https://dl.fbaipublicfiles.com/nevergrad/allxps/list.html`, 2020.

Jérémy Rapin, Pauline Dorval, Jules Pondard, Nicolas Vasilache, Marie-Liesse Cauwet, Camille Couprie, and Olivier Teytaud. Openly revisiting derivative-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 267–268, 2019.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.

Ingo Rechenberg. Evolution strategy: Nature's way of optimization. In *Optimization: Methods and applications, possibilities and limitations*, pages 106–126. Springer, 1989.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Maria Refinetti, Stéphane d'Ascoli, Ruben Ohana, and Sebastian Goldt. The dynamics of learning with feedback alignment. *arXiv preprint arXiv:2011.12428*, 2020.

Alfréd Rényi. On measures of entropy and information. Technical report, HUNGARIAN ACADEMY OF SCIENCES Budapest Hungary, 1961.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, 2015.

John R Rice. The algorithm selection problem. In *Advances in computers*, volume 15, pages 65–118. Elsevier, 1976.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

Sylvia Richardson and Walter R Gilks. A bayesian approach to measurement error problems in epidemiology using conditional independence models. *American Journal of Epidemiology*, 138 (6):430–442, 1993.

Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

Hartley Rogers Jr. *Theory of recursive functions and effective computability*. MIT press, 1987.

Raymond Ros and Nikolaus Hansen. A simple modification in cma-es achieving linear time and space complexity. In Günter Rudolph, Thomas Jansen, Nicola Beume, Simon Lucas, and Carlo Poloni, editors, *Parallel Problem Solving from Nature – PPSN X*, pages 296–305, Berlin, Heidelberg, 2008. Springer, Springer Berlin Heidelberg. ISBN 978-3-540-87700-4.

S. Rota Bulò, B. Biggio, I. Pillai, M. Pelillo, and F. Roli. Randomized prediction games for adversarial machine learning. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11): 2466–2478, 2017.

Deboleena Roy, Indranil Chakraborty, Timur Ibrayev, and Kaushik Roy. Robustness hidden in plain sight: Can analog computing defend against adversarial attacks? *arXiv preprint arXiv:2008.12016*, 2020.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *NIPS*, pages 3215–3225, 2017.

Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. Finding global minima via kernel approximations. *arXiv preprint arXiv:2012.11978*, 2020.

Jakob Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 938–947. PMLR, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs using optimal transport. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkQkBnJAb.

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.

Hadi Salman, Andrew Ilyas, L. Engstrom, Ashish Kapoor, and A. Madry. Do adversarially robust imagenet models transfer better? *ArXiv*, abs/2007.08489, 2020.

Ralf Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3):263–278, 1996.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

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

Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. *arXiv preprint arXiv:2110.11773*, 2021b.

M. Scetbon and G. Varoquaux. Comparing distributions: ℓ_1 geometry improves kernel two-sample testing, 2019a.

Meyer Scetbon and Marco Cuturi. Linear time sinkhorn divergences using positive features, 2020.

Meyer Scetbon and Gael Varoquaux. Comparing distributions: ℓ_1 geometry improves kernel two-sample testing. In *Advances in Neural Information Processing Systems*, volume 32, pages 12327–12337, 2019b.

Meyer Scetbon, Laurent Meunier, Jamal Atif, and Marco Cuturi. Equitable and optimal transport with multiple agents. In *International Conference on Artificial Intelligence and Statistics*, pages 2035–2043. PMLR, 2021a.

Meyer Scetbon, Laurent Meunier, and Yaniv Romano. An ℓ^p-based kernel conditional independence test. *arXiv preprint arXiv:2110.14868*, 2021b.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5014–5026, 2018.

M. Schumer and K. Steiglitz. Adaptive step size random search. *Automatic Control, IEEE Transactions on*, 13:270–276, 1968.

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

V. Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, M. Chiang, and Prateek Mittal. Improving adversarial robustness using proxy distributions. *ArXiv*, abs/2104.09425, 2021.

Rajat Sen, Ananda Theertha Suresh, Karthikeyan Shanmugam, Alexandros G. Dimakis, and Sanjay Shakkottai. Model-powered conditional independence test, 2017.

Rajat Sen, Karthikeyan Shanmugam, Himanshu Asnani, Arman Rahimzamani, and Sreeram Kannan. Mimic and classify: A meta-algorithm for conditional independence testing. *arXiv preprint arXiv:1806.09708*, 2018.

Ozan Sener and Vladlen Koltun. Learning to guide random search. In *Proc. of 8th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2020a. URL https://openreview.net/forum?id=B1gHokBKwS.

Ozan Sener and Vladlen Koltun. Learning to guide random search. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=B1gHokBKwS.

A. Shafahi, Parsa Saadatpanah, C. Zhu, Amin Ghiasi, C. Studer, D. Jacobs, and T. Goldstein. Adversarially robust transfer learning. *ArXiv*, abs/1905.08232, 2020.

Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *International Conference on Learning Representtation*, 2018.

Soroosh Shafieezadeh Abadeh, Peyman M Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. *Advances in Neural Information Processing Systems*, 28, 2015.

Rajen D. Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48, Jun 2020.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Shai Shalev-Shwartz and Yoram Singer. Efficient learning of label ranking by soft projections onto polyhedra. *Journal of Machine Learning Research*, 7(Jul):1567–1599, 2006.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

Tianhong Sheng and Bharath K. Sriperumbudur. On distance and kernel measures of conditional independence. *arXiv: Statistics Theory*, 2019.

Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

Chengchun Shi, Tianlin Xu, Wicher Bergsma, and Lexin Li. Double generative adversarial networks for conditional independence testing. *Journal of Machine Learning Research*, 22(285): 1–32, 2021a.

Hongjian Shi, Mathias Drton, and Fang Han. On azadkia-chatterjee's conditional dependence coefficient. *arXiv preprint arXiv:2108.06827*, 2021b.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

Carl-Johann Simon-Gabriel, Yann Ollivier, Leon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In *International Conference on Machine Learning*, pages 5809–5817, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sahil Singla and Soheil Feizi. Skew orthogonal convolutions. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Sahil Singla, Surbhi Singla, and Soheil Feizi. Householder activations for provable robustness against adversarial attacks. *arXiv preprint arXiv:2108.04062*, 2021a.

Sahil Singla et al. Fantastic four: Differentiable and efficient bounds on singular values of convolution layers. In *International Conference on Learning Representations*, 2021b.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Maurice Sion. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958. URL https://projecteuclid.org:443/euclid.pjm/1103040253.

Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*, 2018.

Zbigniew Maciej Skolicki. *An Analysis of Island Models in Evolutionary Computation*. PhD thesis, USA, 2007. AAI3289714.

Urban Škvorc, Tome Eftimov, and Peter Korošec. Gecco black-box optimization competitions: progress from 2009 to 2018. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 275–276, 2019.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, NIPS'12, pages 2951–2959, 2012.

Marilda Sotomayor and Alvin Roth. Two-sided matching: A study in game-theoretic modelling and analysis. *Econometric Society Monographs*, (18), 1990.

Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, Gert RG Lanckriet, et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

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.

Ton Steerneman. On the total variation and hellinger distance between signed measures; an application to product measures. *Proceedings of the American Mathematical Society*, 88(4):684–688, 1983.

Charles Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proc. of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 197–206. University of California Press, 1956. URL https://projecteuclid.org/euclid.bsmsp/1200501656.

H. Steinhaus. Sur la division pragmatique. *Econometrica*, 17:315–319, 1949. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1907319.

Ingo Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26(2):225–287, 2007.

Rainer Storn and Kenneth Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, December 1997. ISSN 0925-5001.

Eric V Strobl, Kun Zhang, and Shyam Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *Journal of Causal Inference*, 7(1), 2019.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

Bruno Sudret. Meta-models for structural reliability and uncertainty quantification, 2012.

Haodong Sun, Haomin Zhou, Hongyuan Zha, and Xiaojing Ye. Learning cost functions for optimal transport. *arXiv preprint arXiv:2002.09650*, 2020a.

Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. Stealthy and efficient adversarial attacks against deep reinforcement learning. *To appear in Proceedings of the AAAI Conference on Artificial Intelligence*, 2020b.

Patrick D. Surry and Nicholas J. Radcliffe. Inoculation to initialise evolutionary search. In Terence C. Fogarty, editor, *Evolutionary Computing*, pages 269–285, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg. ISBN 978-3-540-70671-7.

Zoltán Szabó and Bharath Sriperumbudur. Characteristic and universal tensor product kernels. *Journal of Machine Learning Research*, 18:233, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Robert E. Tarjan. Dynamic trees as search trees via euler tours, applied to the network simplex algorithm. *Mathematical Programming*, 78(2):169–177, 1997.

F. Teytaud. A new selection ratio for large population sizes. *Applications of Evolutionary Computation*, pages 452–460, 2010. URL http://hal.inria.fr/inria-00456335/PDF/SAlarge_2_.pdf.

Fabien Teytaud and Olivier Teytaud. Why one must use reweighting in estimation of distribution algorithms. In *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009*, pages 453–460, 2009.

Olivier Teytaud. Conditioning, halting criteria and choosing lambda. In *EA07*, Tours, France, 2007. URL https://hal.inria.fr/inria-00173237.

Olivier Teytaud, Sylvain Gelly, and Jérémie Mary. On the ultimate convergence rates for isotropic algorithms and the best choices among various forms of isotropy. In *Proceedings of PPSN*, pages 32–41, 2006. doi: 10.1007/11844297_4. URL https://doi.org/10.1007/11844297_4.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

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.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

Nicolás García Trillos and Ryan Murray. Adversarial classification: Necessary conditions and geometric flows. *arXiv preprint arXiv:2011.10797*, 2020.

Asher Trockman et al. Orthogonalizing convolutional layers with the cayley transform. In *International Conference on Learning Representations*, 2021.

Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 1, 2008.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *International Conference on Learning Representation*, 2019.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

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

Hoang Tuy. Dc optimization: theory, methods and algorithms. In *Handbook of global optimization*, pages 149–216. Springer, 1995.

I. Vajda. Note on discrimination information and variation. *IEEE Trans. Inform. Theory*, 16(6): 771–773, Nov. 1970.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Mauro Vallati, Frank Hutter, Lukás Chrpa, and Thomas Leo McCluskey. On the effective configuration of planning domain models. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI'15)*, 2015.

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.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

T. van Erven and P. Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

Vladimir N. Vapnik. *Statistical Learning Theory*. 1998.

J.M. Varah. A lower bound for the smallest singular value of a matrix. *Linear Algebra and its Applications*, 11(1):3–5, 1975. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(75)90112-3. URL http://www.sciencedirect.com/science/article/pii/0024379575901123.

Konstantinos Varelas, Ouassim Ait ElHara, Dimo Brockhoff, Nikolaus Hansen, Duc Manh Nguyen, Tea Tusar, and Anne Auger. Benchmarking large-scale continuous optimizers: The BBOB-largescale testbed, a COCO software guide and beyond. *Appl. Soft Comput.*, 97(Part):106737, 2020. doi: 10.1016/j.asoc.2020.106737. URL https://doi.org/10.1016/j.asoc.2020.106737.

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.

Gunjan Verma and Ananthram Swami. Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8646–8656. Curran Associates, Inc., 2019.

Paul Viallard, Eric Guillaume VIDOT, Amaury Habrard, and Emilie Morvant. A pac-bayes analysis of adversarial robustness. *Advances in Neural Information Processing Systems*, 34, 2021.

Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, 2018.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, art. arXiv:1907.10121, Jul 2019.

John Von Neumann. Uber ein okonomsiches gleichungssystem und eine verallgemeinering des browerschen fixpunktsatzes. In *Erge. Math. Kolloq.*, volume 8, pages 73–83, 1937.

James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. A mathematical theory of attention. *arXiv preprint arXiv:2007.02876*, 2020.

Haibin Wang, Sujoy Sikdar, Xiaoxi Guo, Lirong Xia, Yongzhi Cao, and Hanpin Wang. Multi-type resource allocation with partial preferences. *arXiv preprint arXiv:1906.06836*, 2019a.

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*, 2020a.

Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. *Advances in Neural Information Processing Systems*, 33:19511–19522, 2020b.

Weiran Wang and Miguel A. Carreira-Perpinan. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application, 2013.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019b.

Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *International Conference on Machine Learning*, pages 5133–5142. PMLR, 2018.

Andrew Warren. Wasserstein conditional independence testing. *arXiv preprint arXiv:2107.14184*, 2021.

Andre Wibisono. Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In *Conference on Learning Theory*, pages 2093–3027. PMLR, 2018.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Rey Reza Wiyatno and Anqi Xu. Physical adversarial textures that fool visual object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4822–4831, 2019.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295, 2018.

Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, pages 8400–8409, 2018.

Eric Wong, Frank R. Schmidt, and J. Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations, 2019.

Eric Wong, Leslie Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. *ArXiv*, abs/2001.03994, 2020.

Dongxian Wu, Shutao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *arXiv: Learning*, 2020.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, 2018.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

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.

Lin Xu, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Satzilla: portfolio-based algorithm selection for sat. *Journal of artificial intelligence research*, 32:565–606, 2008.

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*, 2020a.

Karren Dai Yang, Karthik Damodaran, Saradha Venkatachalapathy, Ali C Soylemezoglu, GV Shivashankar, and Caroline Uhler. Predicting cell lineages using autoencoders and optimal transport. *PLoS computational biology*, 16(4):e1007828, 2020b.

X. Yang, J. Cao, K. Li, and P. Li. Improved opposition-based biogeography optimization. In *The Fourth International Workshop on Advanced Computational Intelligence*, pages 642–647, 2011.

Deng Yao, Zheng Xi, Zhang Tianyi, Chen Chen, Lou Guannan, and Kim Miryung. An analysis of adversarial attacks and defenses on autonomous driving models. In *18th Annual IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2020.

Xin Yao and Yong Liu. Fast evolutionary programming. In *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 451–460. MIT Press, 1996.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International Conference on Machine Learning*, pages 7085–7094, 2019.

Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

Man-Chung Yue, Daniel Kuhn, and Wolfram Wiesemann. On linear optimization over wasserstein balls. *arXiv preprint arXiv:2004.07162*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87.

Anru Zhang and Yuchen Zhou. On the non-asymptotic and sharp lower tail bounds of random variables, 2018.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *International conference on Machine Learning*, 2019a.

Jingfeng Zhang, Jianing Zhu, Gang Niu, B. Han, M. Sugiyama, and M. Kankanhalli. Geometry-aware instance-reweighted adversarial training. *ArXiv*, abs/2010.01736, 2020.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*, 2012.

Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1381–1396. IEEE, 2019b.

Qinyi Zhang, Sarah Filippi, Seth Flaxman, and D. Sejdinovic. Feature-to-feature regression for a two-step conditional independence test. In *UAI*, 2017.

Tong Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5(Oct):1225–1251, 2004a.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004b.

Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

Steeve Zozor and P-O Amblard. Stochastic resonance in discrete time nonlinear AR(1) models. *IEEE transactions on Signal Processing*, 47(1):108–122, 1999.