Robust Optimization for Adversarial Deep Learning

Investigation of Danskin's Theorem and BOBYQA-based black-box attacks in the context of image classification by deep neural networks from the perspective of robust optimization

Master thesis by Lars Steffen März (ORCID-iD: 0009-0008-7402-8731)

Date of submission: October 19, 2023

1. Review: Prof. Dr. Stefan Ulbrich
Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Mathematics Department
Optimization

GitHub: https://github.com/lsmaerz/rob_opt_adv_deep_learning.git

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

Hiermit erkläre ich, Lars Steffen März (ORCID-iD: 0009-0008-7402-8731), dass ich die vorliegende Arbeit gemäß § 22 Abs. 7 APB der TU Darmstadt selbstständig, ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe mit Ausnahme der zitierten Literatur und anderer in der Arbeit genannter Quellen keine fremden Hilfsmittel benutzt. Die von mir bei der Anfertigung dieser wissenschaftlichen Arbeit wörtlich oder inhaltlich benutzte Literatur und alle anderen Quellen habe ich im Text deutlich gekennzeichnet und gesondert aufgeführt. Dies gilt auch für Quellen oder Hilfsmittel aus dem Internet.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, October 19, 2023

L. März

Abbreviations and Notation

a.e. almost everywhere

a.s. almost surely

BOBYQA bound optimization by quadratic approximation

CCG concise confidence gain

CDF cumulative distribution function

CEL cross-entropy loss

C&W Carlini-Wagner

DNN deep neural network

DRO distributionally robust optimization

EAD elastic net attacks to DNNs

FGSM fast gradient sign method

GAS gradient alignment score

i.i.d. independent and identically distributed

ILSVRC ImageNet Large Scale Visual Recognition Challenge

KLD Kullback-Leibler divergence

MACER robust training via maximizing the certified radius

MDI mean decrease in impurity

PASS perceptual similarity score

PGD projected gradient descent

PCA principle component analysis

ReLU rectified linear unit

RO robust optimization

RV random variable

SCG simple confidence gain

SLLN strong law of large numbers

SO stochastic optimization

SSIM structural similarity index

Symbol	Value Specification	Meaning
n	$= 3 \cdot$ number of pixels $\in \mathbb{N}$	number of color channel values per image
$[m]$	$= \{1, \dots, m\}$	set of positive integers up to m
sign	$\mathbb{R}^n \to \{-1, 0, 1\}^n$	sign function
ReLU$(\cdot)$	$= (\cdot)_+$	rectified linear unit (ReLU)
softmax	$\mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^d$	softmax operator
t	$\in \mathbb{R}^+$	softmax temperature
log	$= \log_e = \ln$	natural logarithm
$\mathcal{RV}(\cdot)$		set of ·-valued random variables (RVs)
$\widehat{\mathcal{X}}$	$= \{0, \dots, 255\}^n$	set of feasible color channel values
$\mathcal{X}$	$= [0, 1]^n$	normalized set of relaxed color channel values
$\Pi_R^p(\cdot)$	$= \arg\min_{r \in R} \Vert \cdot - r \Vert_p$	projection of r onto R w.r.t. p-norm
X	$\in \mathcal{RV}(\mathcal{X})$	RV of relaxed color channel values
x	$\in \mathcal{X}$	relaxed color channel values
$\mathcal{C}$	$\subseteq \mathbb{N}$	set of class label
C	$\in \mathcal{RV}(\mathcal{C})$	random variable of class labels
c	$\in \mathcal{C}$	class label
$\mathcal{P}$	$= \{(p_c) \in [0, 1]^{\mathcal{C}} : \sum_c p_c = 1\}$	probability simplex
$p = (p_c)_{c \in \mathcal{C}}$	$= F(x) \in \mathcal{P}$	prediction confidence vector
δ	$\in \mathbb{R}^n$	adversarial perturbation
$\widetilde{x}$	$= x + \delta \in \mathcal{U}_p^{\varepsilon_p}(x)$	adversarial example of x
Θ	$\subseteq \mathbb{R}^{\dim(\Theta)}$	set of feasible deep neural network (DNN) weights
θ	$\in \Theta$	vector of DNN weights
F	$\Theta \times \mathbb{R}^n \to \mathcal{P}$	soft classifier
f	$\Theta \times \mathbb{R}^n \to \mathcal{C}$	hard classifier
F_σ	$\Theta \times \mathbb{R}^n \to \mathcal{P}$	smoothed soft classifier (Def. 3.4)
f_h	$\Theta \times \mathbb{R}^n \to \mathcal{C}$	hard smoothed hard classifier (Def. 3.4)
f_s	$\Theta \times \mathbb{R}^n \to \mathcal{C}$	soft smoothed hard classifier (Def. 3.4)
$\widehat{F_\sigma}(\theta, x)$	$\mathcal{RV}(\mathcal{P})$	approx. smoothed soft classifier (Def. 3.5)
$\widehat{f_h}(\theta, x)$	$\mathcal{RV}(\mathcal{C})$	approx. hard smoothed hard classifier (Def. 3.5)
$\widehat{f}_s(\theta, x)$	$\mathcal{RV}(\mathcal{C})$	approx. soft smoothed hard classifier (Def. 3.5)
N	$\in \mathbb{N}$	cardinality of the training data set
$\{(x_i, c_i) : i \in [N]\}$	$\subseteq \mathcal{X} \times \mathcal{C}$	training data set
$\mathcal{M}(\cdot)$		set of probability measures on the set ·
$\sim$		distributional equality
$\mathbb{D}$	$\sim (X, C), \in \mathcal{M}(\mathbb{R}^n \times \mathcal{C})$	joint distribution of features and labels
dirac$_x(\cdot)$	$= \mathbb{1}[x \in \cdot]$	Dirac measure in x
$\widehat{\mathbb{D}}$	$= N^{-1} \sum_{i \in [N]}$ dirac$_{(x_i, c_i)}$	empirical joint distribution of features and labels
$\mathcal{A}$	$\subseteq \mathcal{M}(\mathbb{R}^n)$	ambiguity set of adversarial distributions
$\mathbb{A}$	$\in \mathcal{A}$	adversarial distribution
ε_p	> 0	p-norm perturbation bound
$\overline{\mathcal{B}}_p^{\varepsilon_p}(x)$	$= \{x + \delta \in \mathbb{R}^n : \Vert\delta\Vert_p \leq \varepsilon_p\}$	closed p-norm ball with radius ε_p around x
$\overline{\mathcal{B}}_p^{\varepsilon_p}$	$= \overline{\mathcal{B}}_p^{\varepsilon_p}(0)$	closed p-norm ball with radius ε_p around 0
$\mathcal{U}_p^{\varepsilon_p}(x)$	$= (\overline{\mathcal{B}}_p^{\varepsilon_p}(x) \cap \mathcal{X}) - x$	adversarial uncertainty set around x
$\mathcal{S}_p^{\varepsilon_p}(x)$	$= \{x + \delta \in \mathbb{R}^n : \Vert\delta\Vert_p = \varepsilon_p\}$	p-norm sphere with radius ε_p around x
$\mathcal{S}_p^{\varepsilon_p}$	$= \mathcal{S}_p^{\varepsilon_p}(0)$	p-norm sphere with radius ε_p around 0
I	$=$ diag$(1, \dots, 1)$	identity matrix of appropriate dimension
$\mathcal{N}(\mu, \Sigma)$	$\in \mathcal{M}(\mathbb{R}^n)$	normal distribution with exp. μ and cov. Σ
Φ	$\mathbb{R} \to [0, 1]$	cumulative distribution function (CDF) of $\mathcal{N}(0, 1)$

Assumption 0.1. Throughout the entire document, whenever the arg max operation is applied to a confidence vector $p \in \mathcal{P}$, we will assume that there is a unique largest component, i.e.

$$\left| \arg\max_{c \in \mathcal{C}} p_c \right| = 1.$$

We can enforce this via some strict ordering $\prec$ on $\mathcal{C}$ and

$$\arg\max_{c \in \mathcal{C}} p_c := \min_{\prec} \left\{ c \in \mathcal{C} : p_c = \max_{c' \in \mathcal{C}} p_{c'} \right\}.$$

This significantly improves readability in several formulas without introducing significant notational inaccuracy, since such decision boundaries are lower-dimensional Lebesgue nullsets in the input space $\mathcal{X} \subset \mathbb{R}^n$. This comes from the fact that the logits $F_{K-1}(x)$ of a ReLU-based K-layer DNN $F = F_K$ are computed by a series of piecewise affine linear transformations $F_1, \ldots, F_{K-1}$. Therefore, the points where the largest logit $F_{K-1}(\theta, x)_c$ is not unique, form lower-dimensional planes in the input space. Finally, the softmax layer F_K preserves monotonicity, meaning that these planes coincide with the decision boundaries, i.e. the points where there is no unique class of largest prediction confidence $F(\theta, x)_c = p_c$.

Contents

1 Introduction

1.1 Abstract

Recent results demonstrated that images can be adversarially perturbed to a visually indistinguishable extent in order to misguide classifiers with high standard accuracy into making confident misclassifications. Adversarial examples may even be targeted to a class the attacker chooses [93, 100, 47] and transfer between different DNNs in a black-box setting [86, 52, 15, 6, 10, 57, 86, 105, 100, 89, 30, 90, 43, 82, 47], meaning that perturbations computed on one DNN are likely to confuse other DNNs. This poses a concrete and acute security risk in digital domains like content moderation [6], but also in physical contexts like facial recognition and autonomous driving [85, 24] where adversarial samples proved to survive printing and re-capturing [42]. The phenomenon was first discovered in 2014 by Szegedy et al. [86] and has been subject of hundreds of papers ever since, both from an attacker's and a defender's point of view. There seems to be no apparent end to an arms race of frequently published attacks and defenses as no universal, provable and practical prevention method has been developed yet.

In this work, we show that verifying ReLU-based DNNs against adversarial examples is NP-hard (cf. Proposition 1.5). Furthermore, we model the adversarial training problem as a distributionally robust optimization problem (DRO) to provide a formal framework for two of the most promising defenses so far: Randomized fast gradient sign method (FGSM)-based adversarial training and randomized smoothing. Additionally, we propose two step size schemes for multi-step adversarial attacks that yield unprecedented low true-label-confidences (cf. Contribution 4.1). To make p-norm bounded attacks more comparable for different values of p, we define two norm rescaling functions (4.3) and (4.4) before validating them on ImageNet (cf. Contribution 4.2). Moreover, we give an explanation as to why first-order adversarial training is successful from an empirical data augmentation perspective despite lacking the mathematical guarantees from Danskin's Theorem (Theorem 3.1) by analyzing cosine similarities of model parameter gradients on ImageNet (cf. Contribution 4.3). Finally, we give an update on the performance results from [92, 93] of bound optimization by quadratic approximation (BOBYQA) black-box attacks on CIFAR-10 by exposing instances of the two aforementioned state-of-the-art defenses to it (cf. Contribution 4.4).

This thesis is structured as follows: Chapter 1 introduces the concept of adversarial examples, embeds this work into the literature, formulates the problem of interest, presents our theoretical results and summarizes our main contributions. Then, Chapter 2 showcases gradient-based white-box and zeroth-order black-box attacks, in particular BOBYQA. Subsequently, Chapter 3 deals with the defensive strategies of adversarial training and randomized smoothing. In Chapter 4, we numerically derive the four latter contributions from above. Finally, Chapter 5 concludes this thesis and suggests possible directions for future research.

1.2 Adversarial Examples

The phenomenon of adversarial examples for image classification tasks was first discovered by Szegedy et al. [86] in 2014. However, already in the 2000s, analogous observations in other domains like spam filters and malware detection were made [8, 7, 18, 38, 50]. To develop an understanding of adversarial examples, we start with an intuitive definition before formalizing it. In fact, there is no standardized definition to date, but they agree in the following key conditions:

- The clean sample was correctly classified by the DNN.
- The perturbed sample is wrongly classified by the DNN.
- Both samples are visually similar.

There are two immediate problems with this definition resulting from the ambiguity of the term 'visually similar':

(1) **Meaning:** What does 'visually similar' mean? Possible meanings, from most restrictive to most lenient, include:

   (a) Both images are (almost) visually indistinguishable to humans, cf. Figure 1.1 with at most $\varepsilon_\infty = 0.02$ difference in each color channel value of every pixel.

   (b) There are no obvious perturbation artefacts noticeable to humans, i.e. the images might look different but both look natural, cf. Figure 1.2 with at most $\varepsilon_\infty \approx 0.01$ difference in each color channel value of every pixel.

   (c) The perturbed image is still labelled correctly by humans, but is allowed to look artificially altered, cf. Figure 1.3 with at most $\varepsilon_\infty = 0.1$ difference in each color channel value of every pixel.

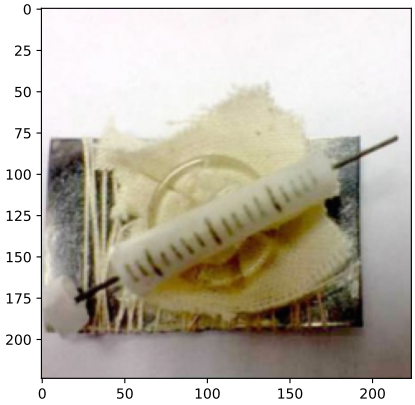
   We will mostly follow (a) and (b) via $\varepsilon_\infty \leq 0.04$ since it poses the strongest constraints and still allows us to compute remarkably successful attacks, regularly with true-label-confidences of 0% up to arbitrarily many decimals with Algorithm 4, as just indicated and discussed in Contribution 4.1.

(2) **Quantification:** How to measure the similarity perceived by humans at scale, i.e. with a computer? As a distance metric, p-norms for $p \in \{1, 2, \infty\}$ have prevailed for that purpose due to being ubiquitous in mathematics and being suitable constraints as they induce convex level sets with continuous boundary. Alternatives include $p = 0$, i.e. there is a maximum number of perturbed pixels and metrics that try to replicate human perception like Voronoi Cells [36] and the perceptual similarity score (PASS)[75] or the structural similarity index (SSIM) [26] because p-norms have been shown to be inappropriate [89] and neither necessary nor sufficient for visual similarity [81].

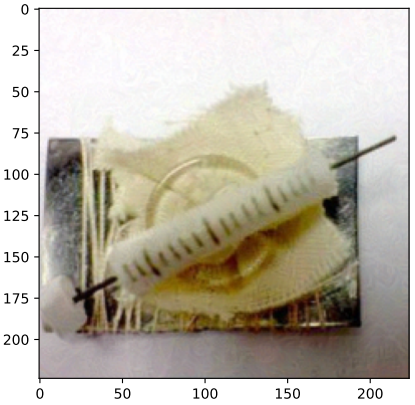
   For comparability, we stay consistent with previous literature and focus on p-norms for $p \in \{1, 2, \infty\}$.

To formalize this intuition, we state the definition of adversarial examples used in this work.

Definition 1.1 (Adversarial Example)**.** *Consider a hard classifier $f : \Theta \times \mathcal{X} \to \mathcal{C}$ and a pair (x, c) of a feature vector with its ground truth label. An input $\widetilde{x} := x + \delta$ with $\delta \in \mathcal{U}_p^{\varepsilon_p}$, i.e. $\|\delta\|_p \leq \varepsilon_p$ and $\widetilde{x} \in \mathcal{X} := [0, 1]^n$, is called an (untargeted) (p, ε_p)-adversarial example with adversarial perturbation δ if $f(\theta, x) = c$ and $f(\theta, \widetilde{x}) \neq c$. If f assigns $\widetilde{x}$ to a specific, previously chosen class $\widetilde{c} \neq c$, then $\widetilde{x}$ is called targeted.*

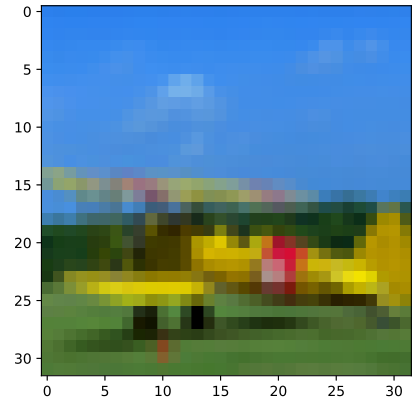


(a) clean, $\varepsilon_\infty = 0$

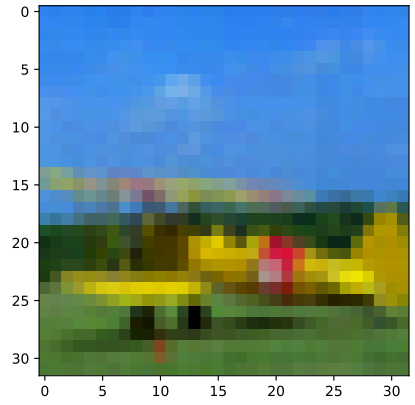


(b) adversarial, $\varepsilon_\infty = 0.02$

Figure 1.1: Picture of a syringe (label 845) from the ImageNet validation data set. Clean image (a) correctly classified by naturally pretrained `PyTorch` DNN with confidence of 65.85% and indistinguishably perturbed sample (b) with 0.00% confidence for 'syringe' and 99.99% confidence for the targeted label 'screw' (label 783).

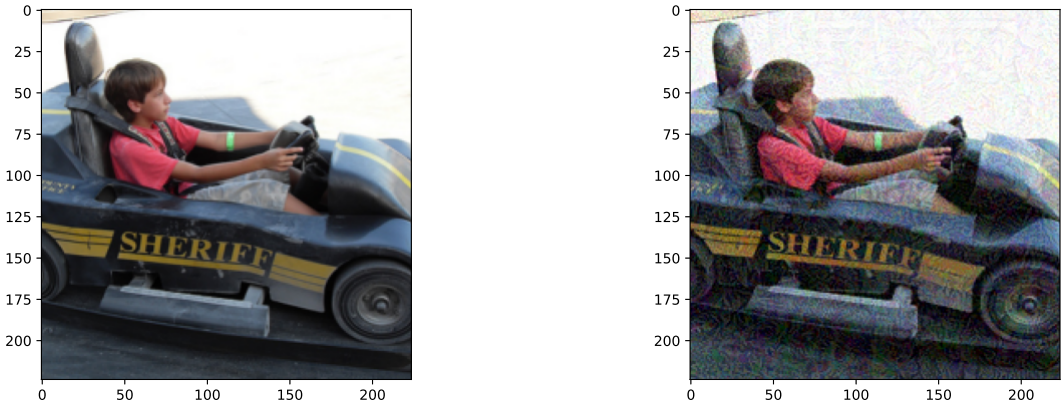


(a) clean, $\varepsilon_\infty = 0$



(b) adversarial, $\varepsilon_\infty \approx 0.01$

Figure 1.2: Picture of an airplane (label 0) from the CIFAR-10 test data set. Clean image (a) correctly classified by naturally trained MadryLab DNN [23] and distinguishable but natural looking sample (b) classified as target label 'ship' (label 8).



(a) clean, $\varepsilon_\infty = 0$ (b) adversarial, $\varepsilon_\infty = 0.1$

Figure 1.3: Picture of a go-kart (label 573) from the ImageNet validation data set. Clean image (a) correctly classified with confidence of 97.74% by naturally pretrained PyTorch DNN and visibly perturbed sample (b) with 0.00% confidence for 'go-kart' and 100.00% confidence for 'rifle' (label 764).

The attacker aims to find such perturbations δ. On the contrary, the defender attempts to train the DNN in a way to achieve a notion of robustness, like the following, against them.

Definition 1.2 (Adversarial Robustness)**.** *A hard classifier $f : \Theta \times \mathbb{R}^n \to \mathcal{C}$ is called (p, ε_p)-robust with perturbation radius $\varepsilon_p \geq 0$ in a point $x \in \mathcal{X}$ if there exist no (p, ε_p)-adversarial example for x. The classifier f is referred to as globally (p, ε_p)-robust (with some confidence) if it is (p, ε_p)-robust in all points $x \in \mathcal{X}$ (or they at least represent a set with empirical probability larger than that confidence in the data set).*

Remark 1.3 (Other Notions of Adversarial Robustness). Note that there are alternative definitions of adversarial robustness. For example, often the maximum or average perturbation radius ε_p for each image to remain correctly classified over a validation set is considered. There are also distributionally robust analogies to the aforementioned point-wise, metric bounded robustness that leverage probability measure metrics [53].

Given the unprecedented success and especially generalization capability of DNNs to unseen images in recent times, it might be surprising that such small image corruptions, that pose no challenge to the human eye, are so detrimental. Naturally, this raises the question where this susceptibility originates from. Up to now, there is no definitive answer as to why the phenomenon of adversarial examples exists. However, there are multiple theories for the root cause. Some of the main lines of argument can be clustered as follows.

- **High Dimensionality:** Images are data of high-dimensional shape. Consequently, [29] sees a high likelihood that the input space contains some direction of steep loss ascent for the classifier. Further, adversarial robustness does not remove adversarial examples but rather makes them harder to find by introducing small deformations into the loss landscape that mask the true gradient, or more precisely, make the gradient less meaningful locally, cf. Section 2.1.2.
- **Low Probability Regions:** Adversarial examples represent blind spots close to the thin empirical data distribution as they are unnaturally perturbed [86, 63, 104, 67]. During training, the classifier accepts

being incorrect there, even with high confidence, since the probability distribution puts low emphasis on these regions. Beyond that, [86] conjectures that the set of such blind spots is dense around the empirical data manifold making it almost always possible to find adversarial examples close to a clean sample.

- **Nearby Decision Boundaries:** In general, feeding into the first two explanations, small distances of natural samples to decision boundaries allow for adversarial examples but also randomly corrupted images to be misclassified [90]. Both phenomenons seem to be highly correlated [27].
- **Model Capabilities:** DNNs behave locally too linearly, according to [30]. Further, robustness increases with and even requires higher model capacity [43, 52]. Fundamentally, [10, 25, 59] show that convolutional neural networks do not understand the concept of the labels but only the visual shapes and patterns, which are sensitive to alterations. As a consequence, DNNs posses inherently wrong decision boundaries [21].

Despite multiple promising explanations and computational results, it is hard to determine a unique, true theory due to their low comparability, which can be attributed to different choices w.r.t. model architecture, learning procedure, data sets, success metrics and the assumed defense and attack taxonomy. Furthermore, the results in this young field tend to be more empirical than theoretical, rarely validated by a third party (one exception is [52]) and, most importantly, often with a short life span as the timeline of defensive distillation showcased [61, 62, 63, 9].

Developing methods to compute strong adversarial examples must not happen with malicious intentions in mind. One the one hand, training with adversarial examples not only increases adversarial robustness [82], but also appears to improve generalizability by acting as a regularizer [69, 80, 32]. On the other hand, attacking DNNs with adversarial examples is the method of choice for upper bounding robustness because rigorous methods using mixed-integer linear programming [49] or satisfiability modulo theories [35], that guarantee lower bounds, are hardly applicable to state-of-the-art DNN sizes from a complexity standpoint.

1.3 Related Work

Locally, this work is most closely related to different sets of papers, depending on the contribution of interest. Our theoretical contributions from Section 1.4 and Proposition 1.5 relate to the entire adversarial deep learning community. The numerical Contributions 4.1 and 4.2, regarding adversarial step lengths for zero true-label-confidence and p-norm rescaling, find application in all fields that investigate p-norm bounded adversarial attacks. Our Contribution 4.3, regarding Danskin's Theorem (Theorem 3.1), specifically provides answers to questions posed in [80] and [95], i.e. why adversarial training increases robustness and when strong adversaries are necessary for training or weak ones suffice. Our response is consistent with the interpretation of adversarial training given in [98]. Lastly, the experiments regarding targeted BOBYQA black-box attacks can be seen as an update of [92, 93] to modern defenses and strengthen the advice from [3] and [89] about incorporating zeroth-order attacks into attack portfolios for evaluating defenses.

More broadly, we want to embed our work regarding BOBYQA attacks (B) and gradient-based adversarial attacks (G) into the threat model taxonomy of Table 1.1.

For almost each combination of the above and even more aspects, [64, 10, 100, 16, 78, 65, 104, 99] include suitable attacks or reference papers dealing with that type of attack. Note that in the cases with an asterisk, the roles of objective function and constraint are reversed. Similarly, the defenses considered here, namely

Attribute	Possibilities
knowledge	white-box (G), gray-box, black-box (B)
goal	confidence reduction, untargeted miscl. (G), targeted miscl. (B), min. perturb.*
image/target	given (B, G), selectable
constraints	p-norm (B, G), none, guaranteed miscl.*, other (e.g. PASS, SSIM, Voronoi Cells)
applicability scope	individual (B, G), universal
derivative degree	zeroth-order (B), first-order (G), second-order

Table 1.1: Threat Model Taxonomy, cf. [64, 100].

Attribute	Possibilities
application time	proactive (A), reactive (S)
robustness guarantees	empirical (A), certified (S)
specificity	attack-sensitive, attack-agnostic (A, S)

Table 1.2: Defense Categorization, cf. [100].

adversarial training (A) [3, 80, 95, 43, 28, 77, 82, 104] and randomized smoothing (S) [57, 69, 101, 60, 16, 70, 78, 27, 103], can be categorized as in Table 1.2.

Within the proactive defenses, i.e. the ones happening prior to query-time, one distinguishes approaches that modify the architecture [101] or - as in the case of (A), defensive distillation [61, 62, 63, 9] and others from [45] - the training procedure. The reactive defenses take place at query-time and roughly split into two groups: They either manipulate all images before evaluating the DNN, as in the case of (S) and feature squeezing [97, 96], or they rely on detecting adversarial examples [51] and reconstructing a clean version [54].

1.4 Problem Formulation

1.4.1 Network Architecture and Layers

To model the adversarial attack and training problems, we start by describing the RGB encoding of images. Each picture has a width and height that determine its total number of pixels. Each pixel's color is represented by its three color channels, namely red, green and blue. They take values in $\{0, \dots, 255\}$ as they allocate eight bits or one byte of storage each. Defining n as the total number of color channel values per image, i.e. the image resolution times three, an image x can be represented as $x \in \widehat{\mathcal{X}} := \{0, \dots, 255\}^n$. During training and attacking, multiple operation like scaling, averaging and convolutions are performed. Under these, $\widehat{\mathcal{X}}$ is not closed. Taking gradients is not even possible in this discrete set. Hence, we will internally work with the normalized relaxation $\mathcal{X} := [0, 1]^n$ instead.

For our purposes, it suffices to define a DNN soft classifier F as a composition of K layers F_k which are parameterized by weights $\theta \in \Theta$. Given an input x, this map

$$F := F_{[K]} := F_1 \circ \dots \circ F_K : \Theta \times \mathbb{R}^n \to \mathcal{P}$$

assigns class probabilities p_c from the probability simplex $\mathcal{P} := \{(p_c) \in [0, 1]^{\mathcal{C}} : \sum_c p_c = 1\}$ to each of the classes in $\mathcal{C}$. To ensure that the sum is normalized, the last layer F_K is a softmax layer, that transforms

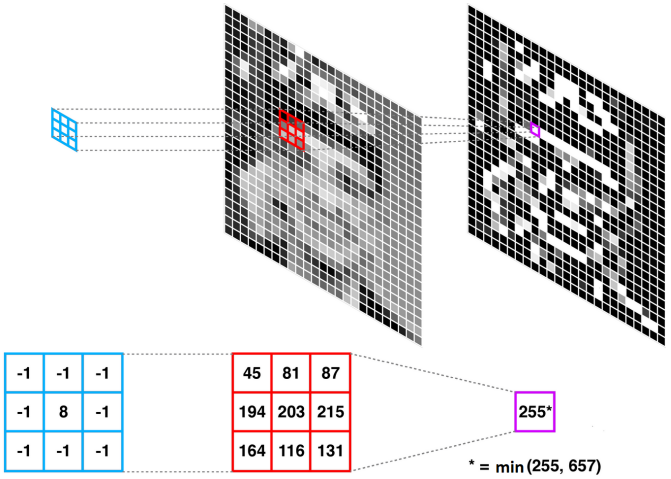


Figure 1.4: Visualization of a 3×3 kernel applied to a picture of a face [33].

the penultimate layer's neurons $F_{K-1}(x)$, called logits, into class probabilities in a monotonic, smooth way. Concretely, we denote

$$p_c := F(\theta, x)_c := \text{softmax}(t, F_{K-1}(\theta, x))_c := \frac{\exp(F_{K-1}(\theta, x)_c/t)}{\sum_{j \in \mathcal{C}} \exp(F_{[K-1]}(\theta, x)_j/t)} \quad \forall c \in \mathcal{C}$$

for some temperature $t > 0$ with more on its purpose provided in Section 1.4.2. The associated hard classifier f is derived by

$$f : \Theta \times \mathcal{X} \to \mathcal{C}, \quad x \mapsto \arg\max_{c \in \mathcal{C}} p_c,$$

while keeping Assumption 0.1 for uniqueness in mind. For instance, the widely used ReLU-based ResNet50 convolutional DNN architecture [34] consist of $K = 52$ layers: 48 of them perform convolutions (c.f. Figure 1.4) using learnable kernels of different sizes to extract features like shapes and patterns, another two execute max- and average-pooling to summarize details and reduce the image resolution in the beginning and the end [48]. Ultimately, this process concludes with a fully connected layer that combines these features to form meaningful structures for classification and the aforementioned softmax layer to output the class label probabilities [48]. Depending on the precise setup, this equates to about $|\Theta| \approx 2.5 \cdot 10^7$ trainable parameters. To learn them and to attack DNNs, we choose the cross-entropy loss (CEL) as our objective function.

1.4.2 Loss Function Choice

A loss function $L : \mathcal{P} \times \mathcal{C} \to \mathbb{R}^+$ takes in a prediction confidence vector $p \in \mathcal{P}$ and the ground-truth label $c \in \mathcal{C}$ to output a penalty depending on the prediction quality. Similar to [16] and [78], we want to motivate the use of the CEL

$$L^{\text{CE}}(F(\theta, x), c) := -\log\left(F(\theta, x)_c\right)$$

as a smooth approximation to the, arguably more expressive, but discontinuous $\{0, 1\}$-loss

$$L^{01}(F(\theta, x), c)) := \mathbb{1}\left[\arg\max_{c' \in \mathcal{C}} F(\theta, x)_{c'} \neq c\right]. \tag{1.1}$$

First, we observe

$$F(\theta, x)_c = \text{softmax}(t, F_{K-1}(\theta, x))_c \to \begin{cases} 1 - L^{01}(F(\theta, x), c)) & \text{if } \ t \to 0, \\ 1/|\mathcal{C}| & \text{if } \ t \to \infty. \end{cases} \tag{1.2}$$

Using (1.1) and the approximate relationship between softmax and arg max from (1.2) for low temperatures $t \in (0, 1]$, we derive

$$\begin{aligned} \mathbb{P}_\delta[f(\theta, x+\delta) = c] &= \mathbb{P}_\delta \left[\arg\max_{c' \in \mathcal{C}} F(\theta, x+\delta)_{c'} = c \right] \\ &= \mathbb{E}_\delta \left[\mathbb{1} \left[\arg\max_{c' \in \mathcal{C}} F(\theta, x+\delta)_{c'} = c \right] \right] \\ &= \mathbb{E}_\delta \left[1 - L^{01}(F(\theta, x+\delta), c)) \right] \\ &\approx \mathbb{E}_\delta \left[F(\theta, x+\delta)_c \right]. \end{aligned}$$

Taking the logarithm on both sides to punish low true-label-confidences over-proportionally hard, we finally arrive at the CEL L^{CE} due to

$$\begin{aligned} \log\left(\mathbb{P}_\delta[f(\theta, x+\delta) = c]\right) &\approx \log\left(\mathbb{E}_\delta \left[F(\theta, x+\delta)_c \right]\right) \\ &\geq \mathbb{E}_\delta \left[\log\left(F(\theta, x+\delta)_c \right) \right] \\ &= -\mathbb{E}_\delta \left[L^{\text{CE}}(F(\theta, x+\delta), c) \right]. \end{aligned}$$

Here, we employed Jensen's inequality and the concavity of the logarithm. Consequently, minimizing the expected CEL increases the lower bound on (the logarithm of) the probability of correct classification. This motivates the incoming use of the CEL in (DRO) and its derived problems as a smooth approximation to the discontinuous $\{0, 1\}$-loss for low temperatures, like the common and here employed $t = 1$. Note that there is another perspective that motivates the CEL as a special case of the Kullback-Leibler divergence (KLD) [80].

1.4.3 Robust Optimization Problems

Using the network and loss modelling from above, we formulate the different robust optimization problems. We start with the standard supervised classification training setup

$$\min_{\theta \in \Theta} R_{\text{CO}}(\theta) := \mathbb{E}_{(X,C) \sim \mathbb{D}} \left[L^{\text{CE}}\left(F(\theta, X), C\right) \right]. \tag{CO}$$

In practice, the data distribution $\mathbb{D}$ of the pair of $(\mathcal{X} \times \mathcal{C})$-valued ground-truth RVs (X, C) in the clean optimization (CO) problem is unknown. It is approximated by the distribution $\widehat{\mathbb{D}}$ within a representative training data set $\{(x_i, c_i) : i \in [N]\} \subseteq \mathcal{X} \times \mathcal{C}$ that ideally can be seen as independent and identically distributed (i.i.d.) realizations of (X, C). Replacing $\mathbb{D}$ by

$$\widehat{\mathbb{D}} := \frac{1}{N} \sum_{i \in [N]} \text{dirac}_{(x_i, c_i)}$$

as in [68] yields the empirical clean optimization problem

$$\min_{\theta \in \Theta} \widehat{R}_{\text{CO}}(\theta) := \frac{1}{N} \sum_{i \in [N]} L^{\text{CE}}\left(F(\theta, x_i), c_i\right). \tag{E-CO}$$

With the same ideas, we now introduce the adversarial case. Consider (CO), but with the image RV X being perturbed by some $\mathbb{R}^n$-valued noise Δ with an unknown distribution $\mathbb{A}$ that the adversary may choose out of the ambiguity set $\mathcal{A} \subseteq \mathcal{RV}(\mathbb{R}^n)$. Then, we obtain the distributionally robust optimization problem

$$\min_{\theta\in\Theta} R_{\text{DRO}}(\theta) := R(\mathbb{D}, \mathcal{A}, \theta) := \mathbb{E}_{(X,C)\sim\mathbb{D}}\left[\sup_{\Delta\sim\mathbb{A}\in\mathcal{A}} \mathbb{E}_{\Delta\sim\mathbb{A}}\left[L^{\text{CE}}\left(F(\theta, \Pi_{\mathcal{X}}^{\infty}(X+\Delta)), C\right)\right]\right]. \tag{DRO}$$

This will be the foundation of our framework, which is visualized in Figure 1.5. Notice that, by the unboundedness of the logarithm in the CEL, the adversary can achieve arbitrarily high loss, making the inner supremum necessary. In contrast, the training problem has an objective value bounded below by zero, justifying it as a minimization problem. In our case of p-norm bounded adversaries with perturbation radius ε_p, the ambiguity set $\mathcal{A}$ only contains distributions with support on $\overline{\mathcal{B}}_p^{\varepsilon_p}$. As $X+\Delta$ may sometimes be outside of $\mathcal{X}$, we clip it via the ∞-norm projection $\Pi_{\mathcal{X}}^{\infty} : \mathbb{R}^n \to \mathcal{X}$. From that point of view, the clean optimization problem (CO) can be interpreted as

$$\min_{\theta\in\Theta} R_{\text{CO}}(\theta) = R(\mathbb{D}, \mathcal{A}_0, \theta), \tag{CO}$$

meaning a minimization in the presence of an incapable adversary with $\mathcal{A}_0 = \{\text{dirac}_0\}$. Similarly, its empirical analogue (E-CO) can be expressed as

$$\min_{\theta\in\Theta} \widehat{R}_{\text{CO}}(\theta) = R(\widehat{\mathbb{D}}, \mathcal{A}_0, \theta). \tag{E-CO}$$

To derive the robust optimization problem that motivates standard adversarial training, consider a maximally dangerous adversary with $\mathcal{A} \supseteq \mathcal{A}_{\text{dirac}} := \{\text{dirac}_x : x \in \mathcal{U}_p^{\varepsilon_p}(x)\}$, enabling to choose $\Delta \sim \text{dirac}_\delta$ for the strongest perturbation $\delta \in \mathcal{U}_p^{\varepsilon_p}(x) := (\overline{\mathcal{B}}_p^{\varepsilon_p}(x) \cap \mathcal{X}) - x$. This transforms (DRO) into

$$\min_{\theta\in\Theta} R_{\text{RO}}(\theta) := R(\mathbb{D}, \mathcal{A}_{\text{dirac}}, \theta) = \mathbb{E}_{(X,C)\sim\mathbb{D}}\left[\sup_{\delta(\theta,X,C)\in\mathcal{U}_p^{\varepsilon_p}(X)} L^{\text{CE}}\left(F(\theta, X+\delta), C\right)\right] \tag{RO}$$

with an inner adversarial problem and an outer training problem, which is usually solved by alternating inner maximization and outer minimization (cf. Section 3.1) based on Danskin's Theorem (Theorem 3.1). As before, it is empirically approximated via sampling by

$$\min_{\theta\in\Theta} \widehat{R}_{\text{RO}}(\theta) := R(\widehat{\mathbb{D}}, \mathcal{A}_{\text{dirac}}, \theta) = \frac{1}{N}\sum_{i\in[N]} \sup_{\delta_i(\theta,x_i,c_i)\in\mathcal{U}_p^{\varepsilon_p}(x_i)} L^{\text{CE}}\left(F(\theta, x_i+\delta_i), c_i\right). \tag{E-RO}$$

For the singleton $\mathcal{A}(x) = \mathcal{A}_{\text{normal}}(x) := \left\{\mathcal{N}(0, \sigma^2 I)|_{\mathcal{U}_p^{\varepsilon_p}}\right\}$, we obtain the stochastic optimization problem

$$\min_{\theta\in\Theta} R_{\text{SO}}(\theta) := R(\mathbb{D}, \mathcal{A}_{\text{normal}}, \theta) := \mathbb{E}_{(X,C)\sim\mathbb{D}}\left[\mathbb{E}_{\Delta\sim\mathcal{N}(0,\sigma^2 I)|_{\mathcal{U}_p^{\varepsilon_p}}}\left[L^{\text{CE}}\left(F(\theta, X+\Delta), C\right)\right]\right] \tag{SO}$$

that motivates randomized smoothing (cf. Section 3.2). It is again approximated in the sense of a Monte Carlo simulation with M realizations by

$$\min_{\theta\in\Theta} \widehat{R}_{\text{SO}}(\theta) := R(\widehat{\mathbb{D}}, A_{\text{normal}}, \theta) = \frac{1}{MN}\sum_{i\in[N]}\sum_{j\in[M]} L^{\text{CE}}\left(F(\theta, x_i+\delta_{ij}), c_i\right) \tag{E-SO}$$

with i.i.d. realizations $(\delta_{ij}) \sim \mathcal{N}(0, \sigma^2 I)|_{\mathcal{U}_p^{\varepsilon_p}}$. Since all of the above problems, including the inner ones on its own, are non-linear and non-convex optimization problems, solving them is NP-hard [56, 84]. Especially, the highly irregular loss landscape and the high dimension complicate finding optimal solutions.

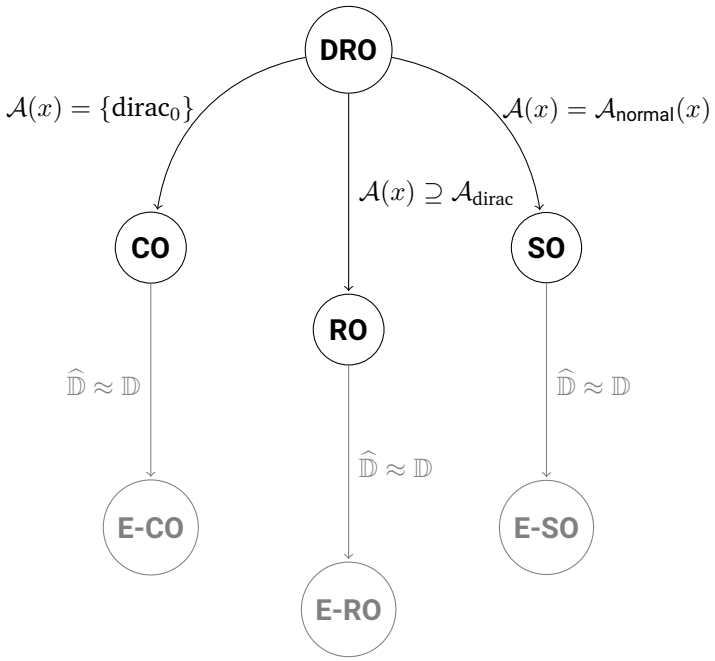


Figure 1.5: Overview of the adversarial training problems from Section 1.4.3.

1.4.4 NP-hardness of Certifying Robustness

Besides that, also certifying the robustness of a ReLU-based DNN is NP-hard. We show this by leveraging the following result.

Theorem 1.4 (Verifying Properties in DNNs with ReLUs is NP-Complete, cf. [35] App. I)**.** *Let $F : \Theta \times \mathbb{R}^n \to \mathcal{P}$ be a soft classifier with ReLU-activations and let φ denote a property that is a conjunction of linear constraints on the inputs $x \in \mathbb{R}^n$ and outputs $p \in \mathcal{P}$ of F, i.e. $\varphi(x, p) := \varphi_1(x) \wedge \varphi_2(p)$. We say that φ is satisfiable on F if there exists an assignment A for x and p such that both*

$$F(\theta, x_A) = p_A \quad \textit{and} \quad \varphi(x_A, p_A)$$

hold true. Then, given F and φ, it is NP-hard to determine if φ is satisfiable on F.

Choosing and modelling the appropriate property φ, we deduce the following statement.

Proposition 1.5 (Certifying robustness is NP-hard)**.** *Consider a soft classifier $F : \Theta \times \mathbb{R}^n \to \mathcal{P}$ and a ground truth pair $(x, c) \in \mathcal{X} \times \mathcal{C}$. Then, it is NP-hard to determine if there exists an adversarial example $\widetilde{x}$, which may be targeted or untargeted and bounded 1- or ∞-norm. In particular, it is NP-hard to find such $\widetilde{x}$, if it exists.*

Proof. The second part of the claim follows from the first since finding an appropriate adversarial example implies its existence. Consequently, using Theorem 1.4, it remains to express the requirements above in terms of linear constraints $\varphi_1(x)$ and $\varphi_2(p)$. Clearly, for ∞-norm bounds, we see

$$\begin{aligned} \widetilde{x} \in \mathcal{U}_\infty^{\varepsilon_\infty} \iff \quad & 0 \leq \widetilde{x}_i \leq 1 \\ & \wedge\ x_i - \varepsilon_\infty \leq \widetilde{x}_i \leq x_i + \varepsilon_\infty \end{aligned}$$

and similarly, for 1-norm bounds we observe

$$\begin{aligned}\widetilde{x} \in \mathcal{U}_1^{\varepsilon_1} \iff \quad & 0 \leq \widetilde{x}_i \leq 1 \\ & \wedge\ x_i - \widetilde{x}_i \leq \nu_i \\ & \wedge\ \widetilde{x}_i - x_i \leq \nu_i \\ & \wedge \sum_{i\in[N]} \nu_i \leq \varepsilon_1.\end{aligned}$$

Remembering $p_c = F(\theta, x)_c$ and

$$f(\theta, x) \in \operatorname*{arg\,max}_{c\in\mathcal{C}} p_c,$$

we can reformulate the demand for targeted miss-classification to a class $\widetilde{c}$ via

$$p_{\widetilde{c}} > p_c \ \forall c \in \mathcal{C} \setminus \{\widetilde{c}\} \implies f(\theta, x) = \widetilde{c}.$$

and

$$p_{\widetilde{c}} \geq p_c \ \forall c \in \mathcal{C} \setminus \{\widetilde{c}\} \impliedby f(\theta, x) = \widetilde{c},$$

respectively. The distinction originates from the undefined behaviour in the unlikely case of a non-unique most probable class label, cf. Assumption 0.1. Keeping this in mind, an untargeted attack is inevitably successful if

$$\exists \lambda \in [0, 1] \quad \text{s.t.} \quad \lambda \geq p_c \ \forall c \in \mathcal{C} \setminus \{\widetilde{c}\} \quad \text{and} \quad \lambda < p_{\widetilde{c}}.$$

Making the second inequality non-strict, yields the corresponding necessary condition. ■

Remark 1.6.

- The statement of Proposition 1.5 may be misleading since we and countless other papers demonstrated that finding adversarial examples is usually tractable for varying image dimensions, DNN architectures and attack taxonomies in the absence of defence mechanisms. Often, this can be done computationally efficient and in a reasonable amount of time using a few gradient ascent steps $\nabla_x L^{\text{CE}}(F(\theta, x), c)$ for untargeted misclassification or descent steps $-\nabla_x L^{\text{CE}}(F(\theta, x), \widetilde{c})$ for targeted misclassification. However, Proposition 1.5 allows that finding adversarial examples can be done quickly sometimes but shows that there is no algorithm that always finds them guaranteed, if existent, while keeping the running time polynomial.
- The inherent difficulty suggested by Proposition 1.5 is still a problem at scale:
  - Despite recent advancements based on mixed-integer linear programming [49], satisfiability modulo theories [35] or randomized smoothing [57, 69, 101, 60, 16, 70, 78, 27, 103], most defenses lack (strong) certified lower robustness bounds. Therefore, trying to compute adversarial perturbations via potent attacks for different ε_p is often the method of choice to bound the robustness from above as tightly as possible [17]. This has to be done at as many points of the empirical data manifold as computationally tractable to overcome the curse of dimensionality associated with high-resolution images [104].
  - In analogy, standard adversarial training methods augment the data set by computing adversarial perturbations for each image in each epoch. This also results in a considerable computational effort.

  This is why, in these two cases, it is common practice to resort to approximate solutions to the inner adversarial problem.

1.5 Contributions

The contributions of this thesis are twofold: On the one hand, theoretical results regarding the following aspects are presented:

- In Proposition 1.5, we leverage a result from [35] to show that executing a successful targeted or untargeted adversarial attack on an ReLU-based DNN by finding an adversarial perturbation is NP-hard.
- We provide a common framework for distributionally robust (DRO), stochastic (SO) and robust optimization (RO) in the context of adversarial deep learning and present their corresponding empirical counterparts, cf. Section 1.4. Each of them is motivated by their underlying assumption on the perturbations that the DNN will face. This framework is especially useful since several papers do not explicitly state the problem, that a specific attack or defense solves. From that perspective, randomized smoothing [57, 69, 101, 60, 16, 70, 78, 27, 103] can be seen as an attempt to solve (SO) and standard adversarial training [3, 80, 95, 43, 28, 77, 82, 104] as an attempt to solve (RO).

On the other hand, we supply new numerical results by answering below questions:

- To ensure visual similarity, adversarial images are commonly given a maximum allowed p-norm distance to the respective clean sample. The most popular choices in the literature are $p \in \{1, 2, \infty\}$. Each of them induces a differently sized and shaped set of allowed perturbations. Clearly, by the well-known inequality $\|\cdot\|_r \leq n^{1/r-1/p}\|\cdot\|_p$ for $r, p \in [1, \infty], r \leq p$, it holds

$$\|\cdot\|_\infty \leq \|\cdot\|_2 \leq \|\cdot\|_1 \tag{1.3}$$

  and

$$\|\cdot\|_1 \leq \sqrt{n}\|\cdot\|_2 \leq n\|\cdot\|_\infty. \tag{1.4}$$

  In the case of images, n is given by the product of the number of pixels per image and the three color channels. For CIFAR-10 with 32×32 pixel images, this equates to $3,072$ and for the compressed 256×256 pixel (and usually 224×224 center-cropped) ImageNet data set to $196,608$ ($150,528$) dimensions. As a consequence, these inequalities are weak for $(1, \ldots, 1)^\top$ despite being tight for unit vectors. Hence, (1.3) and (1.4) only provide bounds on the radius for the p-norm balls to be nested. These values span a broad interval of radii in between them on how to choose ε_p for the attack strengths to be comparable. We propose a transformation that recognizes the p-norm ball's Lebesgue-volume as the main criterion for scaling ε_p and show that both scaling functions induce similarity between attacks on ImageNet to different degrees, cf. Contribution 4.2.
- In the process of validating this contribution on ImageNet, we proposed two projected gradient descent (PGD) attacks with harmonically and geometrically decaying step lengths. They showed no downsides and some upsides in our analysis when compared with constant step sizes from the literature. Most notably, the harmonically decreasing step lengths enabled finding adversarial examples with unprecedented zero true-label-confidences, cf. Contribution 4.1.
- Adversarial training relies on computing adversarial examples at training-time for the model to learn on. Finding those adversarial inputs is generally accomplished by a gradient ascent scheme in the input space, bounded by a p-norm ball around the natural query. Apart from this, a generic stochastic gradient descent scheme via backpropagation in the space of model parameters is performed, just like in the non-adversarial setting. From a data science perspective, augmenting the training data by the

type of adversarial inputs you want the model to classify correctly, seems reasonable. In fact, most papers on this topic [3, 80, 95, 43, 28, 77, 82, 104] take that perspective and do not legitimize the approach any further whilst presenting state-of-the-art robustness results. A mathematical motivation, only mentioned in [52] of all the incorporated data science papers here, comes from Danskin's Theorem (Theorem 3.1). However, numerous assumptions of it are violated in the context of supervised image classification so that it is not self-evident that this approach works in practice. To this end, we provide numerical results that help to explain the success of such ascending-descending algorithms for robust image classification training despite these infringements. Using these results, we suggest an answer to the questions raised in [80, 95] about when strong adversaries are necessary for training, when weak ones suffice and how they induce robustness: The accuracy, to which the adversarial problem is solved, should be decreasing over the course of the training to reduce costs and the robustness comes from Danskin's Theorem being in some sense insensitive to assumption violations in practice. This training idea remains to be tested in a future work, cf. Contribution 4.3.

- BOBYQA-based black-box attacks have been considered in two papers [92, 93]. There, it has been tested against naturally and adversarially trained DNNs on MNIST, CIFAR-10 and ImageNet. The defenses included defensive distillation [63, 61, 62] for MNIST and CIFAR-10 as well as single- and multi-step gradient adversarial training [23] for CIFAR-10 and ImageNet. However, recent findings [10, 9] suggest that defensive distillation proves ineffective against 2-norm Carlini-Wagner (C&W) attacks and the attacked gradient-based pretrained DNNs from [23, 52] significantly relied on the so-called gradient masking effect [52, 15, 6, 57, 101, 105, 3, 89, 90, 104]. Therefore, we give an update to the reported results in [92, 93] by confronting state-of-the-art defenses with the BOBYQA-based black-box attack. Concretely, we attack DNNs from [95] and [78], which are trained via a randomly initialized FGSM (R-FGSM) to prevent gradient masking and a randomized smoothing approach, respectively. In conclusion, the BOBYQA-based attack looses much of its effectiveness when facing these defenses on CIFAR-10 but should be added to and remain in the portfolio of test attacks conducted to defenses for various reasons, cf. Contribution 4.4.

2 Adversarial Problem and Attacks

In this chapter, we focus on the problem of creating adversarial examples. Specifically, we present selected methods within two branches of attacks: First, we give an overview of commonly applied gradient-based methods in a white-box setting [42, 52, 10, 101, 105, 37, 100, 3, 80, 89, 60, 16, 95, 30, 43, 82, 104] and motivate the randomization of some of them via phenomenons called gradient masking [6], gradient obfuscation [3] and catastrophic adversarial overfitting [95]. Then, we give a brief introduction to zeroth-order adversarial attacks with a focus on the BOBYQA-based targeted black-box attack from [92, 93].

2.1 First-Order White-Box Attacks

First-order white-box attacks are optimization methods to create targeted or untargeted adversarial examples. The term 'white-box' refers to the assumption that the DNN is fully known to the attacker from the architecture down to every single weight. The attacks are of 'first-order' as they only perform first derivative function evaluations, i.e. only gradients and no Hessians are computed. There is a large collection of such attacks, however, there are two kinds of attacks that stand out in terms of the attention they receive in the literature. They differ in the way they incorporate the goals of similarity and misclassification:

- The first category minimizes the distortion under the constraint of misclassification: In 2014, Szegedy et al. [86] proposed the L-BFGS attack that solves the problem

$$\min_{\widetilde{x}\in\mathcal{X}} \gamma\|x-\widetilde{x}\|_2^2 + L^{\text{CE}}(F(\theta,\widetilde{x}),\widetilde{c})$$

  targeted to a class $\widetilde{c}$ as a relaxed proxy for the more desirable but ill-constrained problem

$$\min_{\widetilde{x}\in\mathcal{X}} \|x-\widetilde{x}\|_2^2 \quad \text{s.t.} \quad f(\widetilde{x})=\widetilde{c},$$

  which minimizes the 2-norm distortion under the constraint of a targeted misclassification. The loss term in the relaxed problem removes the constraint by incentivizing high prediction confidences of the target label $\widetilde{c}$. On a similar note, in 2017, Carlini et al. [10] proposed three attacks, called C&W ℓ_p-attacks for each $p\in\{0,1,2\}$, that also take the similarity goal into account by adding a penalty term. The $p=2$ version has experienced the largest recognition so far [10, 82, 104]. There, they also removed the box-constraint $\widetilde{x}\in\mathcal{X}$ via a component-wise hyperbolic tangent transformation $\widetilde{x}=(\tanh(w)+1)/2$ and incorporated the targeted misclassification goal by maximizing the gap between the target class' logit and the other logits. More precisely, they proposed solving

$$\min_{w\in\mathbb{R}^n} \left\|\frac{1}{2}(\tanh(w)+1)-x\right\|_2^2 + \gamma Z\left(\frac{1}{2}(\tanh(w)+1)\right),$$
$$Z(y) := \max\{F_{K-1}(y)_c : c\neq\widetilde{c}\} - F_{K-1}(y)_{\widetilde{c}}.$$

Later, the C&W ℓ_2-attack was considered in a randomized version [104] and embedded into a class of more general class of elastic net attacks to DNNss (EADs) [14, 82].

- We consider the category of attacks that bound the distortion and optimize the misclassification in the form of the prediction confidence or loss. Explicitly, we express the misclassification goal as the CEL and incorporate the similarity goal as a p-norm constraint. We focus on the inner problem

$$\sup_{\delta \in \mathcal{U}_p^{\varepsilon_p}(x)} L^{\text{CE}}\left(F(\theta, x+\delta), c\right) \tag{ADV-UNT}$$

of the robust optimization problem (RO) for untargeted misclassification and

$$\min_{\delta \in \mathcal{U}_p^{\varepsilon_p}} L^{\text{CE}}\left(F(\theta, x+\delta), \widetilde{c}\right) \tag{ADV-TAR}$$

for targeted attacks. Common attempts to approximate solutions to these problems are clustered within the template of the next section.

2.1.1 Attack Template

Within the framework of Algorithm 1, we can express many of the p-norm bounded white-box attacks from the literature.

Algorithm 1 – FO-WB-Attack

Input: image $x \in \mathcal{X}$, true label $\widetilde{c} \in C$, target label $c \in \mathcal{C} \cup \{\textit{none}\}$, norm exponent $p \geq 1$, adversarial radius $\varepsilon_p > 0$, soft classifier $F : \Theta \times \mathcal{X} \to \mathcal{P}$, number of adversarial steps $S \in \mathbb{N}$, adversarial step length $(\omega_s)_{s \in [S]} \subseteq \mathbb{R}^+$, adversarial randomization $R \in \mathcal{RV}(\mathbb{R}^n)$

Output: adversarial example $\widetilde{x} \in x + \mathcal{U}_p^{\varepsilon_p}(x)$

random offset
$\widetilde{x} := \Pi^p_{\mathcal{U}_p^{\varepsilon_p}}(x + R)$
for $s = 1, \ldots, S$ **do**
    # gradient computation
    if $\widetilde{c} = \textit{none}$ **then**
        $g_s := \nabla_x L^{\text{CE}}(F(\theta, \widetilde{x}), c)$
    else
        $g_s := -\nabla_x L^{\text{CE}}(F(\theta, \widetilde{x}), \widetilde{c})$
    end if
    # gradient normalization
    if $p = \infty$ **then**
        $g_s := \text{sign}(g)$
    else
        $g_s := g_s / \|g_s\|_p$
    end if
    # update
    $\widetilde{x} := \Pi^p_{\mathcal{U}_p^{\varepsilon_p}}\left(\widetilde{x} + \omega_s g_s\right)$
end for

In fact, the popular family of FGSM and PGD attacks with their randomized counterparts R-FGSM and R-PGD can be categorized as shown in Table 2.1.

Name	p	ω_s	S	R	$\widetilde{c}$	Examples
Untargeted FGSM	∞	ε	1	0	none	[30, 80, 101, 95, 60, 80]
Targeted FGSM	∞	ε	1	0	chosen $\widetilde{c}$	[42, 43]
Least-Likely FGSM	∞	ε	1	0	$\widetilde{c} = \arg\min_{c' \in \mathcal{C}} p_{c'}$	[89]
Untargeted R-FGSM	∞	ε	1	$\mathcal{U}\left[\overline{\mathcal{B}}_p^{\varepsilon_p}\right]$	none	[95]
Targeted R-FGSM	∞	$\varepsilon - \sigma$	1	$\mathcal{N}(0, \sigma^2)$	none	[89]
Least-Likely R-FGSM	∞	$\varepsilon - \sigma$	1	$\mathcal{N}(0, \sigma^2)$	none	[89]
PGD Single-Step	1	ε	1	0	none	[80]
	2	ε	1	0	none	[101, 80]
Untargeted PGD Multi-Step	∞	constant	> 1	0	none	[52, 23]
Targeted PGD Multi-Step	∞	$1/255$	> 1	0	chosen $\widetilde{c}$	[42, 43]
Least-Likely PGD Multi-Step	∞	$\geq \varepsilon/S$	> 1	0	$\widetilde{c} = \arg\min_{c' \in \mathcal{C}} p_{c'}$	[89]
R-PGD Multi-Step	∞	constant	> 1	$\mathcal{U}\left[\overline{\mathcal{B}}_p^{\varepsilon_p}\right]$	none	[52]

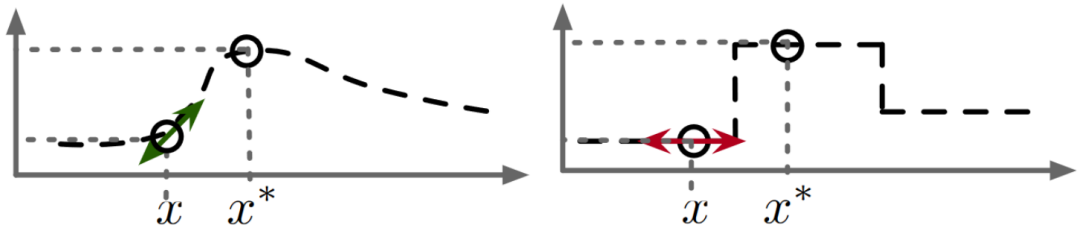
Table 2.1: p-Norm bounded first-order attacks categorized as variants of Algorithm 1.

As we will see in Section 3.1, these attacks also play a crucial role in adversarial training. There, they are utilized to compute adversarial examples for the DNN to learn on.

2.1.2 Randomization and a False Sense of Security

To motivate the use of the randomization term $R \in \mathcal{RV}(\mathbb{R}^n)$ in Algorithm 1, we give a short introduction to three phenomenons called gradient masking [6, 65], gradient obfuscation [3, 89, 105] and catastrophic adversarial overfitting [95, 65].

The term 'gradient masking', introduced in [6], describes the behaviour of DNNs during adversarial training where they are incentivized to flatten their loss landscape $x \mapsto L^{\text{CE}}(F(\theta, x), c)$ to minimize gradient lengths. One the one hand, this is desirable from a defensive point of view as gradient lengths become less expressive for the steepness of the local loss landscape. One the other hand, it rarely removes large fluctuations that can still be exploited by adversaries, cf. Figure 2.1.

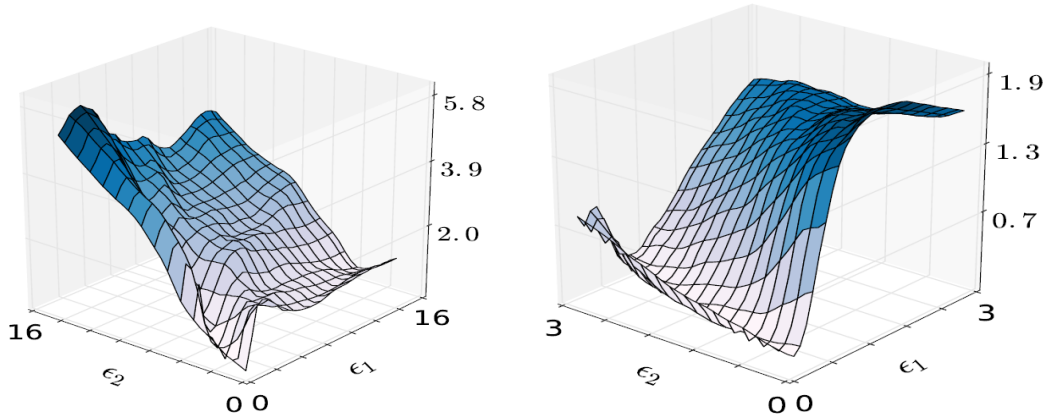


(a) natural (b) defended

Figure 2.1: Loss landscape in the case of masked gradients, with (b) adversarially trained to mask its gradients and (a) being a naturally trained (substitute) model. Image from [65].

However, Figure 2.1 highlights another way in which this poses a challenge to the attacker: The gradient's direction becomes a less meaningful indicator of the local loss landscape. This observation was coined

'gradient obfuscation' in [3]. More generally, it describes the introduction of small artefacts in the loss landscape around the training images to misguide the attacker into directions that do not increase the loss when making a step of significant length, cf. Figure 2.2.



(a) large ϵ_1 and ϵ_2 (b) small ϵ_1 and ϵ_2

Figure 2.2: Loss landscape in the case of obfuscated gradients, with (b) zoomed in on (a). Here, ϵ_1 is the coordinate into the direction $g := \nabla_x L^{\mathsf{CE}}(F(\theta, x), c)$ and ϵ_2 into a direction orthogonal to g. Note that g, pointing towards ϵ_1, is only a good local approximation in (b) but not in (a), hence misguiding attackers. Image from [89].

However, that confusion only applies to adversaries that take gradient information precisely and exclusively in the clean sample x. Beyond that, performing adversarial training exclusively with single-step methods like FGSM can cause 'catastrophic adversarial overfitting', which was discovered in [95]. It refers to a distortion of the naturally trained decision boundaries in a way that makes the DNN even more susceptible to black-box, multi-step or randomized attacks, cf. Figure 2.3. In fact, the DNN might even be more precise on FGSM perturbed images than clean ones after training due to 'label leaking' [43].

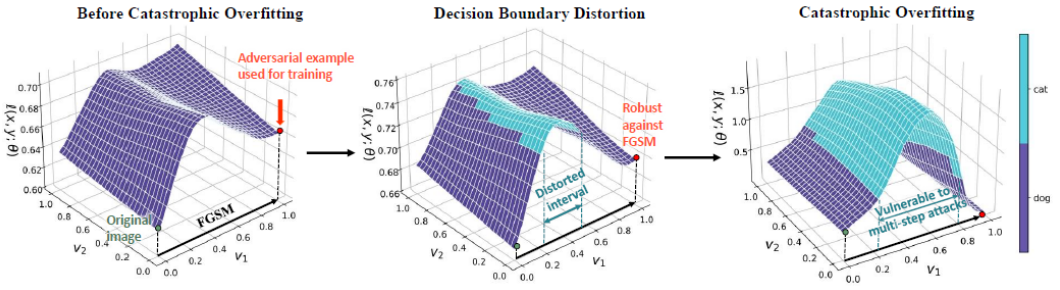


Figure 2.3: Training timeline of a normal decision boundary evolving into a distorted decision boundary. (Left) shows the loss surface before catastrophic overfitting with a FGSM adversarial direction $v_1 := \nabla_x L^{\mathsf{CE}}(F(\theta, x), c)$ and a random direction v_2. The red point marks an adversarial example $x + v_1$ generated from x with label 'dog'. (Middle) displays the loss surface after learning with $x + v_1$. The distortion starts occurring. (Right) demonstrates that as training continues, the decision boundary distortion grows uncontrollably such that overall local robustness decreases. Image from [37].

Therefore, gradient obfuscation and catastrophic overfitting motivate to first perform a slight random perturbation R to get stronger attacks, both from an offensive as well as a defending perspective. One method to test a DNN for all three concepts involves performing white- and black-box attacks [105] to expose a possible 'false sense of security' [3] provided by them. Since white-box adversaries have more information about the victim, they should perform superior. But if these phenomenons constitute the main line of defense, then the transferred or model-based black-box attacks outperform the white-box attacks due to being unfazed by small or misleading gradients and distorted decision boundaries in the original model. This holds by design of these attacks: Transferred attacks use a naturally trained substitute model to compute the adversarial example via first-order methods and model-based attacks only use point evaluations of the defended model. We will learn more on that in the next section.

2.2 Zeroth-Order Black-Box Attacks

This section gives a concise introduction to black-box attacks against DNNs [15, 6, 92, 93, 57, 105, 100, 3, 89, 82] with main emphasis on the BOBYQA-based targeted attack [92, 93] that will be run in the numerical tests of Section 4.4. Black-box attackers, as opposed to white-box attackers, posses no knowledge about the victim DNN and only the ability to receive the prediction confidence vector $(p_c)_{c\in\mathcal{C}} = F(\theta, x)$ of samples they query. This is a weaker attacker model compared to white-box adversaries, who can obtain gradients. Nevertheless, it is an important attacker model for at least two reasons: They represent a sanity check for the aforementioned 'false sense of of security' [3] given by the phenomenons explained in Section 2.1.2. Furthermore, the black-box setting is arguably the more realistic one in the real-world, e.g. when attacking an online classification tool like `clarifai.com` from an user standpoint like [6] did.

2.2.1 Two Common Approaches

Black-box attacks can be categorized into two disjoint groups with distinct approaches:

- Attacks based on transferability between different DNNs create an adversarial example on a naturally trained substitute classifier to which they have white-box access. Consequently, this allows the application of first-order methods on that proxy model. Transferability, i.e. the tendency that adversarial samples of an originally attacked DNN also confuse other DNNs, was discovered simultaneously with the phenomenon of adversarial examples for image classification in 2014 by Szegedy et al. [86]. Subsequently, it was reproduced and further investigated in countless other papers [52, 15, 6, 10, 57, 105, 100, 3, 89, 30, 90, 43, 82, 47].
- Model-based attacks, like ZOO [15], TR-Attack [99], BOBYQA [92, 93] and others [93], rely on approximating properties of the input's loss landscape $x \mapsto L^{\mathrm{CE}}(F(\theta, x), c)$ solely via the zeroth-order information gained from repeated querying. Most notably, this involves estimating first and second derivatives of the loss function via finite differences [15, 6, 99] or developing a local interpolation model of the loss [92, 93].

Up to now, transferred black-box attacks are more prominent in the literature. However, there are arguments to be made for (a) model-based, (b) targeted black-box attacks to receive more attention:

(a) As demonstrated in [92] and [43], increasing the number of gradient steps in white-box attacks increases the attacker's success, but it decreases transferability to other DNNs. Hence, assuming modern DNNs become increasingly robust to single-step attacks, then attackers might not be successful at performing transferred black-box attacks at all.

(b) Similarly, [47] showed that targeted attack transfer less than untargeted ones. Subsequently, taking a closer look at model-based targeted black-box attacks, like BOBYQA, seems appropriate.

Beyond that, a further investigation is overdue when considering that [92] and [93] are the only papers investigating BOBYQA and showing its competitiveness relative to other black-box attacks. In general, BOBYQA is a state-of-the art zeroth-order optimizer [12, 13] which finds application in other domains where the number of queries is to be minimized, e.g. climate modelling [87]. Hence, we will examine the methodology of model-based targeted BOBYQA black-box attacks below.

2.2.2 The BOBYQA Algorithm

We start by summarizing the main ideas of the BOBYQA method [72, 73, 71] before delving into the implementation for creating adversarial examples [13, 92, 93]. First, consider the targeted misclassification problem (ADV-TAR) for $p = \infty$, i.e.

$$\min_{\delta \in \mathcal{U}_\infty^{\varepsilon_\infty}} L^{\text{CE}}\left(F(\theta, x+\delta), \widetilde{c}\right). \tag{ADV-TAR}$$

For notational simplicity, we omit the fixed parameters θ and x to define $\psi_{\widetilde{c}}(\cdot) := L^{\text{CE}}(F(\theta, x+\cdot), \widetilde{c})$. Additionally, we resolve the notation of the adversarially exploitable neighbourhood to reformulate (ADV-TAR) as box-constrained nonlinear program

$$\min_{\delta \in \mathbb{R}^n} \psi_{\widetilde{c}}(\delta) \quad \text{s.t.} \quad \|\delta\|_\infty \leq \varepsilon_\infty \text{ and } x+\delta \in [0,1]^n.$$

To derive the problem BOBYQA will solve, we note that [92, 93] minimize a related objective

$$\Psi_{\widetilde{c}}(\delta) := \psi_{\widetilde{c}}(\delta) - \sum_{c \neq \widetilde{c}} \psi_c(\delta),$$

instead. However, both functions are closely related and induce the same ordering on $\delta \in \mathbb{R}^n$ as they have the same level sets

$$\begin{aligned}
& \psi_{\widetilde{c}}(\delta_1) \leq \psi_{\widetilde{c}}(\delta_2) \\
\iff \quad & -\log(F(\theta, \widetilde{x}_1)_{\widetilde{c}}) \leq -\log(F(\theta, \widetilde{x}_2)_{\widetilde{c}}) \\
\iff \quad & F(\theta, \widetilde{x}_1)_{\widetilde{c}} \geq F(\theta, \widetilde{x}_2)_{\widetilde{c}} \\
\iff \quad & \frac{1}{F(\theta, \widetilde{x}_1)_{\widetilde{c}}} \leq \frac{1}{F(\theta, \widetilde{x}_2)_{\widetilde{c}}} \\
\iff \quad & \frac{1 - F(\theta, \widetilde{x}_1)_{\widetilde{c}}}{F(\theta, \widetilde{x}_1)_{\widetilde{c}}} \leq \frac{1 - F(\theta, \widetilde{x}_2)_{\widetilde{c}}}{F(\theta, \widetilde{x}_2)_{\widetilde{c}}} \\
\iff \quad & \exp\left(\log\left(\frac{1 - F(\theta, \widetilde{x}_1)_{\widetilde{c}}}{F(\theta, \widetilde{x}_1)_{\widetilde{c}}}\right)\right) \leq \exp\left(\log\left(\frac{1 - F(\theta, \widetilde{x}_2)_{\widetilde{c}}}{F(\theta, \widetilde{x}_2)_{\widetilde{c}}}\right)\right) \\
\iff \quad & \exp\left(\log(1 - F(\theta, \widetilde{x}_1)_{\widetilde{c}}) - \log(F(\theta, \widetilde{x}_1)_{\widetilde{c}})\right) \leq \exp\left(\log(1 - F(\theta, \widetilde{x}_2)_{\widetilde{c}}) - \log(F(\theta, \widetilde{x}_2)_{\widetilde{c}})\right) \\
\iff \quad & \exp(\Psi_{\widetilde{c}}(\delta_1)) \leq \exp(\Psi_{\widetilde{c}}(\delta_2)) \\
\iff \quad & \Psi_{\widetilde{c}}(\delta_1) \leq \Psi_{\widetilde{c}}(\delta_2).
\end{aligned}$$

We adopt this objective and w.l.o.g. suppose some fixed target class $\widetilde{c}$ to define $\Psi := \Psi_{\widetilde{c}}$ and formulate the problem to be solved

$$\min_{\delta \in \mathbb{R}^n} \Psi(\delta) \quad \text{s.t.} \quad \|\delta\|_\infty \leq \varepsilon_\infty \text{ and } x + \delta \in [0,1]^n. \tag{2.1}$$

BOBYQA approaches this problem by setting up and iteratively improving a quadratic interpolation model

$$Q_k(\delta) := c_k + g_k^\top \delta + \delta^\top H_k \delta$$

with symmetric Hessian H_k that satisfies the interpolation conditions

$$Q_k(\delta) = \Psi(\delta) \quad \forall \delta \in \mathcal{I}_k := \{\delta_1^k, \ldots, \delta_d^k\} \tag{INT-CND}$$

in each iteration k. The set of interpolation points $\mathcal{I}_k$ has cardinality $d \in \{n+1, \ldots, n(n+1)/2 + n + 1\}$ to allow for anything from a linear estimation with $d = n+1$ and $H_k = 0$ up to Q_k being uniquely determined by (INT-CND) for $d = n(n+1)/2 + n + 1$. To improve the model, we approximately solve the trust region problem

$$\min_{\delta \in \mathbb{R}^n} Q_k(\delta) \quad \text{s.t.} \quad \|\delta\|_2 \leq r_k, \ \|\delta\|_\infty \leq \varepsilon_\infty \text{ and } x + \delta \in [0,1]^n \tag{TRP}$$

via first-order information $\nabla_\delta Q_k(\delta^k_{i_{\text{best}}})$ in the best solution seen so far

$$\delta^k_{i_{\text{best}}} \in \operatorname*{arg\,min}_{\delta \in \mathcal{I}_k}(\Psi(\delta)).$$

Subsequently, we use the obtained approximate minimizer δ_*^k to replace a specifically chosen point $\delta^k_{i_{\text{out}}}$ in $\mathcal{I}_k$ to derive $\mathcal{I}_{k+1}$. That means concretely

$$\delta_i^{k+1} := \begin{cases} \delta_i^k & \text{if} \quad i \neq i_{\text{out}}, \\ \delta_*^k & \text{if} \quad i = i_{\text{out}} \end{cases} \quad \forall i \in [d].$$

Note that in some cases BOBYQA does not update $\mathcal{I}_{k+1}$ according to (TRP) but performs an alternative iteration called RESCUE to promote good linear independence between the interpolation points. Moreover, the choice of the radius r_k and leaving index i_{out} takes multiple criteria like roundoff errors and interpolation point distance into account to ensure numerical stability. We refer interested readers to [72] for a comprehensive presentation. To obtain the new quadratic model Q_{k+1}, we do not solve (INT-CND) from scratch in each iteration. Instead, we compute an update model

$$Q_U(\delta) := Q_{k+1}(\delta) - Q_k(\delta) =: c_U + g_U^\top \delta + \delta^\top H_U \delta$$

via the update interpolation conditions

$$Q_U(\delta_i^{k+1}) = \begin{cases} 0 & \text{if} \quad i \neq i_{\text{out}}, \\ \Psi(\delta_i^{k+1}) - Q_k(\delta_i^{k+1}) & \text{if} \quad i = i_{\text{out}}, \end{cases} \quad \forall i \in [d] \tag{INT-UPD}$$

and add Q_U to Q_k once calculated. Remembering that $n \approx 3 \cdot 10^3$ for CIFAR-10 and $n \approx 2 \cdot 10^5$ for ImageNet, it is generally reasonable to work with $d \ll n(n+1)/2 + n + 1$. Then, (INT-UPD) is underdetermined, i.e. there are unused degrees of freedom. BOBYQA resolves this lack of uniqueness by incentivizing the Hessian to change as little as possible from iteration k to $k+1$ w.r.t. the Frobenius norm

$$\|H\|_F := \sqrt{\sum_{i \in [n]} \sum_{j \in [n]} H_{ij}^2}$$

in the so-called least Frobenius norm update problem

$$\min_{c_U, g_U, H_U} \|H_U\|_F \quad \text{s.t.} \quad \text{(INT-UPD)}. \tag{LFNU}$$

If $\mathcal{I}_k$ is chosen appropriately [73], then (LFNU) is a strictly convex minimization problem with its unique optimizer given by the solution of the $(d+1+n) \times (d+1+n)$ linear system of equations

$$\left(\begin{array}{c|cc} A & \mathbf{1}^\top & D^\top \\ \hline \mathbf{1} & 0 & 0 \\ D & 0 & 0 \end{array}\right) \cdot \left(\begin{array}{c} \xi \\ \hline c_U \\ g_U \end{array}\right) = \left(\begin{array}{c} P \\ \hline 0 \\ 0 \end{array}\right) \tag{OPT-CND}$$

with $(A_{ij})_{ij} = ({\delta_i^{k+1}}^\top \delta_j^{k+1}/2)_{ij} \in \mathbb{R}^{d \times d}$, $D := \left(\delta_1^{k+1} | \dots | \delta_d^{k+1}\right) \in \mathbb{R}^{n \times d}$, $\mathbf{1} := (1, \dots, 1) \in \mathbb{R}^{1 \times d}$ and $(P_i)_i \in \mathbb{R}^d$ given by the right-hand side of (INT-UPD). Its solution yields the coefficients for Q_U, where the Hessian is given by

$$H_U := \sum_{i \in [d]} \xi_i \cdot \delta_i^{k+1} {\delta_i^{k+1}}^\top.$$

Again, rather than solving (OPT-CND) in each iteration separately, it is beneficial in terms of runtime and roundoff errors to compute the inverse of the system matrix once and update it each time one interpolation point is substituted. Choosing the first set of interpolation points $\mathcal{I}_0$ appropriately, allows for an efficient computation of the initial inverse. For details, we refer to [73, 71]. In conclusion, the model for the next iteration is given by

$$Q_{k+1}(\delta) = Q_k(\delta) + Q_U(\delta) = c_k + c_U + (g_k + g_U)^\top \delta + \delta^\top (H_k + H_U) \delta$$

and satisfies (INT-CND) by construction. This process terminates if $\delta_{i_{\text{best}}}^k$ is a satisfactory solution to (2.1). In the context of adversarial examples, this is the case when

$$f\left(\theta, x + \delta_{i_{\text{best}}}^k\right) = \widetilde{c}.$$

Otherwise, the algorithm stops when the maximum number of iterations or - as in our case - the maximum number of $3,000$ function evaluations is reached. The general BOBYQA algorithm is implemented in the `Py-BOBYQA` package [13] for `Python`. This implementation is the backbone of the code in [92, 93] which we modified in order to attack modern adversarial defenses.

Staying consistent with [92, 93], we left all major parameter unchanged, e.g. $d = n + 1$ with $H = 0$. This allows the validation of our results on BOBYQA against the natural and MadryLab defense [23]. Additionally, our results against R-FGSM adversarial training and randomized smoothing can be seen as a direct update as we include two additional contemporary defenses.

To further reduce the number of optimization dimensions and minimize the number of queries per attack, [92, 93] employ a strategy called 'hierarchical lifting' which originates from [31] and has previously been leveraged to adversarial attacks in [15]. The main idea is to compute the perturbation δ within a sequence of growing lower-dimensional subspaces $\left(\mathbb{R}^{b_l}\right)_{l \geq 0}$ on different levels $l \geq 0$, instead of on $\mathbb{R}^n$ directly. These subspaces consist of b_l blocks of pixels that form a grid for each color channel on the image. For each block, one virtual pixel is determined as a representative. These pixels form a vector $\hat{\eta}_l \in R^{b_l}$ that is optimized with the BOBYQA algorithm. Subsequently, these values are mapped back onto the image via the selection matrix $S_l \in \{0,1\}^{n \times b_l}$ before being interpolated to the actual pixels of their respective blocks via an interpolation matrix $L_l \in \{0,1\}^{n \times n}$ in a piece-wise constant manner, cf. Figure 2.4.

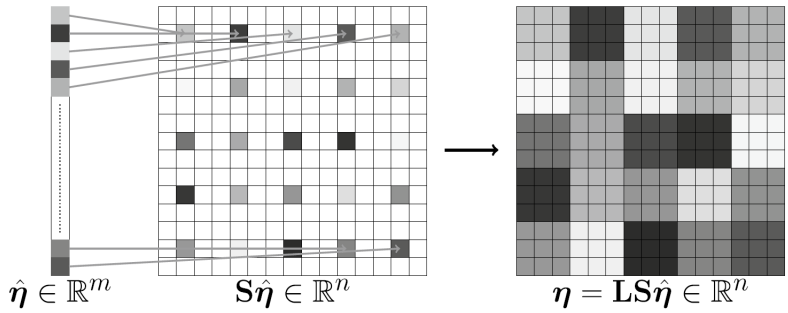


Figure 2.4: BOBYQA block-lifting with piece-wise constant interpolation of a sparse grid onto the whole image: Sorting matrix S assigns optimization variables $\hat{\eta}$ onto a sparse grid of pixels that get optimized by the BOBYQA approach. Finally, the surrounding pixels are perturbed according to their closest neighbours via a lifting matrix L. Image from [93].

This process is applied recursively to each of the resulting blocks on the next level $l+1$, yielding a tree of optimization problems with solutions $(\hat{\eta}_l)_{l\geq 0}$ where each node has $b_{l+1} = 4b_l$ children. The perturbations of their parents are inherited and considered as fixed. Initially, the image is divided into $b_0 = 2 \cdot 2 \cdot 3$ problems, i.e. four image quadrants per color channel. Each level of the tree is traversed in an order that tries to manipulate more influential blocks of pixels first to increase the chance of a premature termination. To this end, the blocks on each level are sorted by the variance of mean intensity among the neighboring blocks to identify regions of high contrast like contours and patterns. The result is a sequence of perturbations

$$\eta_l := L_l S_l \hat{\eta}_l = L_l S_l \sum_{i \in [b_l^j]} \hat{\eta}_l^i \in \mathbb{R}^n \qquad \forall l \in \{0, \ldots, \lfloor \log_4(n/3) \rfloor - 1\}$$

that add up to the perturbation δ. Hence, in the case of $32 \times 32 \times 3$ CIFAR-10 images, there are no more than four levels of problems. This strategy is called hierarchical block-lifting and visualized in Figure 2.5.

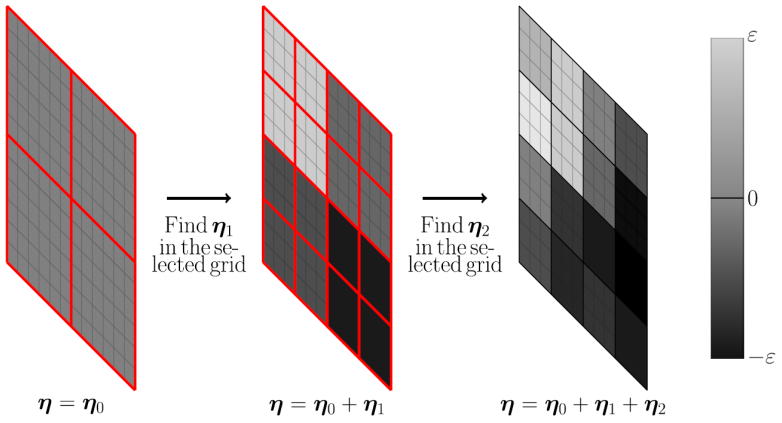
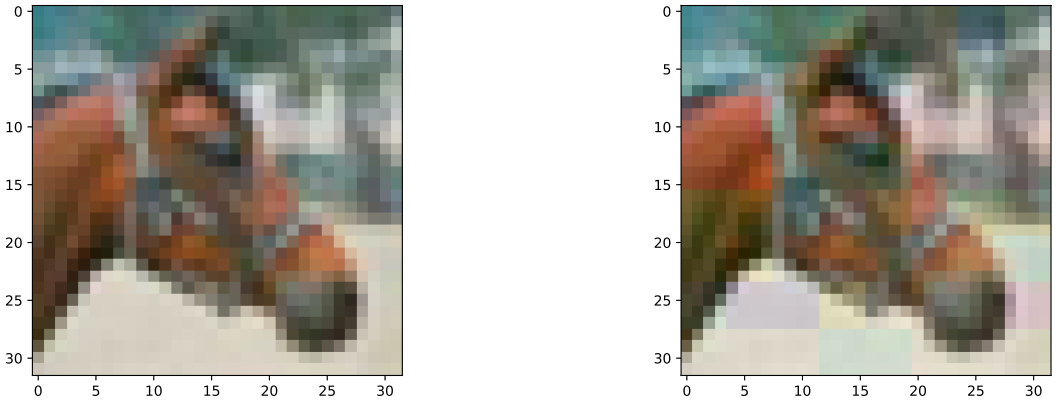


Figure 2.5: BOBYQA hierarchical block-lifting: Recursive application of the block-lifting to the lower-dimensional subproblems yields a tree of optimization problems to be solved by the BOBYQA algorithm. The subproblems are traversed in order of decreasing variance of mean intensity among adjacent blocks. Image from [93].



(a) clean, $\varepsilon_\infty = 0$ (b) adversarial, $\varepsilon_\infty = 9/255$

Figure 2.6: Picture of a horse (label 7) from the CIFAR-10 test data set. Clean image (a) correctly classified by adversarially trained LocusLab [95] DNN and noticeably perturbed sample (b) classified as target label 'cat' (label 3). High $\varepsilon_\infty \approx 0.04$ allows to see the applied block-lifting.

Figure 2.6 presents a successful targeted attack that was computed with this method. The high perturbation radius $\varepsilon_\infty = 9/255$ accentuates the hierarchical block-wise perturbations. In contrast, Figure 1.2 shows another targeted attack but with $\varepsilon_\infty = 3/255$ and no obvious signs of the BOBYQA attack.

Having understood the concept of such attacks, we again take the dimensionality reduction one step further: Since b_l grows exponentially in l, hierarchical lifting is combined with another clustering technique, referred to as domain sub-sampling. It relies on partitioning the set of blocks $[b_l]$ into disjoint subsets of size k_l which are characterized by matrices $\Omega_j \in \{0,1\}^{b_l \times k_l}$. The meta-variables optimized on these clusters are denoted by $\tilde{\eta}_l^j \in \mathbb{R}^{k_l}$ for $j \in \lceil b_l/k_l \rceil$, yielding

$$\hat{\eta}_l = \sum_{j=1}^{\lceil b_l/k_l \rceil} \Omega_j \tilde{\eta}_l^j \in \mathbb{R}^{b_l} \qquad \forall l \in \{0, \dots, \lfloor \log_4(n/3) \rfloor - 1\}.$$

To summarize, at each level l and for each cluster j on that level, the adversary improves the existing solution to (2.1) by computing a higher resolution perturbation δ via solving

$$\begin{aligned} \min_{\tilde{\eta}_l^j \in \mathbb{R}^{k_l}} \quad & \Psi\left(x + \sum_{l'<l} \eta_{l'} + L_l S_l \left(\sum_{j'<j} \hat{\eta}_l^{j'} + \Omega_j \tilde{\eta}_l^j\right)\right) \\ \text{s.t.} \quad & \left\| \sum_{l'<l} \eta_{l'} + L_l S_l \left(\sum_{j'<j} \hat{\eta}_l^{j'} + \Omega_j \tilde{\eta}_l^j\right) \right\|_\infty \leq \varepsilon_\infty \\ & x + \sum_{l'<l} \eta_{l'} + L_l S_l \left(\sum_{j'<j} \hat{\eta}_l^{j'} + \Omega_j \tilde{\eta}_l^j\right) \in [0,1]^n \end{aligned}$$

with the BOBYQA algorithm. We refer to [92, 93] for additional information on the quantity and size of these clusters as well as the assignment of blocks to clusters. We adopted the default setting from the accompanying `GitHub` repository of [93] as these settings proved successful for them and us.

3 Robustification Problem and Methods

The previous chapter introduced the creation of adversarial examples mainly from an attacker's point of view. This chapter presents two ways in which they are leveraged to rubistify DNNs as a defender. We focus on two of the most promising robustification methods, namely standard adversarial training with first-order adversary [3, 80, 95, 43, 28, 77, 82, 104] in Section 3.1 and randomized smoothing with Gaussian noise [57, 69, 101, 60, 16, 70, 78, 27, 103] in Section 3.2. Both defenses, and in fact many more from the non-detection-based in Section 1.3, are different in their approach, but share the common goal of making the classifier more regular in the sense of a less curvy and more linear loss landscape $x \mapsto L^{\text{CE}}(F(\theta, x), c)$. Doing so, the attacker is offered less regions nearby to the natural image in $\mathcal{X}$ to be exploited for misclassification.

3.1 Adversarial Training by Robust Optimization

3.1.1 Adversarial Training Scheme

Standard adversarial training can be seen as an attempt to approximately solve the robust optimization problem

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i \in [N]} \sup_{\delta_i (\theta, x_i, c_i) \in \mathcal{U}_p^{\varepsilon_p}(x_i)} L^{\text{CE}}\left(F(\theta, x_i + \delta_i), c_i\right) \tag{E-RO}$$

during training. To do this, a solution to the inner adversarial problem

$$\sup_{\delta \in \mathcal{U}_p^{\varepsilon_p}(x)} \left[\xi(\theta, \delta, c) := L^{\text{CE}}\left(F(\theta, x + \delta), c\right)\right] \tag{ADV-UNT}$$

is estimated for each image in the training data set $\{(x_i, c_i) : i \in [N]\}$ in every epoch for the DNN to learn on. We formulate a generic adversarial training in Algorithm 2 to roughly encapsulate the different approaches in the literature [52, 101, 105, 37, 100, 3, 80, 95, 43, 28, 77, 82]. Here, we compute the approximate solutions to (ADV-UNT) by applying the first-order white-box methods from Section 2.1. Apart from this, a stochastic gradient descent procedure via backpropagation is performed, just like in natural training.

Within the template of Algorithm 2, we can express both defenses that we will test later. The R-FGSM training from LocusLab [95] uses $p = \infty$, $\varepsilon_p \in \{2.5/255, 5/255\}$, $F =$ ResNet50, $S = 1$, $\omega_1 = \varepsilon_p$, $R = \mathcal{U}\left[\overline{\mathcal{B}}_p^{\varepsilon_p}\right]$, $\alpha = 1$ for ImageNet and $F =$ ResNet110, $\varepsilon_p = 8/255$ for CIFAR-10. The multi-step PGD training from MadryLab [23] does not clearly state their training attacks, but it is a deterministic PGD approach with $S > 1$, $R = 0$ and constant attack step size as indicated in `https://github.com/MadryLab/robustness/issues/25`. Further questions regarding the other parameters remained unanswered. More details can be found in the respective papers and `GitHub` repositories. Tables 4.2 and 4.8 give a structured summary of the most important parameters.

There is a data scientific and a mathematical perspective to motivate adversarial training as in Algorithm 2:

Algorithm 2 – Adversarial-Training

Input: training set $(\{(x_i, c_i) : i \in [N]\} \subseteq \mathcal{X} \times \mathcal{C}$, weight $\alpha \in [0, 1]$, batch size $B \in \mathbb{N}$, number of epochs $E \in \mathbb{N}$, soft classifier $F : \Theta \times \mathcal{X} \to \mathcal{P}$, parameter initialization $\theta_0 \in \Theta$, learning rate $(\beta_s)_{e \in [E]} \subseteq \mathbb{R}^+$, norm exponent $p \geq 1$, adversarial radius $\varepsilon_p > 0$, number of adversarial steps $S \in \mathbb{N}$, adversarial step length $(\omega_s)_{s \in [S]} \subseteq \mathbb{R}^+$, adversarial randomization $R \in \mathcal{RV}(\mathbb{R}^n)$

Output: trained soft classifier $F(\theta_E, \cdot) : \mathcal{X} \to \mathcal{P}$

```
# epochs
for e = 1, ..., E do
    # partition
    ∪̇ B_e := [N] with |B_e| ≈ B
    # batches
    for i ∈ B_e do
        # perturb
        x̃_i := FO-WB-Attack(x_i, c_i, none, p, ε_p, F(θ_{e-1}, ·), S, (ω_s), R)
        # backpropagate
        g_e := g_e + α∇_θ L^CE(F(θ_{e-1}, x̃_i), c_i) + (1 − α)∇_θ L^CE(F(θ_{e-1}, x_i), c_i)
    end for
    g_e := g_e / B
    # update
    θ_e := θ_{e-1} − β_e g_e
end for
```

- **Data Augmentation:** Classifiers that are trained to classify clean images are trained on a data set containing as many natural images as possible to learn the map from unperturbed image data to the class labels, i.e. with $\alpha = 0$. Analogously, if a DNN is trained with adversarial robustness in mind, the goal for the classifier is to learn the map from adversarially perturbed data to the class labels. Therefore, it seems reasonable to let the classifier train on adversarial examples, i.e. with $\alpha > 0$. More generally, data augmentation has been successfully applied in many machine learning applications by artificially increasing the size of the training data set via generic label-preserving operations like scaling, flipping, cropping or rotating training data [83].
- **Danskin's Theorem:** A standard result in the field of robust optimization is Danskin's Theorem. It provides a characterization of directional derivatives and (sub-)gradients of functions that are defined as the optimal value of a maximization problem. In the min-max setting of (E-RO), this maximization corresponds to (ADV-UNT) while the first-order information provided by Danskin's Theorem is used to minimize the robust loss w.r.t. the model parameters θ.

All referenced papers either do not justify the methodology or constrict themselves to the first perspective, apart from [52] which shortly mentions the mathematical background. Interestingly, the assumptions of Danskin's Theorem are generally violated in the considered settings. Therefore, it is not clear why adversarial training proves practically successful in multiple domains [22]. Section 3.1.2 is the theoretical prerequisite to understand and motivate Contribution 4.3, which tries to bridge the gap between the non-existent theoretical guarantees and the empirical success.

3.1.2 Danskin's Theorem

Danskin's Theorem was first formulated by J. M. Danskin in 1967 in [19] and states the following.

Theorem 3.1 (Danskin's Theorem, cf. [19], [4], [5])**.** *Assume $\phi : \mathbb{R}^{d_1} \times U \to \mathbb{R}$ to be a continuous function of two arguments $r \in \mathbb{R}^{d_1}$ and $u \in U$ for a compact set $U \subseteq \mathbb{R}^{d_2}$. Further, suppose that $\phi(\cdot, u)$ is convex for every $u \in U$. By compactness and continuity, the sets*

$$U^*(r) := \arg\max_{u \in U} \phi(r, u) \quad \forall r \in \mathbb{R}^{d_1}$$

of maximizers are well-defined. So, we can also define the convex map

$$\varphi : \mathbb{R}^{d_1} \to \mathbb{R} : r \mapsto \max_{u \in U} \phi(r, u).$$

- *Then, the directional derivative of φ at r in the direction $v \in \mathbb{R}^{d_1}$ is given by*

$$\varphi'(r; v) = \max_{u \in U^*(r)} \phi'_r(r, u; v),$$

  where $\phi'_r(r, u; v)$ denotes the directional derivative of $\phi(\cdot, u)$ at and w.r.t. r in the direction v.

- *If $\phi(\cdot, u)$ is differentiable for all $u \in U^*(r)$ and $\nabla_r \phi(r, \cdot)$ is continuous on $U^*(r)$ for each $r \in \mathbb{R}^{d_1}$, then the subgradients are given by*

$$\partial\varphi(r) = \operatorname{conv}\{\nabla_r \phi(r, u) : u \in U^*(r)\} \quad \forall r \in \mathbb{R}^{d_1}.$$

- *In particular, if $U^*(r)$ is a singleton, i.e. $U^*(r) = \{u^*\}$, and $\phi(\cdot, u^*)$ is differentiable at r, then φ is differentiable at r and the gradient is given by $\nabla\varphi(r) = \nabla_r \phi(r, u^*)$.*

The underlying idea is to use Danskin's Theorem to compute

$$\nabla_\theta L^{\text{CE}}(F(\theta, \widetilde{x}), c) = \nabla_\theta \xi(\theta, \delta^*, c)$$

with adversarial example $\widetilde{x} = x + \delta^*$ and exact solution δ^* of (ADV-UNT). Hypothetically, if all of the above assumptions are satisfied, Theorem 3.1 would indeed directly imply how descent directions of the loss w.r.t. the model parameters can be found. They could be immediately employed to perform a gradient descent scheme.

Corollary 3.2 (θ-Descent Direction via Danskin, cf. [52] Cor. C.2)**.** *Assume $\delta^* \in \mathcal{U}_p^{\varepsilon_p}(x)$ to be the unique solution to the inner adversarial maximization* (ADV-UNT) *for a pair $(x, c) \in \mathcal{X} \times \mathcal{C}$. Further, suppose that $\xi(\cdot, \delta, c)$ is convex for every $\delta \in \mathcal{U}_p^{\varepsilon_p}(x)$ and $\xi(\cdot, 0, c)$ is differentiable. Then, $-\nabla_\theta \xi(\theta, \delta^*, c)$ is a descent direction of $\xi(\cdot, 0, c)$ in θ as long as θ is not optimal already.*

Proof. Since $\xi(\cdot, \cdot, c)$ is continuous, $\mathcal{U}_p^{\varepsilon_p}(x)$ is compact and the loss of the soft classifier F is assumed to be differentiable w.r.t. θ, we can apply Danskin's Theorem (Theorem 3.1) to get differentiability of ξ w.r.t. θ.

We use the smoothness of $L^{\text{CE}}(F(\cdot, x), c)$ to rewrite the directional derivative as a scalar product in the first equality and uniqueness of δ^* in the second equality to calculate

$$\begin{aligned}\xi'_\theta(\theta, 0, c; \nabla_\theta L^{\text{CE}}(F(\theta, x+\delta^*))) &= \max_{\delta \in \mathcal{U}_p^{\varepsilon p}} \nabla_\theta L^{\text{CE}}(F(\theta, x+\delta), c)^\top \nabla_\theta L^{\text{CE}}(F(\theta, x+\delta^*), c) \\ &= \nabla_\theta L^{\text{CE}}(F(\theta, x+\delta^*), c)^\top \nabla_\theta L^{\text{CE}}(F(\theta, x+\delta^*), c) \\ &= \|\nabla_\theta L^{\text{CE}}(F(\theta, x+\delta^*), c)\|_2^2 \\ &\geq 0.\end{aligned}$$

Note that the inequality and signs flip when considering the derivative in the opposite direction. Strict inequality follows from $\nabla_\theta L^{\text{CE}}(F(\theta, x+\delta^*), c) \neq 0$, which holds true if θ is not optimal. This comes from the fact that stationarity is a necessary optimality condition for unconstrained problems and the lack of constraints on $\theta \in \Theta$ in (E-RO). ■

However, even when all other assumptions are fulfilled, the validity of Theorem 3.1 is very sensitive towards inexactness in the computation of $U^*(r)$, i.e. the solutions of $\max_{u \in U} \phi(r, u)$, as our example below highlights.

Example 3.3 (Failure of Danskin's Theorem with Absolute Tolerance). For some arbitrary tolerance $\tau > 0$, consider the quadratic function

$$\phi : \mathbb{R} \times [-1, 1] \to \mathbb{R}, \quad (r, u) \mapsto \frac{(r-u)^2 + \tau u}{2}.$$

It has a ascending symmetric valley along the line $(0, -\tau/2) + \{(r, u) \in \mathbb{R}^2 : r = u\}$ that roughly bisects the third and first quadrant, as can be seen in Figure 3.1.

Then, we see by direct calculation that $U^*(0) = \{1\}$ is the unique maximizer along $r = 0$. However, on the opposite side of U, there is a point of comparable value with a difference of only

$$\phi(0, 1) - \phi(0, -1) = \tau.$$

Despite this, comparing the gradient

$$\nabla\phi(r, u) = \begin{pmatrix} r - u \\ u - r + \tau \end{pmatrix}$$

in these points, yields

$$\nabla\phi(0, 1) = \begin{pmatrix} -1 \\ \tau + 1 \end{pmatrix} \quad \text{and} \quad \nabla\phi(0, -1) = \begin{pmatrix} 1 \\ \tau - 1 \end{pmatrix}$$

with an enclosed angle of

$$\cos(\measuredangle(\nabla\phi(0, 1), \nabla\phi(0, -1))) = \frac{\langle \nabla\phi(0, 1), \nabla\phi(0, -1) \rangle_2}{\|\nabla\phi(0, 1)\|_2 \|\nabla\phi(0, -1)\|_2} = \frac{\tau^2 - 2}{\sqrt{\tau^4 + 4}} \to -1$$

when $\tau \searrow 0$. This means that the angle $\measuredangle(\nabla\phi(0, 1), \nabla\phi(0, -1))$ between the gradients approaches π when τ vanishes. Hence, maximizing $\phi(0, \cdot)$ slightly imprecisely on $U = [-1, 1]$, with an absolute tolerance larger than τ, already yields a completely different optimizer u^* and almost entirely opposing gradients. □

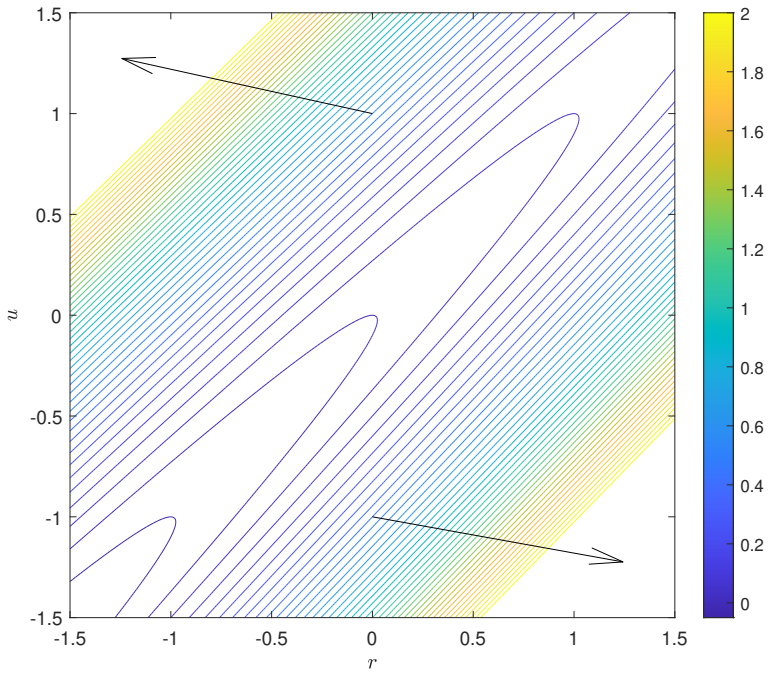


Figure 3.1: Isolines of $\phi(r, u)$ with gradients in $(0, 1)$ and $(0, -1)$ for $\tau = 0.1$. Gradients are scaled by a factor of 0.2 for visualization.

So we have seen that even for a fairly simple quadratic problem in two dimensions, we must solve the maximization subproblem exactly to have the theoretical guarantees of Theorem 3.1. This is especially concerning since our actual adversarial problem (ADV-UNT) is highly non-linear and of much higher dimension n. In fact, solving such problems exactly is NP-hard [84]. Beyond that, local maxima of the inner problem tend to have clustered optimal values and appear almost uniformly distributed within the feasible region $\mathcal{U}_p^{\varepsilon_p}(x)$ [52].

Fundamentally, it is not only hard to satisfy the prerequisites of Theorem 3.1 to leverage Corollary 3.2, but actually not possible in the context of (ADV-UNT) because multiple assumptions are generally violated by default. Understanding $\xi_c := \xi(\cdot, \cdot, c) : \mathbb{R}^{\dim(\Theta)} \times \left(\mathcal{U}_p^{\varepsilon_p}(x) \subseteq \mathbb{R}^n\right) \to \mathbb{R}^+$ as $\phi : \mathbb{R}^{d_1} \times \left(U \subseteq \mathbb{R}^{d_2}\right) \to \mathbb{R}$, we identify multiple shortcomings:

- ξ_c is not continuous w.r.t. x and might not even be defined on the entirety of $\mathcal{U}_p^{\varepsilon_p}(x)$ as the CEL can contain singularities on $\mathcal{U}_p^{\varepsilon_p}(x)$, cf. Contribution 4.1.
- Consequently, $\arg\max_{\delta \in \mathcal{U}_p^{\varepsilon_p}(x)} \xi_c(\theta, \delta)$ is not well-defined as (ADV-UNT) is unbounded above.
- ξ_c is not convex w.r.t. θ in all $\delta \in \mathcal{U}_p^{\varepsilon_p}(x)$.
- ξ_c is not differentiable w.r.t θ for all $\delta \in \mathcal{U}_p^{\varepsilon_p}(x)$ when ReLU-activations are used.

Example 3.3 and all of the above considerations might suggest that there is no chance of capitalizing on Danskin's Theorem for Algorithm 2. Fortunately, in practice, this method proved to be successful anyway in a large number of papers [52, 101, 105, 37, 100, 3, 80, 95, 43, 28, 77, 82]. In Section 4.3, we attempt to give an explanation for this and answer the question from [80], why adversarial training increases robustness.

3.2 Robustification by Randomized Smoothing

Randomized smoothing [57, 69, 101, 60, 16, 70, 78, 27, 103] can be seen as an attempt to solve

$$\min_{\theta\in\Theta} \widehat{R}_{\text{SO}}(\theta) = \frac{1}{MN} \sum_{i\in[N]} \sum_{j\in[M]} L^{\text{CE}}\left(F(\theta, x_i + \delta_{ij}), c_i\right) \tag{E-SO}$$

for some $M \in \mathbb{N}$ and i.i.d. samples $(\delta_{ij}) \sim \mathcal{N}(0, \sigma^2 I)$. Hence, it trains the DNN to be accurate under Gaussian white noise. Smoothing with differently distributed RVs is possible. However, isotropic Gaussian smoothing is preferred as it puts more emphasis on small perturbations close to the empirical data distribution without completely neglecting outliers. Other than that, there are signs that it induces smoother decision boundaries than uniform smoothing [101, 55] and many papers [57, 101, 16, 78, 103] achieve promising results with it. Nevertheless, other structurally different and remarkably successful smoothing techniques exist [69, 70, 60]. For the above reasons and the sake of comparability to existing theoretical and empirical results, we consider Gaussian smoothing. In particular, isotropic Gaussian smoothing induces a tight bound on 2-norm robustness, as Theorems 3.8 and 3.12 will show. To the best of our knowledge, there have not been established similar bounds for other pairs of RVs and (p-)norms since the result in [16] for $p = 2$.

Randomized smoothing, as we consider it [16, 78, 103], is a two step process: At training time, the images are adversarially attacked and perturbed with Gaussian noise before backpropagation takes place in the model parameter space. At query time, the input is perturbed multiple times by Gaussian noise and the results are fed into a 'voting function' [57] that merges these classification results into a final output. Following the chronology, we will first introduce the noise injection during training before looking at the rationale of repeated querying.

3.2.1 Noise Injection at Training Time

Training F with noisy images is necessary for the smoothed soft classifier F_σ to be sufficiently accurate when smoothing at query time. In resemblance of Pseudocode 1 of [78], this can be accomplished by a version of the generic training scheme of Algorithm 3.

Note that in line eight, Algorithm 1 must compute the gradient

$$g = \nabla_x L^{\text{CE}}(F_\sigma(\theta, x), c) = \nabla_x \mathbb{E}_{\delta\sim\mathcal{N}(0,\sigma^2 I)}[F(\theta, x + \delta)].$$

As an estimate, [78] computes a stochastic gradient via Monte Carlo sampling

$$g \approx \nabla_x L^{\text{CE}}\left(\frac{1}{M_{\text{train}}} \sum_{j\in[M_{\text{train}}]} F(\theta, x + \delta_j)_c\right) = -\nabla_x \log\left(\frac{1}{M_{\text{train}}} \sum_{j\in[M_{\text{train}}]} F(\theta, x + \delta_j)_c\right)$$

for some $M_{\text{train}} \in \mathbb{N}$ and i.i.d. realizations $(\delta_j) \sim \mathcal{N}(0, \sigma^2)$. In a trade-off between accuracy and cost, [78] recommends $M_{\text{train}} \in \{1, 2, 4, 8\}$.

Algorithm 3 – Smoothing-Training

Input: training set $(\{(x_i, c_i) : i \in [N]\} \subseteq \mathcal{X} \times \mathcal{C}$, weight $\alpha \in [0, 1]$, batch size $B \in \mathbb{N}$, number of epochs $E \in \mathbb{N}$, soft classifier $F : \Theta \times \mathcal{X} \to \mathcal{P}$, parameter initialization $\theta_0 \in \Theta$, learning rate $(\beta_s)_{e\in[E]} \subseteq \mathbb{R}^+$, norm exponent $p \geq 1$, adversarial radius $\varepsilon_p > 0$, number of adversarial steps $S \in \mathbb{N}$, adversarial step length $(\omega_s)_{s\in[S]} \subseteq \mathbb{R}^+$, adversarial randomization $R \in \mathcal{RV}(\mathbb{R}^n)$, smoothing variance $\sigma^2 \in (0, \infty)$

Output: trained soft classifier $F(\theta_E, \cdot) : \mathcal{X} \to \mathcal{P}$

# epochs
for $e = 1, \ldots, E$ **do**
    # partition
    $\dot{\cup} B_e := [N]$ with $|B_e| \approx B$
    # batches
    for $i \in B_e$ **do**
        # perturb
        $\widetilde{x}_i :=$ FO-WB-Attack$(x_i, c_i, \textit{none}, p, \varepsilon_p, F_\sigma(\theta_{e-1}, \cdot), S, (\omega_s), R)$
        # backpropagate
        $g_e := g_e + \alpha \nabla_\theta L^{\text{CE}}(F(\theta_{e-1}, \widetilde{x}_i), c_i) + (1 - \alpha)\nabla_\theta L^{\text{CE}}(F(\theta_{e-1}, x_i), c_i)$
    end for
    $g_e := g_e / B$
    # update
    $\theta_e := \theta_{e-1} - \beta_e g_e$
end for

3.2.2 Smoothing at Query Time

Having obtained a robust classifier that can deal with Gaussian noise injection, we can further smooth its prediction at query time by drawing multiple normal distributed samples around the query. To formalize this idea, we need the definitions below.

Definition 3.4 (Smoothed Classifiers, cf. [78])**.** *For a soft classifier $F : \Theta \times \mathbb{R}^n \to \mathcal{P}$, define the corresponding smoothed soft classifier under isotropic Gaussian noise by the convolution*

$$F_\sigma : \Theta \times \mathbb{R}^n \to \mathcal{P},\ (\theta, x) \mapsto \big(F(\theta, \cdot) * \mathcal{N}(0, \sigma^2 I)\big)(x) = \mathbb{E}_{\delta\sim\mathcal{N}(0,\sigma^2 I)}[F(\theta, x + \delta)]$$

with variance $\sigma^2 \in (0, \infty)$. By extension, for a hard classifier

$$f : \Theta \times \mathbb{R}^n \to \mathcal{C},\ (\theta, x) \mapsto \operatorname*{arg\,max}_{c\in\mathcal{C}} F(\theta, x),$$

denote the derived smoothed hard classifier with voting function $v : \mathcal{P} \times \mathcal{C} \to \mathbb{R}$ by

$$f_v : \Theta \times \mathbb{R}^n \to \mathcal{C},\ (\theta, x) \mapsto \operatorname*{arg\,max}_{c\in C} \mathbb{E}_{\delta\sim\mathcal{N}(0,\sigma^2 I)}[v(F(\theta, x + \delta), c)].$$

For $v(p, c) := p_c$, we obtain the soft smoothed hard classifier

$$f_s : \Theta \times \mathbb{R}^n \to \mathcal{C},\ (\theta, x) \mapsto \operatorname*{arg\,max}_{c\in C} \mathbb{E}_{\delta\sim\mathcal{N}(0,\sigma^2 I)}[F(\theta, x + \delta)_c].$$

The choice $v(p,c) := \mathbb{1}\left[\arg\max_{c' \in \mathcal{C}} p_{c'} = c\right]$ *gives us the hard smoothed hard classifier*

$$f_h : \Theta \times \mathbb{R}^n \to \mathcal{C},\ (\theta, x) \mapsto \arg\max_{c \in C} \mathbb{P}_{\delta \sim \mathcal{N}(0,\sigma^2 I)}[f(\theta, x+\delta) = c].$$

As presented in Section 1.4.2, the CEL can be understood as a smooth approximation to the $\{0,1\}$-loss and the approximation gets tighter for sinking temperature $t > 0$. The aforementioned convergence directly implies

$$\mathbb{E}_\delta[F(\theta, x+\delta)_c] \to \mathbb{P}_\delta[f(\theta, x+\delta) = c] \quad \text{as} \quad t \searrow 0.$$

Therefore, the voting functions of f_h and f_s are more likely to agree for lower temperatures and more likely to disagree for higher ones. Also note that [16, 78, 103] are distinguished by the voting function they use. As before, it is infeasible to compute the expectations in Definition 3.4 exactly. Instead, they are once again estimated with a Monte Carlo simulation.

Definition 3.5 (Approximated Smoothed Classifiers)**.** *For each* $(\theta, x) \in \Theta \times \mathbb{R}^n$*, define the approximated smoothed classifiers as the RVs*

$$\widehat{F_\sigma}(\theta, x) := \frac{1}{M_{query}} \sum_{j \in [M_{query}]} F(\theta, x + \delta_j) \in \mathcal{RV}(\mathcal{P}),$$

$$\widehat{f_h}(\theta, x) := \arg\max_{c \in C} \frac{1}{M_{query}} \sum_{j \in [M_{query}]} \mathbb{1}[f(\theta, x+\delta_j) = c] \in \mathcal{RV}(\mathcal{C})$$

and

$$\widehat{f_s}(\theta, x) := \arg\max_{c \in C} \frac{1}{M_{query}} \sum_{j \in [M_{query}]} F(\theta, x + \delta_j) \in \mathcal{RV}(\mathcal{C})$$

for $M_{query} \in \mathbb{N}$ *and i.i.d. samples* $(\delta_j) \sim \mathcal{N}(0, \sigma^2 I)$.

M_{query} can be significantly larger than M_{train} as accuracy is more and speed less relevant than during training. For instance [78], advice M_{query} to be possibly as high as 128 when evaluating the robustness of the DNN. Nevertheless, for regular usage a more moderate sample size seems appropriate to reduce costs and waiting times.

Beyond that, Definitions 3.4 and 3.5 prompt another approach to interpreting problems (SO) and (E-SO): The stochastic optimization problem (SO) can be interpreted as natural training without adversary (CO) of the smoothed soft classifier

$$R_{\text{SO}}(\theta) = \mathbb{E}_{(X,C)\sim\mathbb{D}}\left[\mathbb{E}_{\Delta\sim\mathcal{N}(0,\sigma^2 I)}\left[L^{\text{CE}}\left(F(\theta, X+\Delta), C\right)\right]\right] = \mathbb{E}_{(X,C)\sim\mathbb{D}}\left[L^{\text{CE}}\left(F_\sigma(\theta, X), C\right)\right].$$

Similarly, if $M := M_{\text{train}} = M_{\text{query}}$ and the i.i.d. samples $(\delta_j) \sim \mathcal{N}(0, \sigma^2 I)$ from Definition 3.5 are assumed to be independent for different images, then the empirical stochastic optimization problem (E-SO) reduces to the empirical clean optimization problem (E-CO) for the approximated smoothed soft classifier

$$\widehat{R}_{\text{SO}}(\theta) = \frac{1}{MN} \sum_{i \in [N]} \sum_{j \in [M]} L^{\text{CE}}\left(F(\theta, x_i + \delta_{ij}), c_i\right) = \frac{1}{N} \sum_{i \in [N]} \widehat{F_\sigma}(\theta, x_i).$$

To establish some desirable convergence properties from $\widehat{F_\sigma}$ to F, we formulate the following lemma. For readability and because they stay unchanged, we omit the indices on the sample size M and the dependence of all classifiers on θ from now on in this section.

Lemma 3.6. *For i.i.d. RVs $(\delta_j) \sim \mathcal{N}(0, \sigma^2 I)$ with finite variance $\sigma^2 \in (0, \infty)$, the approximated smoothed classifiers are unbiased, in the sense that*

$$F_\sigma(x) = \mathbb{E}\left[\widehat{F_\sigma}(x)\right] \quad \textit{and} \quad f_s(x) = \arg\max_{c \in C} \mathbb{E}\left[\frac{1}{M} \sum_{j=[M]} \mathbb{1}[f(x + \delta_j) = c]\right].$$

Furthermore, they are consistent estimators not only in probability, but even almost surely, meaning

$$\widehat{F_\sigma}(x) \overset{a.s.}{\to} F_\sigma(x) \text{ as } M \to \infty$$

and

$$\widehat{f_s}(x) \overset{a.s.}{\to} f_s(x) \text{ as } M \to \infty$$

in any point $x \in \mathbb{R}^n$.

Proof. Let $x \in \mathbb{R}^n$ and $\sigma^2 \in (0, \infty)$ be arbitrary. Viewing $\widehat{F_\sigma}(x + \delta_j) \in \mathcal{RV}(\mathcal{P})$ and $\widehat{f_s}(x + \delta_j) \in \mathcal{RV}(\mathcal{C})$ as RVs for all $j \in [M]$, we can apply well-known results from probability theory to deduce the claim. For faithfulness, we use $\delta_j \sim \delta$ for all $j \in [M]$ to derive

$$\begin{aligned}
\mathbb{E}\left[\widehat{F_\sigma}(x)\right] &= \frac{1}{M} \sum_{j \in [M]} \mathbb{E}_{\delta_j}[F(x + \delta_j)] \\
&= \frac{1}{M} \sum_{j \in [M]} \mathbb{E}_{\delta}[F(x + \delta)] \\
&= \frac{M}{M} \mathbb{E}_{\delta}[F(x + \delta)] \\
&= F_\sigma(x)
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{M} \sum_{j \in [M]} \mathbb{1}[f(x + \delta_j) = c]\right] &= \frac{1}{M} \sum_{j \in [M]} \mathbb{E}_{\delta_j}\left[\mathbb{1}[f(x + \delta_j) = c]\right] \\
&= \frac{1}{M} \sum_{j \in [M]} \mathbb{P}_{\delta_j}[f(x + \delta_j) = c] \\
&= \frac{1}{M} \sum_{j \in [M]} \mathbb{P}_{\delta}[f(x + \delta) = c] \\
&= \frac{M}{M} \mathbb{P}_{\delta}[f(x + \delta) = c] \\
&= \mathbb{P}_{\delta}[f(x + \delta) = c].
\end{aligned}$$

Hence, taking arg max in the second equality concludes the first part of the proof. For the second part, we note that the a.s. convergence follows from the strong law of large numbers (SLLN) for $\widehat{F_\sigma}(x)$ and from Borel's law of large numbers [94] for $\widehat{f_s}(x)$. Finally, convergence in probability is universally implied by a.s. convergence. ■

Because a.s. convergence implies convergence in probability, Hoeffding's inequality provides a probabilistic bound on the convergence speed of the approximated smoothed soft classifier to the smoothed classifier.

Proposition 3.7 (Probabilistic Bound on Convergence of Sampling Accuracy)**.** *In the context of Lemma 3.6, it holds*

$$\mathbb{P}\left[\left\|\widehat{F_\sigma}(x) - F_\sigma(x)\right\|_\infty \geq \kappa\right] \leq \exp\left(-2\kappa^2 M\right) \quad \forall \kappa \geq 0.$$

Proof. Recalling $\widehat{F_\sigma}(x)_c \in [0,1]$ and $\mathbb{E}\left[\widehat{F_\sigma}(x)_c\right] = F_\sigma(x)_c$ from Lemma 3.6, Hoeffding's inequality implies the bound for each class $c \in \mathcal{C}$. The uniform bound follows from component-wise application. ■

The exponential upper bound gives us hope to achieve a decent approximation accuracy with sufficient confidence κ even for moderate M if the variance σ^2 is comparatively small. In this section, we defined the main mathematical objects of randomized smoothing and how they can be approximated in practice. The next section will use these objects to derive a theoretical certified robustness bound of Gaussian smoothing.

3.2.3 Certified Robustness

Certified robustness bounds provide a neighbourhood around a clean image in which the classifiers prediction does not change. Therefore, from the perspective of p-norm bounded attacks it is the most natural and favourable robustness guarantee. Improving upon previously shown robustness guarantees in [46] and [44], [16] proved a tight 2-norm robustness bound for Gaussian smoothing which we will retrace in this section.

Theorem 3.8 (Certified 2-Norm Robustness via Randomized Smoothing, cf. [16] Thm. 1)**.** *Suppose $f, f_h : \Theta \times \mathbb{R}^n \to \mathcal{C}$ and $\Delta \sim \mathcal{N}(0, \sigma^2 I)$ as in Definition 3.4. Further, assume that for a label $c_A \in \mathcal{C}$, there exist confidences $\underline{p_A}$, $\overline{p_B} \in [0,1]$ such that*

$$\mathbb{P}[f(\theta, x+\Delta) = c_A] \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}[f(\theta, x+\Delta) = c]. \tag{3.1}$$

Then, $\widehat{f_h}(\theta, x+\delta) = c_A$ for all $\delta \in \overline{\mathcal{B}}_2^r$ with radius

$$r := r(\sigma, \underline{p_A}, \overline{p_B}) := \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})\right).$$

We defer the proof for later when we have established some preliminary results in the form of Lemmas 3.10 and 3.11. Instead, we first make the below remark.

Remark 3.9. We observe the following:

- There are no assumptions on f. This is beneficial when it is unclear to what extent the DNN is well-behaved or suffices some regularity conditions like in a black-box setting.
- The certified radius of robustness $r(\sigma, \underline{p_A}, \overline{p_B})$ is monotonically increasing in σ and $\underline{p_A}$ as well as monotonically decreasing in $\overline{p_B}$. This is consistent with the intuition that the smoothed classifier is more robust towards small shifts δ when it is confident in distinguishing the most probable class c_A from the runner-up c_B.
- When $\underline{p_A} \to 1$ and $\overline{p_B} \to 0$ converge, then $r \to \infty$ diverges. This is reasonable due to the Gaussian distribution being supported on all of $\mathbb{R}^n$. So, for $f(\theta, x+\Delta) = c_A$ to hold a.s., it must hold $f(\theta, \cdot) \equiv c_A$ almost everywhere (a.e.).

- The confidences $\underline{p_A}$ and $\overline{p_B}$ cannot be determined precisely. However, Lemma 2, Theorem 3 and Algorithm 2 in [103] as well as [16] and [43] provide in-depth information on their estimation.
- Theorem 2 of [103] shows an analogous result for the soft smoothed hard classifier f_s. It gives a different approach to certification and allows for direct maximization of r via the robust training via maximizing the certified radius (MACER) algorithm. The bounds for f_s and f_h converge as $t \searrow 0$ with the justification from Section 1.4.2 [43].

As a first step towards proving Theorem 3.8, we recall a standard result from mathematical statistics called Neyman-Pearson Lemma.

Lemma 3.10 (Neyman-Pearson, cf. [58], [16] Lem. 3)**.** *Let $X, Y \in \mathcal{RV}(\mathbb{R}^n)$ be RVs with densities ρ_X, ρ_Y and let $h : \mathbb{R}^n \to \{0,1\}$ be a deterministic or random function. Then,*

$$\exists \gamma > 0 \quad \textit{s.t.} \quad \mathbb{P}[h(X) = 1] \geq \mathbb{P}[X \in S_{\leq}] \quad \textit{for} \quad S_{\leq} := \left\{ z \in \mathbb{R}^n : \frac{\rho_Y(z)}{\rho_X(z)} \leq \gamma \right\} \tag{3.2}$$

implies $\mathbb{P}[h(Y) = 1] \geq \mathbb{P}[Y \in S_{\leq}]$ and conversely,

$$\exists \gamma > 0 \quad \textit{s.t.} \quad \mathbb{P}[h(X) = 1] \leq \mathbb{P}[X \in S_{\geq}] \quad \textit{for} \quad S_{\geq} := \left\{ z \in \mathbb{R}^n : \frac{\rho_Y(z)}{\rho_X(z)} \geq \gamma \right\} \tag{3.3}$$

implies $\mathbb{P}[h(Y) = 1] \leq \mathbb{P}[Y \in S_{\geq}]$.

Proof. We assume w.l.o.g. that h is random because the deterministic case can be dealt with by setting $\mathbb{P}[h(x) = 0] = 1 - \mathbb{P}[h(x) = 0] \in \{0,1\}$. For the first implication, we estimate

$$\begin{aligned}
\mathbb{P}[h(Y) = 1] - \mathbb{P}[Y \in S_{\leq}] &= \int_{\mathbb{R}^n} \mathbb{P}[h(z) = 1] \rho_Y(z) dz - \int_{S_{\leq}} \rho_Y(z) dz \\
&= \left[\int_{S_{\leq}{}^c} \mathbb{P}[h(z) = 1] \rho_Y(z) dz + \int_{S_{\leq}} \mathbb{P}[h(z) = 1] \rho_Y(z) dz \right] \\
&\quad - \left[\int_{S_{\leq}} \mathbb{P}[h(z) = 1] \rho_Y(z) dz + \int_{S_{\leq}} \mathbb{P}[h(z) = 0] \rho_Y(z) dz \right] \\
&= \int_{S_{\leq}{}^c} \mathbb{P}[h(z) = 1] \rho_Y(z) dz - \int_{S_{\leq}} \mathbb{P}[h(z) = 0] \rho_Y(z) dz \\
&\geq \gamma \left[\int_{S_{\leq}{}^c} \mathbb{P}[h(z) = 1] \rho_X(z) dz - \int_{S_{\leq}} \mathbb{P}[h(z) = 0] \rho_X(z) dz \right] \\
&= \gamma \Big[\int_{S_{\leq}{}^c} \mathbb{P}[h(z) = 1] \rho_X(z) dz + \int_{S_{\leq}} \mathbb{P}[h(z) = 1] \rho_X(z) dz \\
&\quad - \int_{S_{\leq}} \mathbb{P}[h(z) = 1] \rho_X(z) dz - \int_{S_{\leq}} \mathbb{P}[h(z) = 0] \rho_X(z) dz \Big] \\
&= \gamma \left[\int_{\mathbb{R}^n} \mathbb{P}[h(z) = 1] \rho_X(z) dz - \int_{S_{\leq}} \rho_X(z) dz \right] \\
&= \gamma \left[\mathbb{P}[h(X) = 1] - \mathbb{P}[X \in S_{\leq}] \right] \\
&\geq 0.
\end{aligned}$$

In the first inequality, we used

$$\rho_Y(z) \leq \gamma\rho_X(z) \ \forall z \in S_{\leq} \ \text{ and } \ \rho_Y(z) > \gamma\rho_X(z) \ \forall z \in S_{\leq}^c$$

and in the second, our assumption on the non-negativity of both factors. The other implication works analogously by reversing both inequalities. ■

Building upon this general result, we focus on our distribution of interest, namely isotropic Gaussians.

Lemma 3.11 (Neyman-Pearson for Gaussian White Noise, cf. [16] Lem. 4)**.** *Let $X \sim \mathcal{N}(x, \sigma^2 I)$ and $Y \sim \mathcal{N}(x+\delta, \sigma^2 I)$ be $\mathbb{R}^n$-valued RVs, i.e. $X, Y \in \mathcal{RV}(\mathbb{R}^n)$ with densities ρ_X, ρ_Y. Again, consider a random or deterministic function $h : \mathbb{R}^n \to \{0,1\}$ and some scalar $\beta \in \mathbb{R}$. Denoting the half-spaces induced by δ and β by*

$$S_{\leq} := \{z \in \mathbb{R}^n : \delta^\top z \leq \beta\} \quad \textit{and} \quad S_{\geq} := \{z \in \mathbb{R}^n : \delta^\top z \geq \beta\}, \tag{3.4}$$

we assert

$$\mathbb{P}[h(X) = 1] \geq \mathbb{P}[X \in S_{\leq}] \implies P[h(Y) = 1] \geq \mathbb{P}[Y \in S_{\leq}] \tag{3.5}$$

and conversely

$$\mathbb{P}[h(X) = 1] \leq \mathbb{P}[X \in S_{\geq}] \implies P[h(Y) = 1] \leq \mathbb{P}[Y \in S_{\geq}]. \tag{3.6}$$

Proof. The claim is a consequence of Lemma 3.10 for X and Y being the isotropic Gaussian from the assumption. Hence, what remains to be checked is that for each β there exists some $\gamma > 0$ such that the half spaces induced by δ and β coincide with those induced by γ-bounded density ratios. More precisely, we show

$$\begin{aligned}\forall \beta \in \mathbb{R} \ \exists \gamma = \gamma(\beta) \quad \text{s.t.} \quad \{z \in \mathbb{R}^n : \delta^\top z \leq \beta\} &= \left\{z \in \mathbb{R}^n : \frac{\rho_Y(z)}{\rho_X(z)} \leq \gamma\right\} \\ \text{and} \quad \{z \in \mathbb{R}^n : \delta^\top z \geq \beta\} &= \left\{z \in \mathbb{R}^n : \frac{\rho_Y(z)}{\rho_X(z)} \geq \gamma\right\},\end{aligned}$$

legitimizing the notational ambiguity of $S_{\leq}$ and $S_{\geq}$ between Lemmas 3.10 and 3.11. Plugging in Gaussian densities component-wise, we calculate

$$\begin{aligned}\frac{\rho_Y(z)}{\rho_X(z)} &= \frac{\exp\left(-\frac{1}{2\sigma^2}\sum_{i\in[n]}(z_i - (x_i+\delta_i))^2\right)}{\exp\left(-\frac{1}{2\sigma^2}\sum_{i\in[n]}(z_i - x_i)^2\right)} \\ &= \exp\left(\frac{1}{2\sigma^2}\sum_{i\in[n]}(2z_i\delta_i - \delta_i^2 - 2x_i\delta_i)\right) \\ &= \exp\left(c_1\delta^\top z + c_2\right)\end{aligned}$$

for constants $c_1 := 1/\sigma^2$ and $c_1 := -(2\delta^\top x + \|\delta\|_2^2)/(2\sigma^2)$. Choosing $\gamma := \exp(c_1\beta + c_2)$, we observe

$$\begin{aligned}\delta^\top z \leq \beta &\iff \exp(c_1\delta^\top z + c_2) \leq \gamma \\ \delta^\top z \geq \beta &\iff \exp(c_1\delta^\top z + c_2) \geq \gamma,\end{aligned}$$

which concludes the proof. ■

We return to our primary goal of proving the result on certified 2-norm robustness via randomized smoothing.

Proof of Theorem 3.8. To show the robustness property $f_s(\theta, x+\delta) = c_A$ for all $\delta \in \overline{\mathcal{B}}_2^r$ of the smoothed classifier, we need to show

$$\mathbb{P}_\delta[f_s(\theta, x+\delta+\Delta) = c_A] > \mathbb{P}_\delta[f_s(\theta, x+\delta+\Delta) = c_B] \quad \forall x \in \mathbb{R}^n,\ c_B \in \mathcal{C} \setminus \{c_A\}.$$

Now, w.l.o.g. fix arbitrary $\delta \in \overline{\mathcal{B}}_2^r$, $x \in \mathbb{R}^n$ and $c_B \neq c_A$. Additionally, define the RVs

$$X := x + \Delta \sim \mathcal{N}(x, \sigma^2 I)$$
$$Y := x + \delta + \Delta \sim \mathcal{N}(x+\delta, \sigma^2 I)$$

to reformulate the assumption (3.1) to

$$\mathbb{P}[f(\theta, X) = c_A] \geq \underline{p_A} \quad \text{and} \quad \mathbb{P}[f(\theta, X) = c_B] \leq \overline{p_B} \tag{3.7}$$

and the claim to

$$\mathbb{P}[f(\theta, Y) = c_A] > \mathbb{P}[f(\theta, Y) = c_B]. \tag{3.8}$$

To get started, we want to comply with the framework of Lemma 3.11 by defining the half-spaces

$$S_A := \{z \in \mathbb{R}^n : \delta^\top (z - x) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\}, \tag{3.9}$$
$$S_B := \{z \in \mathbb{R}^n : \delta^\top (z - x) \geq \sigma \|\delta\|_2 \Phi^{-1}(1 - \overline{p_B})\}. \tag{3.10}$$

We compute

$$\begin{aligned}
\mathbb{P}[X \in S_A] &= \mathbb{P}\left[\delta^\top (X - x) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\delta^\top \mathcal{N}(0, \sigma^2 I) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\sigma \delta^\top \mathcal{N}(0, I) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\sigma \|\delta\|_2 \mathcal{N}(0,1) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\mathcal{N}(0,1) \leq \Phi^{-1}(\underline{p_A})\right] \\
&= \Phi(\Phi^{-1}(\underline{p_A})) \\
&= \underline{p_A}
\end{aligned}$$

and similarly

$$\begin{aligned}
\mathbb{P}[X \in S_B] &= \mathbb{P}\left[\delta^\top (X - x) \geq \sigma \|\delta\|_2 \Phi^{-1}(1 - \overline{p_B})\right] \\
&= \mathbb{P}\left[\delta^\top \mathcal{N}(0, \sigma^2 I) \geq \sigma \|\delta\|_2 \Phi^{-1}(1 - \overline{p_B})\right] \\
&= \mathbb{P}\left[\sigma \delta^\top \mathcal{N}(0, I) \geq \sigma \|\delta\|_2 \Phi^{-1}(1 - \overline{p_B})\right] \\
&= \mathbb{P}\left[\sigma \|\delta\|_2 \mathcal{N}(0,1) \geq \sigma \|\delta\|_2 \Phi^{-1}(1 - \overline{p_B})\right] \\
&= \mathbb{P}\left[\mathcal{N}(0,1) \geq \Phi^{-1}(1 - \overline{p_B})\right] \\
&= 1 - \Phi(\Phi^{-1}(1 - \overline{p_B})) \\
&= \overline{p_B}.
\end{aligned}$$

Using (3.7), these identities yield

$$\mathbb{P}[f(\theta, X) = c_A] \geq \mathbb{P}[X \in S_A] \quad \text{and} \quad \mathbb{P}[f(\theta, X) = c_B] \leq \mathbb{P}[X \in S_B].$$

Applying the Gaussian Neyman-Pearson Lemma 3.11 with $h(z) := \mathbb{1}[f(z) = c_A]$ and $h(z) := \mathbb{1}[f(z) = c_B]$, respectively, we conclude

$$\mathbb{P}[f(\theta, Y) = c_A] \geq \mathbb{P}[Y \in S_A] \quad \text{and} \quad \mathbb{P}[f(\theta, Y) = c_B] \leq \mathbb{P}[Y \in S_B].$$

Connecting both inequalities by showing $\mathbb{P}[Y \in S_A] > \mathbb{P}[Y \in S_B]$ would complete the chain of inequalities to

$$\mathbb{P}[f(\theta, Y) = c_A] \geq \mathbb{P}[Y \in S_A] > \mathbb{P}[f(\theta, Y) = c_B] \leq \mathbb{P}[Y \in S_B]$$

and prove the reformulated claim in (3.8). To this end, we first compute

$$\begin{aligned}
\mathbb{P}\left[Y \in S_A\right] &= \mathbb{P}\left[\delta^\top (Y - x) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\delta^\top \mathcal{N}(\delta, \sigma^2 I) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\delta^\top \mathcal{N}(0, \sigma^2 I) + \|\delta\|_2^2 \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\sigma \delta^\top \mathcal{N}(0, I) + \|\delta\|_2^2 \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\sigma \|\delta\|_2 \mathcal{N}(0, 1) + \|\delta\|_2^2 \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A})\right] \\
&= \mathbb{P}\left[\sigma \|\delta\|_2 \mathcal{N}(0, 1) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A}) - \|\delta\|_2^2\right] \\
&= \mathbb{P}\left[\mathcal{N}(0, 1) \leq \Phi^{-1}(\underline{p_A}) - \frac{\|\delta\|_2}{\sigma}\right] \\
&= \Phi\left(\Phi^{-1}(\underline{p_A} - \frac{\|\delta\|_2}{\sigma})\right)
\end{aligned} \tag{3.11}$$

and analogously

$$\begin{aligned}
\mathbb{P}\left[Y \in S_B\right] &= \mathbb{P}\left[\delta^\top (Y - x) \geq \sigma \|\delta\|_2 \Phi^{-1}(\overline{p_B})\right] \\
&= \mathbb{P}\left[\delta^\top \mathcal{N}(\delta, \sigma^2 I) \geq \sigma \|\delta\|_2 \Phi^{-1}(\overline{p_B})\right] \\
&= \mathbb{P}\left[\delta^\top \mathcal{N}(0, \sigma^2 I) + \|\delta\|_2^2 \geq \sigma \|\delta\|_2 \Phi^{-1}(\overline{p_B})\right] \\
&= \mathbb{P}\left[\sigma \delta^\top \mathcal{N}(0, I) + \|\delta\|_2^2 \geq \sigma \|\delta\|_2 \Phi^{-1}(\overline{p_B})\right] \\
&= \mathbb{P}\left[\sigma \|\delta\|_2 \mathcal{N}(0, 1) + \|\delta\|_2^2 \geq \sigma \|\delta\|_2 \Phi^{-1}(\overline{p_B})\right] \\
&= \mathbb{P}\left[\sigma \|\delta\|_2 \mathcal{N}(0, 1) \geq \sigma \|\delta\|_2 \Phi^{-1}(\overline{p_B}) - \|\delta\|_2^2\right] \\
&= \mathbb{P}\left[\mathcal{N}(0, 1) \geq \Phi^{-1}(\overline{p_B}) - \frac{\|\delta\|_2}{\sigma}\right] \\
&= \Phi\left(\Phi^{-1}(\overline{p_B} - \frac{\|\delta\|_2}{\sigma})\right).
\end{aligned} \tag{3.12}$$

Hence, by combining (3.11) and (3.12), we derive

$$\mathbb{P}[Y \in S_A] > \mathbb{P}[Y \in S_B] \iff \|\delta\|_2 < \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})\right). \tag{3.13}$$

Noting that the right hand side coincides with the definition of the certified 2-norm radius of robustness r, we obtain the statement. ■

Before we prove that this bound is tight, we want to mention an alternative view on randomized smoothing from Appendix A in [78] that yields another independent proof of Theorem 3.8: There, the smoothed soft classifier $F_\sigma(\theta, \cdot)$ is understood as the Weierstrass transform of the original soft classifier $F(\theta, \cdot)$. Then, Lipschitz continuity of $F_\sigma(\theta, \cdot)_c$ and its Gaussian quantiles $\Phi^{-1}(F(\theta, \cdot)_c)$ is established in Lemmas 1 and 2, respectively. These results are utilized to directly deduce the certified 2-norm radius by the Lipschitz continuity of $\Phi^{-1}(F_\sigma(\theta, \cdot)_c)$ and the confidence quantile gap $\Phi^{-1}(p_A) - \Phi^{-1}(p_B)$. In contrast to the proof in [16], this proof fosters the intuition that smoothing at query time is another technique to stabilize the DNN prediction, here in terms of Lipschitz continuity.

Finally, we show that there is no larger certified 2-norm radius than the one from Theorem 3.8.

Theorem 3.12 (Tightness of certified 2-norm robustness bound, cf. [16] Thm. 2)**.** *In the setting of Theorem 3.8, suppose $\underline{p_A} + \overline{p_B} \leq 1$ to hold. Then, we have*

$$\forall \delta \notin \overline{\mathcal{B}}_2^r \ \exists f : \Theta \times \mathbb{R}^n \to \mathcal{C} \quad \textit{s.t.} \quad (3.1) \quad \textit{and} \quad f_s(\theta, x + \delta) \neq c_A. \tag{3.14}$$

Proof. We inherit the notation from the proof of Theorem 3.8. Now, we construct the hard classifier $\widetilde{f} : \mathbb{R}^n \to \mathcal{C}$ by setting

$$\widetilde{f}(x) \begin{cases} := c_A & \text{if } x \in S_A, \\ := c_B & \text{if } x \in S_B, \\ \in \mathcal{C} \setminus \{c_A, c_B\} & \text{else.} \end{cases}$$

Provided that $\underline{p_A} + \overline{p_B} \leq 1$, the assumption $\underline{p_A} \geq \overline{p_B}$ implies $1 - \overline{p_B} \geq \underline{p_A}$ and therefore, by monotonicity of the CDF, also $\Phi^{-1}(1 - \overline{p_B}) \geq \Phi^{-1}(\underline{p_A})$. Consequently S_A and S_B from (3.9) are disjoint, making $\widetilde{f}$ well-defined. Additionally, it satisfies the outer inequalities of (3.1) with equality, due to

$$\mathbb{P}\left[\widetilde{f}(x + \Delta) = c_A\right] = \mathbb{P}[X \in S_A] = \underline{p_A}$$

and

$$\mathbb{P}[\widetilde{f}(x + \Delta) = c_B] = \mathbb{P}[X \in S_B] = \overline{p_B}.$$

From the derivation of (3.13), we know that our assumption $\|\delta\|_2 > r$ is equivalent to $\mathbb{P}[Y \in S_A] < \mathbb{P}[Y \in S_B]$. Plugging in the definitions of the RVs and sets, this translates to

$$\mathbb{P}\left[\widetilde{f}(x + \delta + \Delta) = c_A\right] < \mathbb{P}\left[\widetilde{f}(x + \delta + \Delta) = c_B\right].$$

In terms of the corresponding smoothed classifier $\widetilde{f}_\sigma$, this constitutes a misclassification $\widetilde{f}_\sigma(x + \delta) \neq c_A$ which shows the asserted tightness property. ■

Remark 3.13. The assumption $\underline{p_A} + \overline{p_B} \leq 1$ is mild. Having some $\underline{p_A}$ (and potentially a $\overline{p_B}$) valid for (3.1), one can always obtain a (tighter) $\overline{p_B}$ via $\overline{p_B} := 1 - \underline{p_A}$ that is also feasible for (3.1). In fact,

$$\begin{aligned} \max_{c_B \in \mathcal{C} \setminus \{c_A\}} \mathbb{P}[f(x + \Delta) = c_B] &\leq \sum_{c_B \in \mathcal{C} \setminus \{c_A\}} \mathbb{P}[f(x + \delta) = c_B] \\ &= 1 - \mathbb{P}[f(x + \Delta) = c_A] \\ &\leq 1 - \underline{p_A} \\ &=: \overline{p_B} \end{aligned}$$

is a valid choice for $\overline{p_B}$ whenever $\underline{p_A}$ was feasible.

To summarize, Theorems 3.8 and 3.12 establish a direct relationship between Gaussian smoothing at query time and 2-norm robustness in the sense that the set of provably harmless perturbations is precisely a 2-norm ball without additional assumptions of the DNN. In practice, the base classifier F should be trained with adversarial examples and Gaussian noise beforehand via Algorithm 3 to make sufficiently good predictions on noisy images in order to increase the spread between p_A and p_B.

4 Numerical Tests

The final chapter of this work presents the numerical experiments that we conducted to derive Contributions 4.1, 4.2, 4.3 and 4.4. In each section, we focus on showcasing the most conclusive statistics for the respective claim. Nevertheless, the `Python` Code is available in the supplementary material and `GitHub` repository for replication of all our results. The first three contributions are based upon our own `Python` implementation, while the last contribution largely employs the code from [92, 93] that itself leverages the `Py-BOBYQA` [13] and `robustness` [23] packages. In all our presented experiments, we only attack unseen images that have been classified correctly in their natural version.

4.1 Data Sets

Our computations focus on two of the most frequently used publicly available data sets, namely ImageNet and CIFAR-10. We introduce them by providing some basic information.

4.1.1 ImageNet

ImageNet [20] is a large image database widely used for computer vision tasks. It contains more than 14 million hand-annotated high-resolution images which are assigned to $20,000$ classes. Since 2010, the ImageNet project hosts the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [76] in which numerous object detection and image classification algorithms compete. The different entries are compared on the basis of a subset of the ImageNet data set that has received minor updates until 2017. It contains more than 1.2 million training images, $100,000$ test images and $50,000$ validation images from a selection of $1,000$ classes. This subset and the results achieved on it in the competition and elsewhere represent a quasi-standard baseline for various computer vision tasks. Our experiments are performed on the validation images from the 2012 ILSVRC, which will be referred to as 'ImageNet' in the following. As common practice, we resize and center-crop the images to a resolution of 224×224 pixels before usage. Examples of the resulting pictures can be observed in Figures 1.1, 1.3, 4.5 and 4.8.

4.1.2 CIFAR-10

The CIFAR-10 data set [41], named after the 'Canadian Institute for Advanced Research', is a subset of the '80 Million Tiny Images' data set [88] that contains over 79 million labelled 32×32 low-resolution images within more than $75,000$ classes. CIFAR-10 consists of ten classes with $5,000$ training and $1,000$ test images each. Our tests are performed on random samples of the $10,000$ test images. Figure 4.1 gives an overview of the ten CIFAR-10 classes along with some exemplary samples and Figure 1.2 showcased an adversarial example classified as 'ship' of an image originally labelled as 'airplane'.

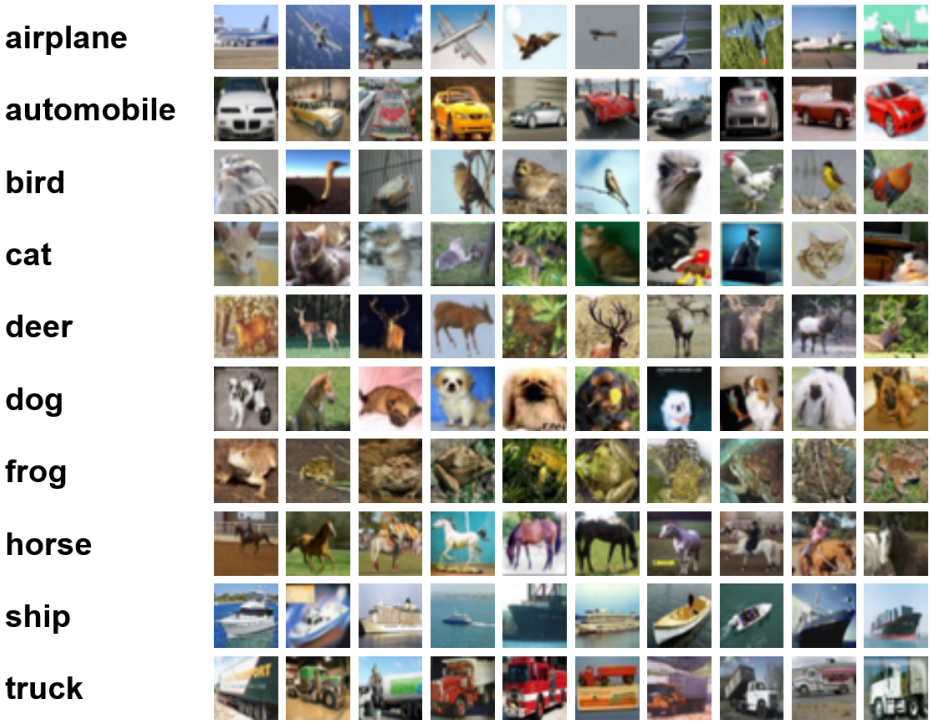


Figure 4.1: All ten classes of the CIFAR-10 data set with a selection of images. Image from [40].

4.2 p-Norm Rescaling and Zero True-Label-Confidence

This section derives and presents our first two Contributions 4.1 and 4.2. We propose two transformation functions T and $\overline{T}$ that can be used to translate perturbation radii between different p-norms such that they induce attacks of comparable strength w.r.t. the top-1 and top-5 accuracies as well as the CEL. The top-k accuracy denotes the proportion of images where the true label c is among the k predictions c' with highest confidence $p_{c'}$.

Additionally, we trialed different step schemes in PGD attacks and discovered that harmonically declining step lengths can reduce the true-label-confidence p_c to practically zero, i.e. below machine precision. To the best of our knowledge, this is unprecedented in the literature as it mainly focused on constant step sizes up to now, cf. Table 2.1.

4.2.1 Transformation Function

The first transformation function scales the p-norm radii according to the Lebesgue volume of the resulting balls $\overline{\mathcal{B}}_p^{\varepsilon_p}$. Intuitively speaking, this should offer adversaries with different p-norms the same amount of perturbation options even though they are arranged in different shapes. To this end, we recall the general formula for the Lebesgue volume of a p-norm ball of radius ε_p in $\mathbb{R}^n$. Going back to Dirichlet and Liouville

(cf. [1] Section 1.8), an inductive proof shows

$$\mathcal{L}\left(\overline{\mathcal{B}}_p^{\varepsilon_p}\right) = \frac{\left(2\Gamma\left(\frac{1}{p}+1\right)\varepsilon_p\right)^n}{\Gamma\left(\frac{n}{p}+1\right)} \quad \forall p \in [1,\infty],$$

where Γ denotes the gamma function. Plugging in the considered values of p and simplifying via $\Gamma(m+1) = m!$, we obtain

$$\mathcal{L}\left(\overline{\mathcal{B}}_p^{\varepsilon_p}\right) = \begin{cases} \frac{(2\varepsilon_p)^n}{n!} & \text{if } p = 1, \\ \frac{(\sqrt{\pi}\varepsilon_p)^n}{\Gamma(n/2+1)} = \frac{(\sqrt{\pi}\varepsilon_p)^n}{(n/2)!} & \text{if } p = 2, \\ (2\varepsilon_p)^n & \text{if } p = \infty, \end{cases}$$

where we assumed n to be even in the case of $p = 2$. This holds true for CIFAR-10 and appropriately cropped ImageNet samples. Hence, to determine the radii ε_1 and ε_2 such that $\mathcal{L}\left(\overline{\mathcal{B}}_p^{\varepsilon_p}\right) = \mathcal{L}\left(\overline{\mathcal{B}}_\infty^{\varepsilon_\infty}\right)$, we first compute

$$\begin{aligned} &\frac{(2\varepsilon_1)^n}{n!} = (2\varepsilon_\infty)^n \\ \iff\ &\varepsilon_1 = \sqrt[n]{n!}\varepsilon_\infty \sim \frac{n}{e}\sqrt[2n]{2\pi n}\varepsilon_\infty \\ \implies\ &\varepsilon_1 \approx \lambda_\infty^1 \varepsilon_\infty := \begin{cases} 55378.6875 \cdot \varepsilon_\infty & \text{if } n = 150,528 \text{ (ImageNet)}, \\ 1131.9422 \cdot \varepsilon_\infty & \text{if } n = 3,072 \quad \text{(CIFAR-10)}. \end{cases} \end{aligned} \tag{4.1}$$

Since $n!$ is too large to be computed and stored explicitly for the above n, we used Stirling's formula

$$n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}.$$

By the inequalities in [74] that bound

$$n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n} \cdot e^{r_n} \quad \text{with} \quad \frac{1}{12n+1} < r_n < \frac{1}{12n} \quad \forall n \in \mathbb{N},$$

the relative approximation error is well below 10^{-5} (10^{-7}) for CIFAR-10 (ImageNet) since $n > 10^4$ ($n > 10^6$) and therefore $r_n < 10^{-5}$ ($r_n < 10^{-7}$) and $\exp(r_n) = 1 + r_n + \mathcal{O}(r_n^2)$ by Taylor's Theorem. An analogous calculation for the 2-norm radius shows

$$\begin{aligned} &\frac{(\sqrt{\pi}\varepsilon_2)^n}{(n/2)!} = (2\varepsilon_\infty)^n \\ \iff\ &\varepsilon_2 = \sqrt[n]{(n/2)!}\frac{2}{\sqrt{\pi}}\varepsilon_\infty \sim \sqrt{\frac{2n}{e\pi}}\sqrt[2n]{\pi n} \cdot \varepsilon_\infty \\ \implies\ &\varepsilon_2 \approx \lambda_\infty^2 \varepsilon_\infty := \begin{cases} 187.7675 \cdot \varepsilon_\infty & \text{if } n = 150,528 \text{ (ImageNet)}, \\ 26.8628 \cdot \varepsilon_\infty & \text{if } n = 3,072 \quad \text{(CIFAR-10)}. \end{cases} \end{aligned} \tag{4.2}$$

Combining these numbers λ_∞^1 and λ_∞^2, we deduce the missing rescaling factors as depicted in Figure 4.2.

The black arc weights originate from (4.1) and (4.2) and the gray ones are derived either by using symmetry

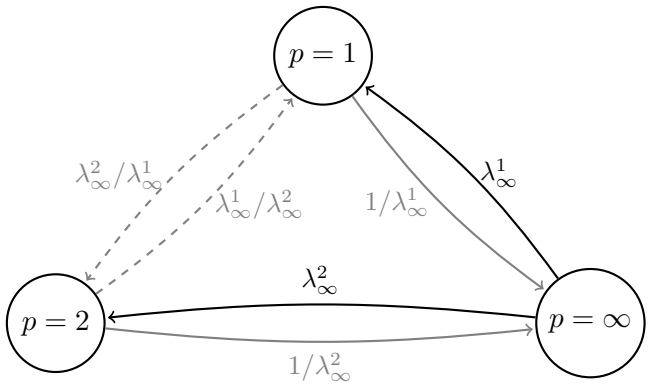


Figure 4.2: Rescaling factors for $p \in \{1, 2, \infty\}$ to equalize the p-norm balls' Lebesgue volume.

(solid) or transitivity (dashed) of the volume equality. We formalize this graph into a norm rescaling function

$$T_{p_1}^{p_2} : \mathbb{R}^+ \to \mathbb{R}^+, \quad \varepsilon_{p_1} \mapsto \varepsilon_{p_2} := \varepsilon_{p_1} \cdot \begin{cases} 1 & \text{if } p_1 = p_2, \\ \lambda^2_\infty/\lambda^1_\infty & \text{if } (p_1, p_2) = (1, 2), \\ 1/\lambda^1_\infty & \text{if } (p_1, p_2) = (1, \infty), \\ \lambda^1_\infty/\lambda^2_\infty & \text{if } (p_1, p_2) = (2, 1), \\ 1/\lambda^2_\infty & \text{if } (p_1, p_2) = (2, \infty), \\ \lambda^1_\infty & \text{if } (p_1, p_2) = (\infty, 1), \\ \lambda^2_\infty & \text{if } (p_1, p_2) = (\infty, 2) \end{cases} \tag{4.3}$$

for $p \in \{1, 2, \infty\}$. For comparison, we use $\|\cdot\|_{p_2} \leq n^{1/p_2 - 1/p_1} \|\cdot\|_{p_1}$ for $p_1, p_2 \in [1, \infty]$, $p_2 \leq p_1$ to define a second transformation function

$$\overline{T}_{p_1}^{p_2} : \mathbb{R}^+ \to \mathbb{R}^+, \quad \varepsilon_{p_1} \mapsto \varepsilon_{p_2} := \varepsilon_{p_1} \cdot \begin{cases} 1 & \text{if } p_1 \leq p_2, \\ n^{\frac{1}{p_2} - \frac{1}{p_1}} & \text{if } p_1 > p_2, \end{cases} \tag{4.4}$$

that insures ε_{p_2} to be the smallest radius with $\overline{\mathcal{B}}_{p_1}^{\varepsilon_{p_1}} \subseteq \overline{\mathcal{B}}_{p_2}^{\varepsilon_{p_2}}$. The motivation behind this transformation is to have a baseline for T to be compared with. Theoretically, (4.4) should make the p_2-norm bounded adversary at least as capable the p_1-norm bounded one due to this inclusion. Figure 4.3 visualizes both transformations for unit balls in $\mathbb{R}^2$.

4.2.2 Hypothesis and Metrics

We expect that scaling the perturbation radii to equal Lebesgue volume via T from (4.3) will yield similarly strong p-norm bounded attacks for $p \in \{1, 2, \infty\}$ since the adversary has the same exploitable space around the clean image to explore. However, the balls are differently shaped and therefore allow for perturbations with different characteristics. For instance, the 1- and 2-norm would allow one pixel to be adjusted massively when keeping the others sufficiently natural. The ∞-norm does not allow such trade-offs and bounds each of the n color channel values separately, cf. Figure 4.3.

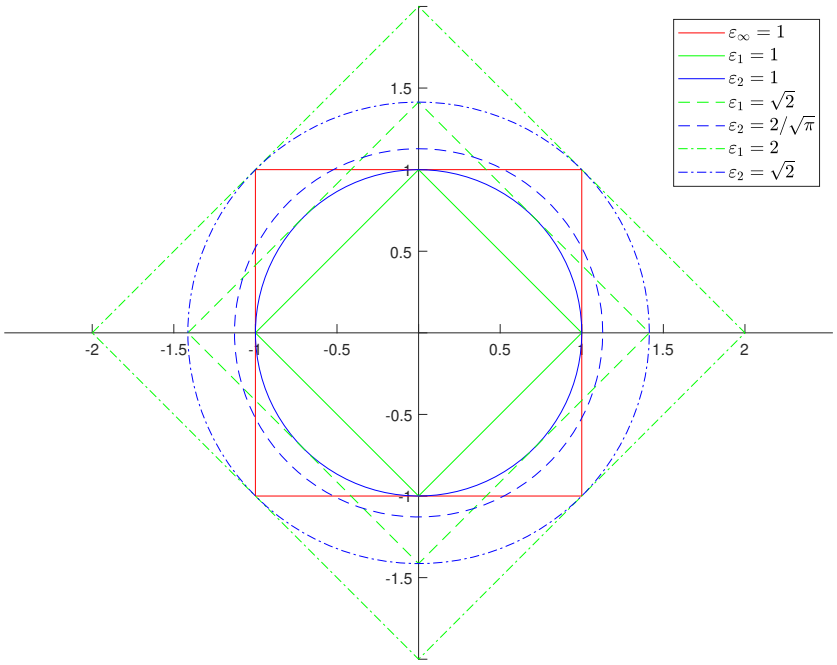


Figure 4.3: Unit p-norm spheres (solid) for $p \in \{1, 2, \infty\}$, T_∞^p-scaled spheres (dashed) and $\overline{T}_\infty^p$-scaled spheres (dash-dotted) for $p \in \{1, 2\}$ in $n = 2$ dimensions.

As a baseline, we execute the entire process analogously for the transformation $\overline{T}$ from (4.4) which creates nested p-norm balls. Here, we hypothesize that attacks with a perturbation set $\overline{\mathcal{B}}_{p_1}^{\varepsilon_{p_1}} \subseteq \overline{\mathcal{B}}_{p_2}^{\varepsilon_{p_2}}$, which is a subset of the one from a different attack, to be weaker than its counterpart.

Our metrics for measuring the attacks strengths are the top-1 and top-5 accuracy as well as the true-label-confidence p_c or equivalently the CEL, i.e. $-\log(p_c)$. To make sure that the scaled norms also preserve visual faithfulness to a comparable degree, we will additionally examine the average p-norm distortions for all $p \in \{1, 2, \infty\}$, not only the one imposed on the attack, and show some of the perturbed samples.

4.2.3 Setup and Implementation

We perform our experiments on the ImageNet validation data set and use a first-order white-box attack similar to Algorithm 1. However, we modified the step lengths and the randomization method in our implementation presented in Algorithm 4: In contrast to Algorithm 1, and therefore to most of the existing literature [42, 52, 101, 105, 37, 100, 3, 80, 89, 60, 16, 95, 30, 43], we use harmonically and geometrically declining step lengths, instead of a constant ones, when perturbing the images. We expect the smaller steps towards the end to be beneficial for finding tiny regions of high loss. Furthermore, in addition to the deterministic steps, we consider random perturbations at each iteration, instead of only once at the beginning with the hope to circumvent obfuscated gradients more effectively. However, we found these

modifications to have little to no effect on relevant metrics like the top-1 and top-5 accuracy on ImageNet with $\varepsilon_\infty = 3/255$ for the naturally trained `PyTorch` DNN and the adversarially trained ones from MadryLab [23] and LocusLab [95]. In fact, there was only one setting in which the performance difference between all six step schemes w.r.t. one of these metrics exceeded 0.2%pt (percentage points): Our declining step schemes slightly outperformed the established constant one in terms of top-5 accuracy reduction when attacking the naturally trained DNN, cf. Figure 4.4.

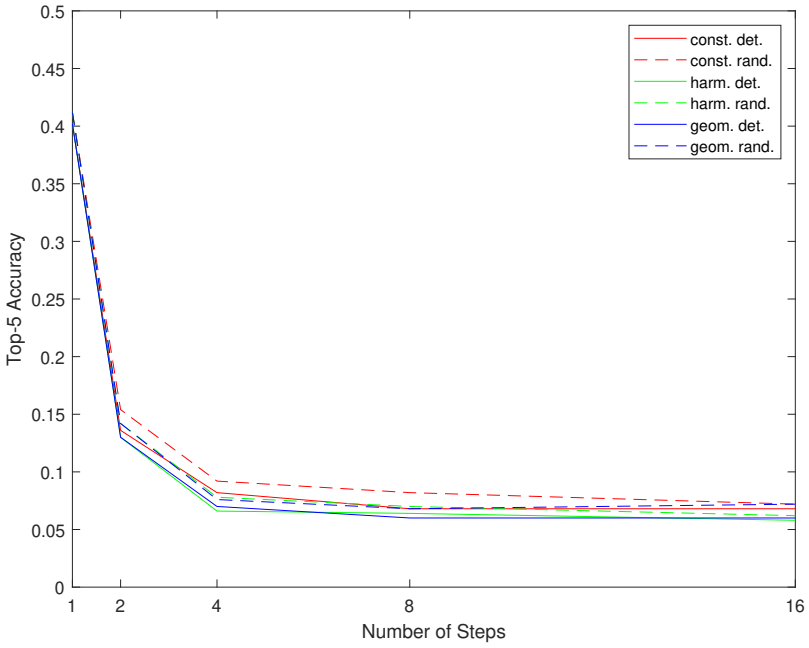


Figure 4.4: Comparison of our proposed decaying step schemes with a constant one using step length $\omega = \varepsilon_\infty / S$ from the literature [89] when attacking the naturally pretrained `PyTorch` DNN on 500 images from the ImageNet validation data set with $\varepsilon_\infty = 3/255$.

One exception to this insensitivity to the step scheme arises when considering the true-label-confidence p_c or equivalently the CEL as success metric: When combining the runs for `seed = 0` and `seed = 42`, we find seven images that got perturbed by the deterministic harmonic step scheme so successfully that p_c was below machine precision of `Python` floats despite the natural images being correctly classified with up to $p_c \geq 95\%$ confidence. In six out of these instances, the 16 step gradient attack was necessary and once eight steps were enough. In analogy, the randomized harmonic steps reached that two times. In the respective preceding attack with half that number of steps, the true-label-confidence was already below 10^{-34}. In our experiments, the constant and geometric step schemes did not reach such levels.

We suspect the geometric step lengths to be too short in the end to fully explore the vulnerable regions of the loss landscape and the constant step sizes to be too long in the end causing oscillations around the zero confidence area rather than entering it. An in-depth investigation of this is pending and could be subject of future research. To the best of our knowledge, there are no publications reporting true-label-confidences of similar magnitude. Hence, we formulate our first numerical contribution.

Contribution 4.1 (Zero True-Label-Confidence)**.** *PGD attacks on the naturally pretrained `PyTorch` ResNet50 DNN with $\varepsilon_\infty = 3/255$ and our harmonically declining step size are able to perturb ImageNet adversarial*

examples with true-level-confidences of several magnitudes below 10^{-34}*. This causes* `Python` *floats to be incapable from distinguishing them from zero and the CEL to be undefined. Consequently, the adversarial problem* (ADV-UNT) *is unbounded from above for some images* x *and small perturbation radii* ε_∞*.*

Algorithm 4 – Perturb

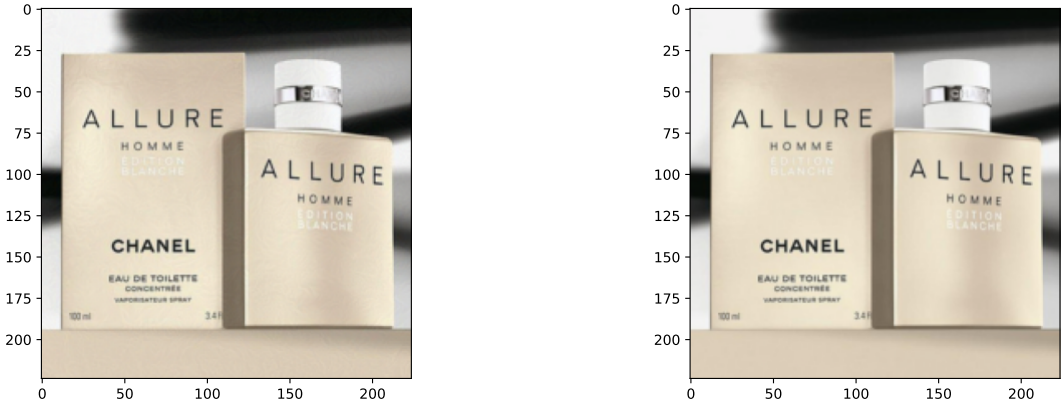
Input: norm exponent $p \geq 1$, radius $\varepsilon_p > 0$, number of samples *num_samples* $\in \mathbb{N}$, data set *data_set* $\in \{\textit{ImageNet}, \textit{CIFAR-10}\}$, decay type *decay* $\in \{\textit{harmonic}, \textit{geometric}\}$, randomization *rand* $\in \{\textit{true}, \textit{false}\}$

Output: adversarial examples $x^k_{\textit{gauss}}$, $x^k_{\textit{unif}}$, $x^k_{\textit{bound}}$, $x^k_{\textit{num\_steps}}$ $\forall$*num_steps* $\in \{1, 2, 4, 8, 16\}$, $k \in [\textit{num\_samples}]$

for $k = 1, \ldots,$ *num_samples* **do**
  # get random pair
  $(x^k, c^k) := \mathcal{U}[\textit{data\_set}]$
  $x^k_{\textit{clean}} := x^k$
  # execute gradient-free attacks
  $x^k_{\textit{gauss}} := \Pi^\infty_{\mathcal{X}} \left(\mathcal{N} \left(x^k, (\varepsilon_p/2)^2 \right) \right)$
  $x^k_{\textit{unif}} := \Pi^\infty_{\mathcal{X}} \left(\mathcal{U} \left[\overline{B}^{\varepsilon_p}_p(x^k) \right] \right)$
  $x^k_{\textit{bound}} := \Pi^\infty_{\mathcal{X}} \left(\mathcal{U} \left[\mathcal{S}^{\varepsilon_p}_p(x^k) \right] \right)$
  # execute first-order white-box attacks
  for *num_steps* $\in \{1, 2, 4, 8, 16\}$ **do**
    # compute step sizes
    if *decay* = *geometric* **then**
      $\textit{sum} := \Sigma_{s=0}^{\textit{num\_steps}-1} 2^{-s} = 2(1 - 0.5^{\textit{num\_steps}})$
      $\omega_s := \varepsilon_p 2^{-s} / \textit{sum} \quad \forall s \in [\textit{num\_steps}]$
    else if *decay* = *harmonic* **then**
      $\textit{sum} := \Sigma_{s=1}^{\textit{num\_steps}} s^{-1}$
      $\omega_s := \varepsilon_p s^{-1} / \textit{sum} \quad \forall s \in [\textit{num\_steps}]$
    end if
    # perform gradient ascent
    for $s = 1, \ldots,$ *num_steps* **do**
      $g_s := \nabla_x L^{\text{CE}}(F(\theta, x^k_s), c^k)$
      if $p = \infty$ **then**
        $g_s := \text{sign}(g_s)$
      else
        $g_s := g_s / \|g_s\|_p$
      end if
      if *rand* = *true* **then**
        $x^k_s := x^k_s + 0.1 \cdot \omega_s \mathcal{U} \left[\mathcal{S}^{\varepsilon_p}_p \right] + 0.9 \cdot \omega_s g_s$
      else
        $x^k_s := x^k_s + \omega_s g_s$
      end if
    end for
    $x^k_{\textit{num\_steps}} := \Pi^\infty_{\mathcal{X}} \left(x^k_{\textit{num\_steps}} \right)$
  end for
end for

Since there are no apparent downsides and possibly small upsides to our method, we decided to employ our step schemes in the numerical experiments conducted with Algorithm 4. It was executed on `seed = 42` and also `seed = 0` for validation purposes. The results did not differ significantly and in the following we will always consider the results calculated for `seed = 42`.

The observations leading to Contribution 4.1 were our motivation to cap the number of steps in Algorithm 4 at 16. On the one hand, we wanted the best attack to be as strong as computationally practical in order to give a good approximation to a solution of the inner problem (ADV-UNT) that Danskin's Theorem demands. On the other hand, more gradient steps further increased the number of samples with $p_c = 0$ in machine precision making the loss $L^{CE}(F(\theta, x), c)$ and its gradients undefined. Figure 1.1 already presented one such example on the `seed = 21` with $\varepsilon_\infty = 0.02$. Figure 4.5 presents another zero true-label-confidence example from our productive framework, i.e. on `seed = 42` for $\varepsilon_\infty = 3/255$.



(a) clean, $\varepsilon_\infty = 0$ (b) adversarial, $\varepsilon_\infty = 3/255$

Figure 4.5: Picture of a perfume (label 711) from the ImageNet validation data set. Clean image (a) correctly classified with confidence of 61.34% by naturally pretrained `PyTorch` DNN and adversarial sample (b) with 0.00% confidence for 'perfume' and 99.84% confidence for 'paper towel' (label 700).

We normalize the proposed step lengths of the first-order attacks to insure $x^k_{\text{num\_steps}} \in \overline{B}_p^{\varepsilon_p}(x^k)$ via

$$\|x^k - x^k_{\text{num\_steps}}\|_p = \left\| \sum_{s \in [num\_steps]} \omega_s g_s \right\|_p \leq \sum_{s \in [num\_steps]} \omega_s \|g_s\|_p = \varepsilon_p$$

without projections after each step that might disturb the trajectory of the gradient ascent method. The gradient-free attacks suffice that by construction.

Note that all attacks take place in the linearly transformed domain of preprocessed images, i.e. each color channel of the attacked images is normalized by the expectation and variance of the training data set before being perturbed and queried to the DNN. This is a common procedure to improve the training success and the here attacked DNNs were pretrained on such data. We calculated the inverse transformation afterwards for presentation and to compare them to the originals. The perturbation radii ε_p still relate to the original image representation in $\mathcal{X}$ of the 224×224 center-cropped ImageNet samples.

DNN				Attack ε_1			Attack ε_2			Attack ε_∞	
Defense	Train on	p	Arch.	$3/255$	651.5	1771	$3/255$	2.209	4.564	$3/255$	$6/255$
Natural	Clean		ResNet50	500	500	500	500	500	500	500	500
MadryLab [23]	PGD	$2, \infty$	ResNet50					500		500	500
LocusLab [95]	R-FGSM	∞	ResNet50							500	500

Table 4.1: Number of attacked samples for each pair of `decay` and `rand` from the ImageNet validation data set in the experiments regarding Danskin's Theorem and norm rescaling. Baseline is $\varepsilon_\infty = 3/255 \approx 0.012$ with ε_p for $p \in \{1, 2\}$ chosen via $\overline{T}_p^\infty$ (left), T_∞^p (middle) and $\overline{T}_\infty^p$ (right).

DNN				Attack ε_2	Attack ε_∞	
Defense	Train on	p	Arch.	2.209	$3/255$	$6/255$
MadryLab [23]	Multi-step	$2, \infty$	ResNet50	3.0	4/255	8/255
LocusLab [95]	R-FGSM	∞	ResNet50		2.5/255	5/255

Table 4.2: Training ε_p regarding the matching p from the 'p'-column for each attack configuration of the experiments regarding Danskin's Theorem and norm rescaling on the ImageNet validation data set. Baseline is $\varepsilon_\infty = 3/255 \approx 0.012$ with attack $\varepsilon_2 = T_\infty^2(3/255) \approx 2.209$.

Table 4.1 presents the number of samples that we tested in Algorithm 4 for each perturbation radius ε_p and defense. The radii, which were used during training of each defended DNN, are listed in Table 4.2 below.

4.2.4 Results and Interpretation

Before we consider all metrics mentioned above, we note that, apart from the observation in Contribution 4.1, the results did not differ much between the harmonic and geometric step lengths, neither in the deterministic nor the randomized case. Therefore, we often only present the metrics in a selection of these cases. The other cases can be reproduced with or examined in the supplementary material.

We start by comparing the top-1 and top-5 accuracy listed in Tables 4.3 and 4.4. Here, $\overline{T}$-scaling seems to adapt the accuracy for $p \in \{1, 2\}$ more precisely to the $p = \infty$ constrained attacks than T-scaling, especially when looking at the top-5 accuracy. Nevertheless, considering that we only attacked samples that were classified correctly originally, the reduction in precision is roughly similar for both rescalings, in particular regarding the top-1 accuracy.

Next we examine the true-label-confidence through the perspective of the CEL in Table 4.5. We notice that for most quantiles the T-transformed attacks lower bound and the $\overline{T}$-transformed attacks generally upper bound the CEL experienced by the original ∞-norm bounded attacks.

Step		Avg. top-1 accuracy (%)			Avg. top-5 accuracy (%)		
Decay	Rand	$\varepsilon_\infty = 3/255$	$\varepsilon_2 = T_\infty^2(\varepsilon_\infty)$	$\varepsilon_1 = T_\infty^1(\varepsilon_\infty)$	$\varepsilon_\infty = 3/255$	$\varepsilon_2 = T_\infty^2(\varepsilon_\infty)$	$\varepsilon_1 = T_\infty^1(\varepsilon_\infty)$
harm.	false	29.66	30.97	30.69	38.86	45.83	43.03
harm.	true	29.74	31.23	30.77	39.49	47.51	44.86
geom.	false	29.66	31.00	30.74	38.89	46.57	43.34
geom	true	29.74	31.26	30.80	39.57	48.14	45.09
Avg.		29.70	31.12	30.75	39.20	47.01	44.08

Table 4.3: Average accuracies over all attacks (excluding clean & Gaussian) with T-scaling according to (4.3) on naturally trained ResNet50 on the ImageNet validation data set.

Step		Avg. top-1 accuracy (%)			Avg. top-5 accuracy (%)		
Dec.	Rnd.	$\varepsilon_\infty = 3/255$	$\varepsilon_2 = \overline{T}^2_\infty(\varepsilon_\infty)$	$\varepsilon_1 = \overline{T}^1_\infty(\varepsilon_\infty)$	$\varepsilon_\infty = 3/255$	$\varepsilon_2 = \overline{T}^2_\infty(\varepsilon_\infty)$	$\varepsilon_1 = \overline{T}^1_\infty(\varepsilon_\infty)$
harm.	false	29.66	30.54	30.26	38.86	38.86	37.06
harm.	true	29.74	30.57	30.14	39.49	39.34	37.37
geom.	false	29.66	30.51	30.26	38.89	39.06	37.14
geom.	true	29.74	30.57	30.20	39.57	39.69	37.49
Avg.		29.70	30.55	30.22	39.20	39.24	37.27

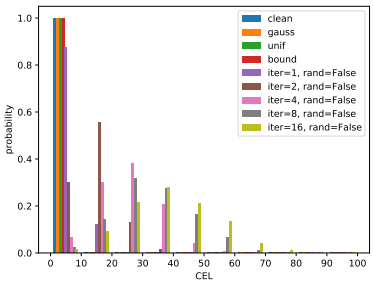
Table 4.4: Average accuracies over all attacks (excluding clean & Gaussian) with $\overline{T}$-scaling according to (4.4) on naturally trained ResNet50 on the ImageNet validation data set.

$Q(\cdot)$	CEL Quantiles				
	$\varepsilon_\infty = 3/255$	$\varepsilon_2 = T^2_\infty(\varepsilon_\infty)$	$\varepsilon_1 = T^1_\infty(\varepsilon_\infty)$	$\varepsilon_2 = \overline{T}^2_\infty(\varepsilon_\infty)$	$\varepsilon_1 = \overline{T}^1_\infty(\varepsilon_\infty)$
10%	3.26	2.00	2.35	2.71	2.85
50%	17.64	12.68	14.26	18.90	22.30
90%	39.38	30.45	32.98	42.04	49.45

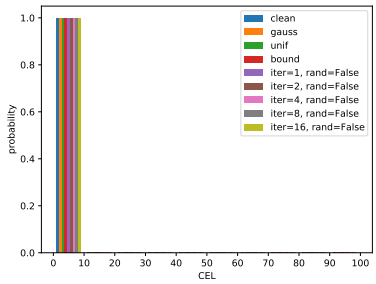
Table 4.5: Quantiles of CELs computed over union of all attacks (excluding clean & Gaussian) with all different step schemes. T-scaling applied according to (4.3) (left) and $\overline{T}$-scaling (4.4) (right) on naturally trained `PyTorch` ResNet50 on ImageNet.

This observation still holds when separating the losses of different attacks and plotting their histograms as done in Figure 4.6. Here, the right column relates to $\overline{T}$, i.e. the smallest ε_p such that $\mathcal{U}_\infty^{\varepsilon_\infty} \subset \mathcal{U}_p^{\varepsilon_p}$, while the middle column corresponds to T, i.e. choosing ε_p such that $\mathcal{L}\left(\overline{\mathcal{B}}_p^{\varepsilon_p}\right) = \mathcal{L}\left(\overline{\mathcal{B}}_\infty^{\varepsilon_\infty}\right)$. In fact, comparing (c) and (f) with (a), we again notice a tendency to slightly lower CEL values and, conversely, in (d) and (g) slightly higher losses than in the baseline case (a).

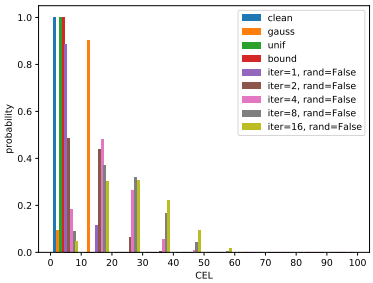
Moreover, the left column relates to the largest ε_p such that $\mathcal{U}_p^{\varepsilon_p} \subset \mathcal{U}_\infty^{\varepsilon_\infty}$, i.e. $\varepsilon_1 = \varepsilon_2 = \varepsilon_\infty$. It clearly demonstrates the need for a radius transformation when using different p-norms because all attacks in (b) and (e) were much weaker than (a) as imposing $\varepsilon_2 = 3/255$ or $\varepsilon_1 = 3/255$ is much more restrictive on the attacker than $\varepsilon_\infty = 3/255$.



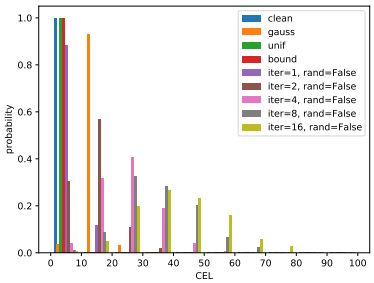
(a) $\varepsilon_\infty = 3/255 \approx 0.012$



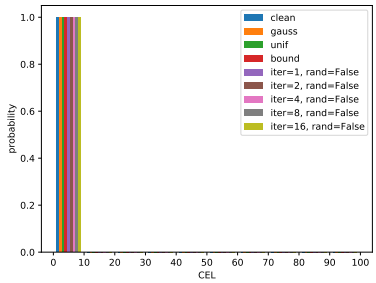
(b) $\varepsilon_2 = \overline{T}_2^\infty(\varepsilon_\infty) \approx 0.012$



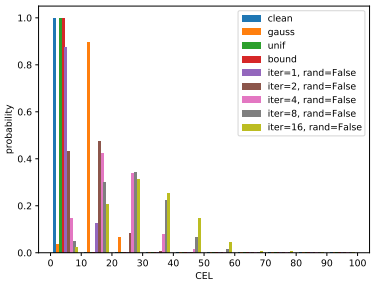
(c) $\varepsilon_2 = T_\infty^2(\varepsilon_\infty) \approx 2.209$



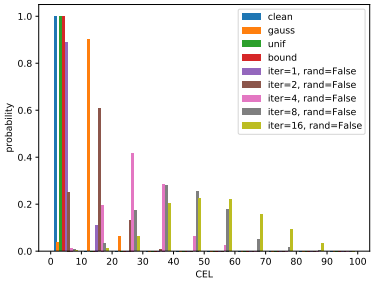
(d) $\varepsilon_2 = \overline{T}_\infty^2(\varepsilon_\infty) \approx 4.564$



(e) $\varepsilon_1 = \overline{T}_2^\infty(\varepsilon_\infty) \approx 0.012$



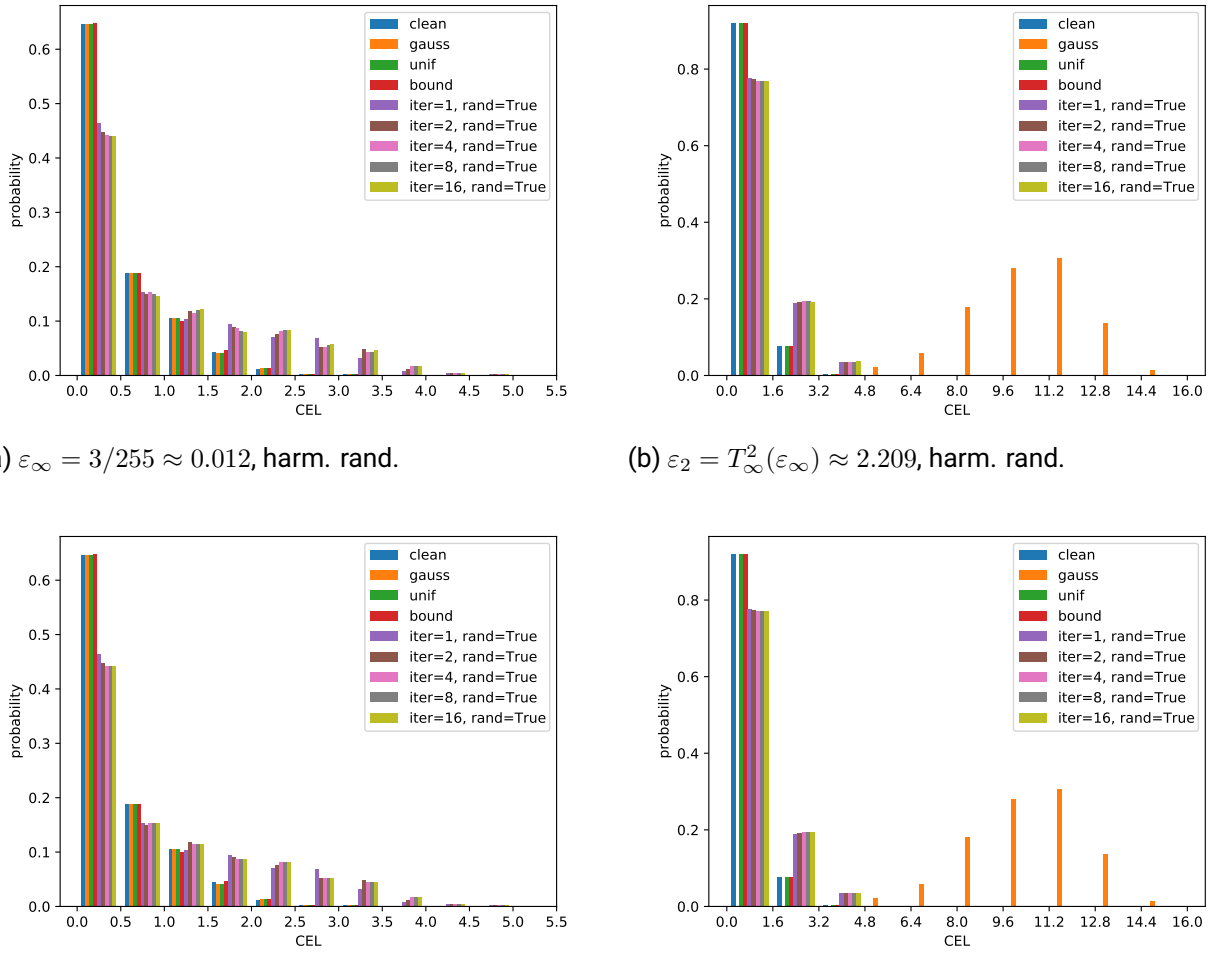
(f) $\varepsilon_1 = T_\infty^1(\varepsilon_\infty) \approx 651.5$



(g) $\varepsilon_1 = \overline{T}_\infty^1(\varepsilon_\infty) \approx 1171$

Figure 4.6: CEL-histograms with bucket size 10 for p-norm bounded white-box attacks on naturally trained ResNet50 DNN with deterministic harmonic step size for $p \in \{1, 2, \infty\}$ with $\overline{T}$-rescaling according to (4.4) (left & right) and T-scaling according to (4.3) (middle) on ImageNet.

The appropriateness of both rescaling functions generalizes to other step lengths, randomizations and adversarially trained DNNs like the MadryLab net [23]. Figure 4.7 shows this at the example of T-transformation and randomized step schemes for the same collection of attacks from Algorithm 4.



(a) $\varepsilon_\infty = 3/255 \approx 0.012$, harm. rand.

(b) $\varepsilon_2 = T_\infty^2(\varepsilon_\infty) \approx 2.209$, harm. rand.

(c) $\varepsilon_\infty = 3/255 \approx 0.012$, geom. rand.

(d) $\varepsilon_2 = T_\infty^2(\varepsilon_\infty) \approx 2.209$, geom. rand.

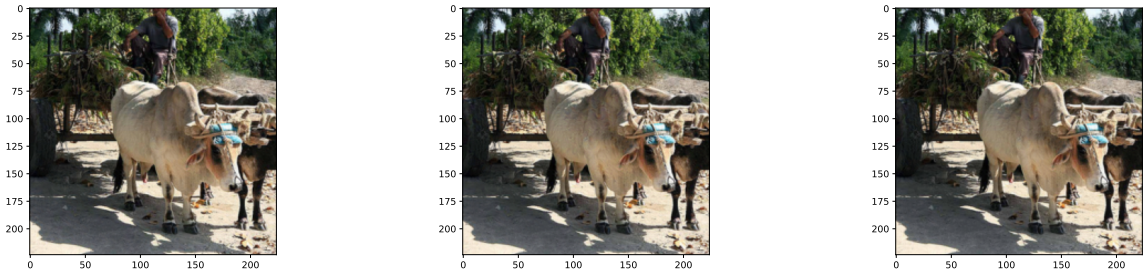
Figure 4.7: CEL-histograms for $p \in \{2, \infty\}$-norm bounded white-box attacks on adversarially trained ResNet50 DNN from MadryLab [23] with randomized step schemes and T-rescaling according to (4.3) on ImageNet. Mind the different abscissas.

Note that this time, unlike to Figure 4.6, all gradient-based attacks are similarly strong and outperform the gradient-free, randomized attacks by a margin that is much smaller than before. Overall, neglecting the Gaussian attack, the CEL is about 12-fold lower compared to the naturally trained case. This indicates the effectiveness of the adversarial training. The Gaussian attack has to be interpreted with caution as it is not bounded by some ε_p-ball, but a clipped $\mathcal{N}(0, \varepsilon_p/2)$-distribution around the clean sample. Given the applied T-scaling, this causes Algorithm 4 to compute a clipped $\mathcal{N}(x, 1.105)$-distribution on $\mathcal{X} = [0,1]^n$ which resulted in unrecognizable pictures that fail to meet any of our intuitive definitions (a), (b) and (c) from the introduction. However, focusing on the other attacks, we again observe that T-scaling, and similarly $\overline{T}$-scaling, assimilate the 2-norm bounded attacks to the ∞-norm bounded ones for different step length decays as well as randomization schemes.

Up to now, we focused on the attacker's misclassification goal. Shifting our attention to the constraint of visual similarity, we note that different p-norms in attacks induce different p-norm distortions, cf. Table

Attack Radius	Distortion $[Q(5\%), Q(50\%), Q(95\%)]$		
	$\|x - \widetilde{x}\|_\infty$	$\|x - \widetilde{x}\|_2$	$\|x - \widetilde{x}\|_1$
$\varepsilon_\infty = 3/255 \approx \mathbf{0.0118}$	**[0.0118, 0.0118, 0.0118]**	[2.7013, 2.9219, 4.5642]	[888.07, 966.66, 1770.80]
$\varepsilon_2 = T_\infty^2(\varepsilon_\infty) \approx \mathbf{2.2090}$	[0.0405, 0.0640, 0.132]	**[1.3218, 1.4732, 2.2087]**	[302.06, 380.19, 557.52]
$\varepsilon_1 = T_\infty^1(\varepsilon_\infty) \approx \mathbf{651.51}$	[0.0452, 0.0789, 0.1860]	[1.5436, 1.8396, 2.8617]	**[403.62, 448.68, 651.38]**

Table 4.6: 90% confidence interval and median of p-norm distortions for each attacker model. Quantiles are computed over the union of all different step schemes, i.e. deterministic and randomized, harmonic and geometric as well as all different number of steps. The radii for $p \in \{1, 2\}$ results from T-scaling according to (4.3) with $\varepsilon_\infty = 3/255$ on naturally trained ResNet50 for ImageNet.



(a) $p = \infty$, $[0.0118, 2.2528, 722.15]$ (b) $p = 2$, $[0.0543, 1.4561, 418.49]$ (c) $p = 1$, $[0.0749, 1.4192, 353.64]$

Figure 4.8: Image of an oxcart (label 690) from the ImageNet validation data set perturbed by Algorithm 4 with 16 deterministic harmonically decaying steps and bounded by different p-norms according to the T-scaled constraint $\varepsilon_\infty = 3/255$. Distortions are denoted in the format $[\|x - \widetilde{x}\|_\infty, \|x - \widetilde{x}\|_2, \|x - \widetilde{x}\|_1]$.

4.6. But, apart form the ∞-norm distortions $\|x - \widetilde{x}\|_\infty$, they remain comfortably within a magnitude of the T-scaled values.

Despite the significant discrepancy in the ∞-norm distortion, most images still look almost indistinguishable from the clean sample or, at least, they do not exhibit obvious perturbation artefacts as can be seen in the instance presented in Figure 4.8. Hence, our similarity standards (a) or (b) are generally satisfied as with the original radius $\varepsilon_\infty = 3/255$.

Consequently, not only the CEL and the top-1 or top-5 accuracy are comparable, but also most distortions and the visual similarity are preserved to large extends by both norm transformation functions. This only partially confirms the initial hypothesis since the $\overline{T}$-scaling for nested p-norm balls delivered unexpectedly good and similar results to the T-scaling that homogenizes the ball volumes. Summarizing our analysis from above, we formulate our second numerical contribution.

Contribution 4.2 (p-Norm Rescaling)**.** *The proposed T- and $\overline{T}$-scaling functions from* (4.3) *and* (4.4) *provide a starting point to scale the perturbation radii on p-norm bounded adversarial attacks for different values of $p \in \{1, 2, \infty\}$ in order to equalize attack strengths and keep visual distortions comparable. Depending on the performance metric, one may be more appropriate than the other, but both give a reasonable estimate.*

4.3 Approximate Solutions in Danskin's Theorem

This section tries to bridge the gap that we sketched in Section 3.1 between the non-existent theoretical guarantees of Danskin's Theorem (Theorem 3.1) and the empirical success of standard adversarial training as in Algorithm 2.

4.3.1 Hypothesis and Metrics

Concretely, we join [80] in asking the question why adversarial training works in practice, i.e. increases adversarial robustness in (E-RO). Theoretical explanations purely based on Danskin's Theorem falter due to multiple violations of its assumptions which can be catastrophic, c.f. Example 3.3. We hypothesize that the model parameter gradient of the CEL $\nabla_\theta L^{\text{CE}}(F(\theta, x), c)$ is correlated enough between approximate solutions and the (possibly existent) exact solution(s) of problem (ADV-UNT) to yield sufficiently precise stochastic gradients for the adversarial training scheme in Algorithm 2 to decrease the CEL. We plan to test our hypothesis by measuring the angles between aforementioned gradients for adversarial examples of different strengths. By the observations leading to Contribution 4.1, we consider the 16 step gradient attack in Algorithm 4 with deterministic, harmonically decreasing step scheme to be a reasonable approximation of a true solution to (ADV-UNT), if it exists.

4.3.2 Setup and Implementation

The idea to measure the similarity of vectors via some notion of angles has been applied before: [52] uses the angle between the input gradients of the respective loss $\nabla_x L(F(\theta, x), c)$ to explain transferability of adversarial examples between DNNs, [89] compares different perturbations with it and [2] measures the linearity of loss landscapes via the gradient alignment score (GAS). The GAS is given by the average cosine similarity between input gradients of random images in a shared neighbourhood. In contrast, we compute the cosine similarity and angles between the model parameter gradients of the loss $\nabla_\theta L(F(\theta, x), c)$ in different adversarial examples according to Algorithm 5 to compare the parameter learning directions they contribute to the update g_e in adversarial training via Algorithm 2.

The presented results again originate from `seed = 42` and are validated on the alternative `seed = 0`. As previously, we use the sample sizes from Table 4.1 and the adversarial radii from Table 4.2. In fact, we focus on the ImageNet validation data set and perturbations with $\varepsilon_\infty \in \{3/255, 6/255\}$.

4.3.3 Results and Interpretation

As before, we will only present a selection of the metrics that we collected. Since the deterministic, harmonically decaying step scheme was the strongest, cf. Contribution 4.1, its perturbation is our best approximation to being optimal for (ADV-UNT). Hence, we will mainly focus our analysis on this setting. Either way, the results did not differ significantly between the step lengths and randomizations. Generally speaking, the geometric and randomized schemes yield slightly higher cosine similarities, i.e. smaller angles, between the model parameter gradients of the CEL in the adversarial examples. Similarly, the mean and median angles and cosine similarities never deviate to a degree that would change our interpretation of the results, cf. Figure 4.11.

Algorithm 5 – Similarity

Input: pairs $(x_i, c_i), (x_j, c_j) \in \mathcal{X} \times \mathcal{C}$, parameterized soft classifier $F(\theta, \cdot) : \mathcal{X} \to \mathcal{P}$

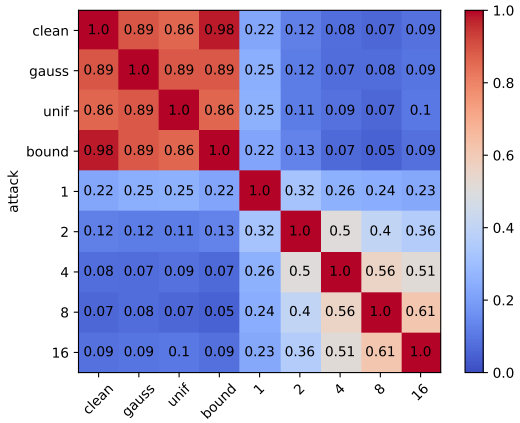
Output: cosine similarity *cos_sim*(i, j), angle in radians *rad_angle*(i, j) and angle in degrees *deg_angle*(i, j) between $\nabla_\theta L^{CE}(F(\theta, x_i), c_i)$ and $\nabla_\theta L^{CE}(F(\theta, x_j), c_j)$

 # compute model parameter gradients
$g_i := \nabla_\theta L^{CE}(F(\theta, x_i), c_i)$
$g_j := \nabla_\theta L^{CE}(F(\theta, x_j), c_j)$
calculate cosine similarity
cos_sim$(i, j) := \frac{\langle g_i, g_j \rangle}{\|g_i\|_2 \|g_j\|_2}$
transform to radian
rad_angle$(i, j) :=$ arccos(*cos_sim*(i, j))
transform to degrees
deg_angle$(i, j) := \frac{180}{\pi} \cdot$ *rad_angle*(i, j)

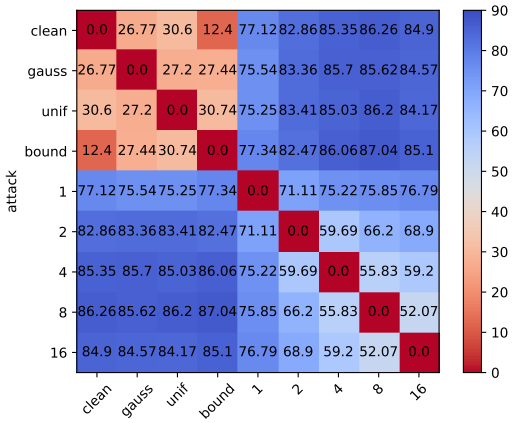
Figure 4.9 shows the median cosine similarities and angles between the gradients for different defenses. We notice that the gradient-free attacks correlate with each other and the clean sample, but not with the images attacked by first-order adversaries. Within them, a similar number of steps increases the similarity, which is reasonable as the gradient ascent scheme is more likely to stop in positions close to each other. Interestingly, even a single-step FGSM or two-step PGD perturbations produce samples with model gradients that, on average, point in a direction that is noticeably related to the one of the 16 step attack. In fact, looking at Figure 4.11, we see that often a single-step FGSM attack still offers valuable information during training about the model parameter gradient of the CEL of stronger attacks and rarely points in an opposing direction on the defended DNNs, i.e. generally encloses acute angles in (c), (d), (e) and (f). Small differences to the completely orthogonal case with cosine similarity of zero and $90°$ angle as in the naturally trained cases (a) and (b) can appear negligible, but [52] found deviations of similar size to contain sufficient information to compute transferred attacks.

When the adversarial radius increases, this information gain of weak attacks seems to decrease, as can be seen when comparing the left and right side of Figure 4.11. Similarly, relating Figure 4.9 and 4.10, the effect seems to be stronger for smaller radii ε_∞. Intuitively, when the attacks access a larger neighbourhood to exploit, it highlights the shortcomings of weak adversaries like FGSM which effectively approximates the loss linearly and maximizes this linear approximation on boundary of the box-constraint. This becomes problematic for larger radii where the function $x \mapsto \nabla_\theta L^{\text{CE}}(F(\theta, x), c)$ has more space to change on the perturbation set, which means that the approximation becomes imprecise and adversarial samples of stronger attacks differ significantly.

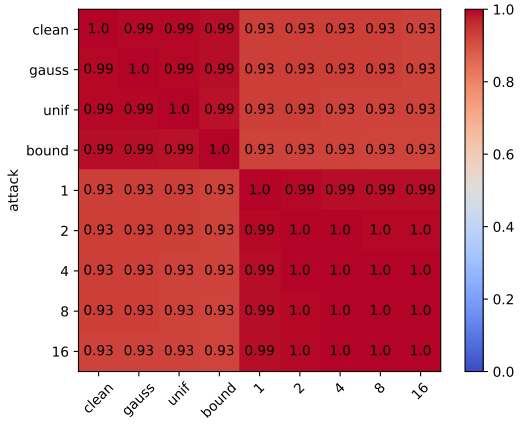
Throughout the training, the angles between gradients $\nabla_\theta L^{\text{CE}}(F(\theta, x), c)$ from weak and strong attacks shrink drastically: The influence of the MadryLab and LocusLab defenses is clearly visible in Figures 4.9 and 4.10 as well as in Figure 4.11 when comparing (a) and (b) with (c), (d), (e) and (f), respectively. Consequently, over the course of adversarial training, the benefit of computing strong attacks declines. This suggests a procedure in more adversarial steps are computed in the beginning of Algorithm 2 and less steps as soon as the gradient similarities increase. Nevertheless, training with a weak FGSM adversary should also yield sufficiently good model parameter gradients for the training procedure to enter the phase of highly correlated gradients, as R-FGSM training proved successful in the LocusLab defense [95].



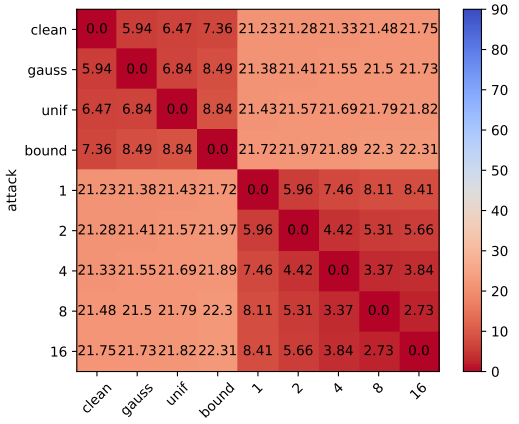
(a) Natural training, cosine similarity



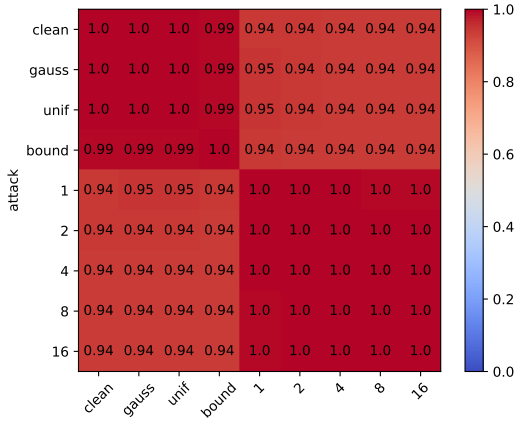
(b) Natural training, angle in degrees



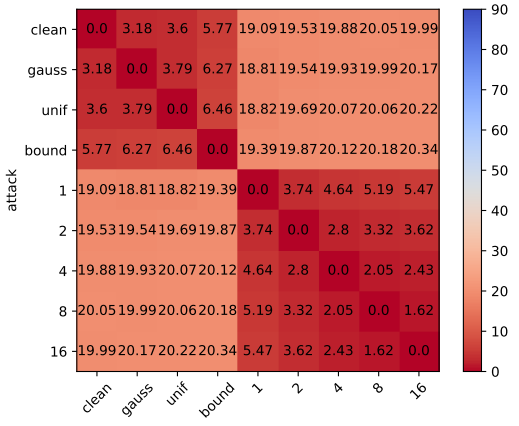
(c) MadryLab training [23], cosine similarity



(d) MadryLab training [23], angle in degrees

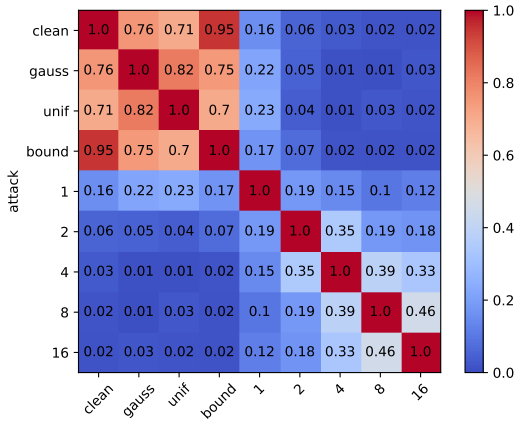


(e) LocusLab training [95], cosine similarity

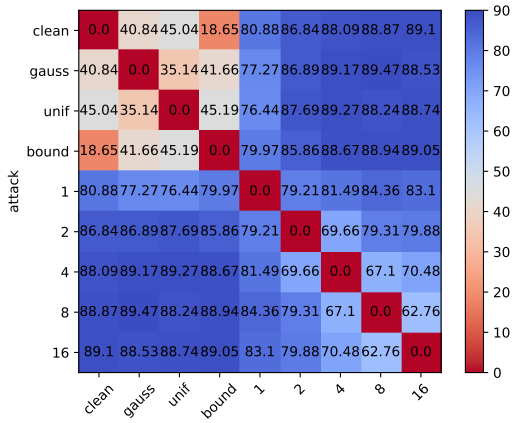


(f) LocusLab training [95], angle in degrees

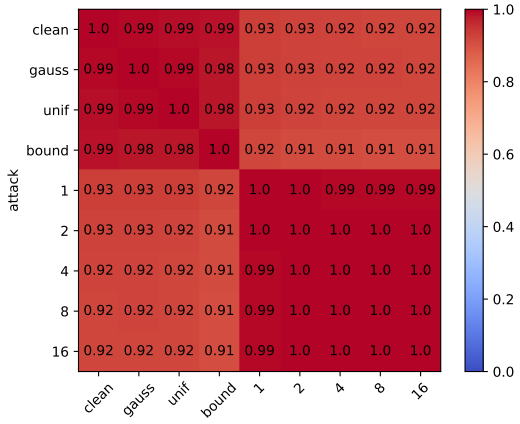
Figure 4.9: Medians of cosine-similarity (left) and angle in degrees (right) between $\nabla_\theta L^{\mathsf{CE}}(F(\theta, x+\delta), c)$ in the adversarial examples of Algorithm 4. Heatmaps computed on ImageNet with perturbation radius $\varepsilon_\infty = 3/255$ and deterministic harmonic step size.



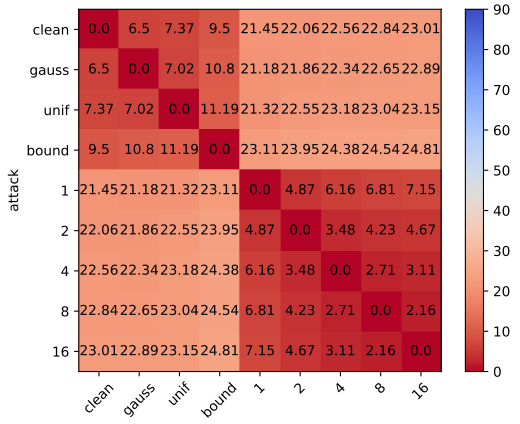
(a) Natural training, cosine similarity



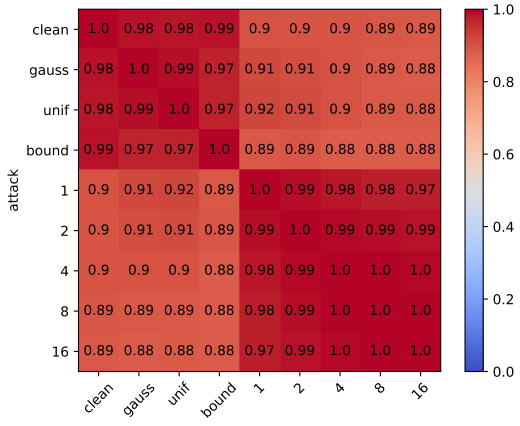
(b) Natural training, angle in degrees



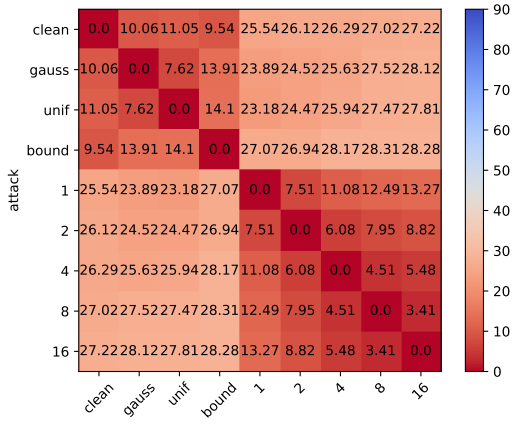
(c) MadryLab training [23], cosine similarity



(d) MadryLab training [23], angle in degrees

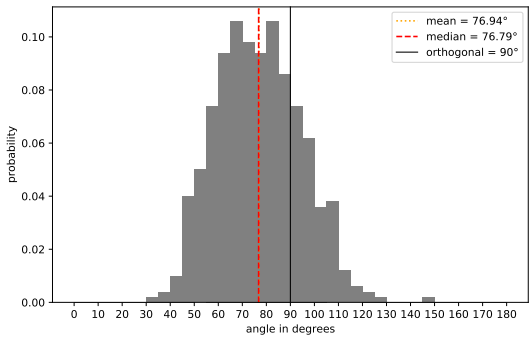


(e) LocusLab training [95], cosine similarity

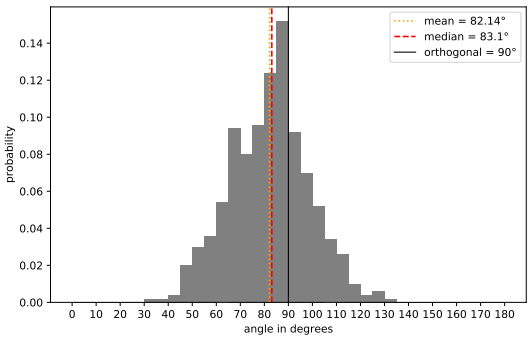


(f) LocusLab training [95], angle in degrees

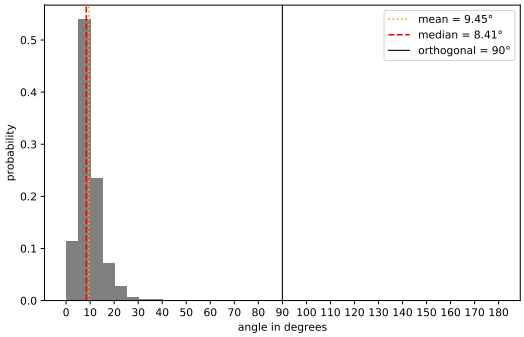
Figure 4.10: Medians of cosine-similarity (left) and angle in degrees (right) between $\nabla_\theta L^{\mathsf{CE}}(F(\theta, x+\delta), c)$ in the adversarial examples of Algorithm 4. Heatmaps computed on ImageNet with perturbation radius $\varepsilon_\infty = 6/255$ and deterministic harmonic step size.



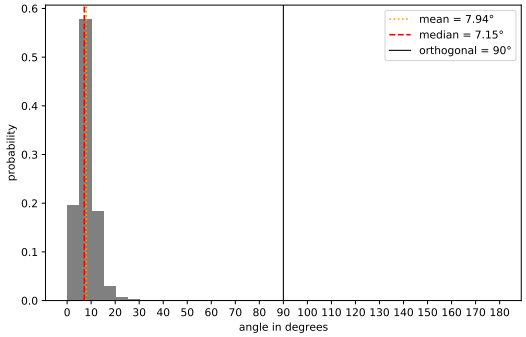
(a) Natural training, $\varepsilon_\infty = 3/255$



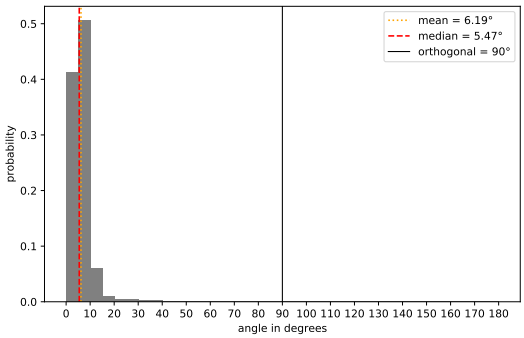
(b) Natural training, $\varepsilon_\infty = 6/255$



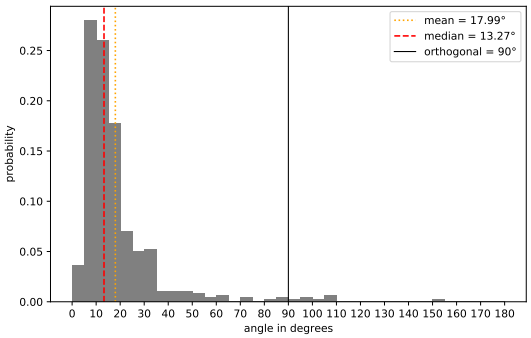
(c) MadryLab training [23], $\varepsilon_\infty = 3/255$



(d) MadryLab training [23], $\varepsilon_\infty = 6/255$



(e) LocusLab training [95], $\varepsilon_\infty = 3/255$



(f) LocusLab training [95], $\varepsilon_\infty = 6/255$

Figure 4.11: Histograms of angles in degrees between model parameter gradients of CEL $\nabla_\theta L^{\mathsf{CE}}(F(\theta, x + \delta), c)$ after one versus 16 gradient ascent steps on ImageNet with perturbation radius $\varepsilon_\infty = 3/255$ (left) and $\varepsilon_\infty = 6/255$ (right) and deterministic harmonic step size in Algorithm 4.

To recapitulate, we saw regarding the question from [80] that Danskin's Theorem is not as sensitive to violations of its assumptions in the context of the empirical robust training problem (E-RO) as it is in Example 3.3. The measured model gradient angles support our hypothesis and extend it by the observation that these angles decrease through adversarial training. This suggests that weaker adversaries are needed in late iterations of adversarial training, which can be a starting point to answer the question raised in [95] about the necessity of strong adversaries during training. We deduce the following contribution.

Contribution 4.3 (Danskin's Theorem in Practice)**.** *Adversarial training has proven empirically successful even though it lacks the theoretical justification of Danskin's Theorem (Theorem 3.1). We can explain this by the similarity between model parameter gradients $\nabla_\theta L^{CE}(F(\theta, x+\delta), c)$ of points in the perturbation set that resulted from differently strong attacks. The angles seem to shrink during adversarial training, making weak adversaries increasingly valuable. This motivates a training scheme with more gradient steps in the beginning and less towards the end, which remains to be experimented with.*

4.4 BOBYQA Black-Box Attacks Against Modern Defenses

Our final contribution aims to update the empirical study of targeted BOBYQA black-box attacks in [92, 93] to state-of-the-art defenses. To this end, we attack the smoothed classifier from Hadi Salman [78] and the R-FGSM trained DNN from LocusLab [95] besides the naturally trained and adversarially trained nets from [23] that have already been tested in [93].

4.4.1 Question and Metrics

Concretely, we want to find out how the targeted BOBYQA black-box attack performs against modern state-of-the-art randomized adversarial training and randomized smoothing. We choose $M_{\text{query}} = 1$ for the query-time Gaussian smoothing and stay consistent with the default settings recommended in the `GitHub` repository of [93] regarding the parameters in the optimization and lifting process, c.f. Section 2.2.2. The attack success is quantified by the CDFs of the adversarial success rate w.r.t. the number of function evaluations of $F(\theta, \cdot)$ that have been executed.

4.4.2 Setup and Implementation

We forked the code from the `GitHub` repository to [93] before merely changing it and the underlying `robustness` package [23] to accommodate for the two additional DNNs that we test. For comparability, we also use the same `seed = 1216` and randomized procedure to pick samples x and target labels $\widetilde{c}$ uniformly distributed from the CIFAR-10 test set. Table 4.7 presents the attack radii ε_∞ and the corresponding number of misclassification pairs $(x, \widetilde{c}) \in \mathcal{X} \times \mathcal{C}$ that we attempt to create. Table 4.8 lists the radii that the models were trained to be robust against. For all defenses, we chose the classifier with the most appropriate training ε_p to withstand the attack.

	DNN			Attack ε_∞		
Defense	Train on	p	Architecture	$3/255$	$6/255$	$9/255$
Natural [23]	Clean		ResNet50	500	500	500
MadryLab [23]	Multi-step	$2, \infty$	ResNet50	500	500	500
LocusLab [95]	R-FGSM	∞	PreAct ResNet18	500	500	500
HadiSalman [78]	Rand. Smoth.	2	ResNet110	500	500	500

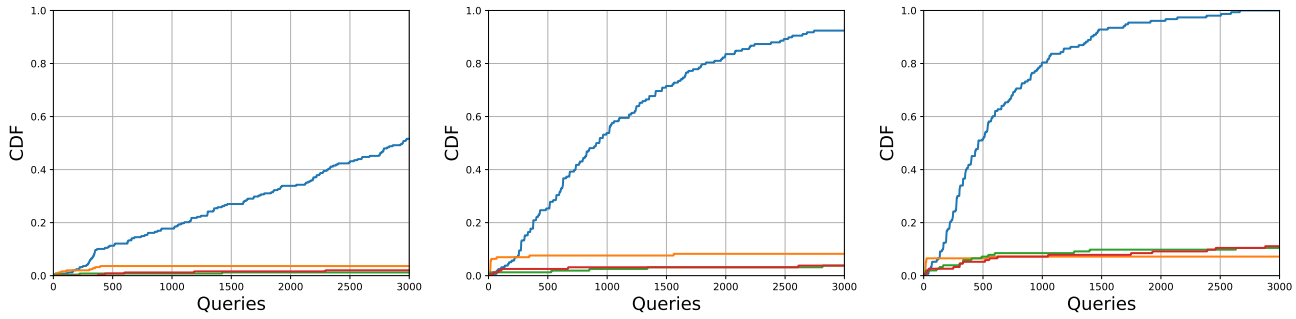
Table 4.7: Number of tested pairs of sample and random target label from the CIFAR-10 test data set for each attack configuration of the experiments regarding BOBYQA black-box attacks.

Defense	DNN Train on	p	Architecture	Attack ε_∞ $3/255$	$6/255$	$9/255$
MadryLab [23]	Multi-step	∞	ResNet50	4/255	8/255	8/255
LocusLab [95]	R-FGSM	∞	ResNet110	8/255	8/255	8/255
HadiSalman [78]	Rand. Smoth.	2	PreAct ResNet18	0.375	0.625	1.0

Table 4.8: Training ε_p regarding p from the 'p'-column for each attack of the BOBYQA black-box attacks. For HadiSalman [78]: Most robust smoothed models for 2-norm bounded PGD attacks with $\varepsilon_2 = T^2_\infty(\varepsilon_\infty)$ were chosen. Entries show the radius closest to ε_2 that was tested for robustness.

4.4.3 Results and Interpretation

First of all, Figure 4.12 emphasizes the strong influence of the attack radius ε_∞ and all adversarial defenses: Each increment of $3/255$ raises the overall attack success rate after at most $3,000$ queries and allows successful perturbations to be found quicker for the natural and adversarially trained DNNs. The smoothed classifier seems to react less sensitive to changes in the radius. All robustification methods mitigate BOBYQA's success well below 5%, 10% and 15% for $\varepsilon_\infty \in \{3/255, 6/255, 9/255\}$, respectively. This holds, despite the latter radius allowing for prominent perturbation artefacts similar to Figure 2.6 and therefore meeting neither criterion (a) nor (b) from our intuitive definition of adversarial examples in the introduction.



(a) $\varepsilon_\infty = 3/255 \approx 0.012$ (b) $\varepsilon_\infty = 6/255 \approx 0.024$ (c) $\varepsilon_\infty = 9/255 \approx 0.035$

Figure 4.12: CDF of the number of queries BOBYQA needs to successfully perform targeted attacks for various attack radii ε_∞ on 500 pairs of CIFAR-10 test images and target labels. Defenses include natural PyTorch pretraining (blue), MadryLab [23] PGD adversarial training (green), LocusLab [95] R-FGSM training (red) and HadiSalman [78] randomized smoothing (orange).

Taking into account that the attacks take place in a black-box setting with random target labels, this success rate is still remarkable and alarming, especially the early jump of the smoothed DNN's CDF. Here, already dozens of queries suffice to misguide the classifier. Ultimately, neither of the more recently published defenses outperforms the PGD-based MadryLab defense. We summarize our findings below.

Contribution 4.4 (BOBYQA-Attacks Against Modern Defenses)**.** *Targeted BOBYQA black-box attacks [92, 93] retain moderate success against state-of-the-art implementations of adversarial training [95] and randomized smoothing [78]. This performance and its unique model-based, rather than transferred, approach to black-box attacks justifies it to be added to and remain in the portfolio of validation attacks for adversarial defenses.*

A similar suggestion is made in [3] and [89] about zeroth-order adversarial attacks as a whole.

5 Conclusion and Outlook

5.1 Conclusion

To summarize, we started by formulating the adversarial training problem as a distributionally robust optimization problem (DRO). By considering specific ambiguity sets, we motivated standard adversarial training and randomized smoothing as attempt to approximately solve the robust (RO) and the stochastic robustification problem (SO), respectively. In the case of gradient-based adversarial training, we established that Danskin's Theorem (Theorem 3.1) does not apply and that this completely invalidates its claim in Example 3.3. However, we provided a possible explanation as to why this approach remains to work in practice, in part answering the question from [80] why adversarial training increases robustness, cf. Contribution 4.3. As a consequence, we expect declining adversarial strength during training to be efficient and give a possible answer to the question raised in [95] about when strong adversaries are necessary for training and when weak ones suffice.

Proposition 1.5 showed the NP-hardness of determining p-norm robustness of a ReLU-based DNNs. Especially, finding an adversarial example for sure, if it exists, is NP-hard. Despite this fact and many advancements in adversarial defenses, DNN remain susceptible to adversarial attacks at query time. Beyond that, there remains a shortage of certified defenses and certification procedures scalable to real-world-sized DNNs. Hence, lower bounds on the robustness radius remain hard to obtain.

Therefore, determining upper bounds on the robustness, by finding strong adversarial examples close to clean data points, remains crucial. The pool of such test attacks needs to be as diverse as the landscape of competitive attacks itself with regards to attack and defense taxonomy as well as optimization strategy. Our numerical results demonstrated the potency of the BOBYQA-based attack against state-of-the-art adversarial training and randomized smoothing. In conjunction with the results from [92] and [93], this clearly shows the competitiveness of this attack. Furthermore, due to its unique zeroth-order model-based black-box nature, it is an outlier in the aforementioned landscape of attacks, considering that most attacks are of first-order and take place in a white-box setting before potentially being transferred to the target DNN. We therefore, for the time being, advocate for BOBYQA-based attacks being investigated more thoroughly and being part of the evaluation set of adversarial defenses, cf. Contribution 4.4.

Moreover, we proposed two PGD attacks with harmonically and geometrically decaying step lengths. The former enabled finding adversarial examples with unprecedented zero true-label-confidences, cf. Contribution 4.1. Furthermore, we defined p-norm rescaling functions that make differently bounded adversaries more comparable when choosing model parameters, pretrained models or interpreting results of different origins. They follow the idea of nested level sets and equal Lebesgue volume, respectively. Both showed promising results on ImageNet when measuring CEL as well as top-1 and top-5 accuracy against various first-order adversaries and under different noises, cf. Contribution 4.2.

5.2 Outlook

Regarding this work in particular, there are several open questions. The most pressing ones are listed below.

- Why is there such a small performance difference between constant step sizes and our proposed step schemes from Algorithm 4 despite their vastly different nature? Is there further potential to increase the threat strength by tuning these parameters? Do our proposed harmonic step lengths also produce zero true-label-confidences for other p-norms and data sets as in Contribution 4.1? Is out hypothesis about the reason of the absence of the phenomenon for the other two step schemes correct?
- Do our proposed p-norm rescaling functions from Contribution 4.2 also induce a similarly strong adversarial attacks outside the considered test cases and data sets?
- How does our suggestion on adversarial training from Contribution 4.3 perform?
- On the one hand, how does the trade-off between the time to classify and the defensive gain develop when increasing the number of samples M_{query} during evaluation of a smoothed classifier in the context of BOBYQA attacks? On the other hand how does the offensive gain behave when increasing the number d of BOBYQA interpolation points from $n+1$ to $2n+1$ like [39] to be reasonable for a quadratic interpolation? Could this further strengthen our suggestion in Contribution 4.4 to recognize BOBYQA attacks as a threat to modern defenses?

More broadly, in the view of the vitality of this research field, there are many directions for future work, both on the attacking and on the defending side.

- New dimension- and query-reduction techniques, that identify the most influential pixels for the attack to focus on, are desirable to speed up attacks and, by extension, also adversarial training. To this end, [92] suggests to apply methods from compressed sensing and [6] uses principle component analysis (PCA). However, there are numerous other approaches to find such pixels. For example, in a white-box setting feature, importance can be measured by the mean decrease in impurity (MDI) score [66], leveraging permutations and random forests, or several confidence gain metrics like simple confidence gain (SCG) and concise confidence gain (CCG) [79]. In a black-box setting, there is a lack of query-efficient feature selection techniques beyond the ones presented in [15, 92, 93].
- It is worthwhile investigating the effect of temperature at training on the robustness of the network, possibly revisiting and combining defensive distillation [61, 62, 63, 9] with more recent defenses.
- There are signs that pretraining and semi-supervised learning improve robustness and complement well with other defenses like randomized smoothing [103, 102, 91, 11]. In general, evaluating combinations of state-of-the-art defense techniques remains a continuous task as the field progresses.
- A recent results [27] draws a strong connection between corruption robustness and adversarial robustness, which goes hand in hand with the success of randomized smoothing, and calls for a deeper collaboration between both fields.
- The notion of p-norm robustness is mathematically convenient but questionable for capturing the essence of human visual perception [81] since, for instance, slight rotations and flips cause large ε_p but pose no challenge to humans [89], showing that small p-norm distances are neither necessary nor sufficient for visual similarity [81]. Therefore, exploring constraints in terms of more elaborate metrics, like Voronoi Cells [36] and PASS [75], leveraging the SSIM [26], seems like an exciting research direction to pursue.

5.3 Acknowledgements

To complete this work, I want to express my deep appreciation to Prof. Stefan Ulbrich for his advice and insightful discussions during the creation of this thesis. Further, I am grateful to the Working Group for Optimization at TU Darmstadt for providing the necessary computational resources to conduct the numerical experiments.

A special thanks goes to Rinor Cakaj for his answers to my technical questions regarding the `Python` implementation and proofreading this work together with Konstanze Klemet and Mino Nicola Kraft. Lastly, I would like to acknowledge Giuseppe Ughi for our exchange on the code of his papers [92, 93].

Bibliography

[1] George E. Andrews, Richard Askey, and Ranjan Roy. *Special Functions*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1999. DOI: 10.1017/CBO9781107325937.

[2] Maksym Andriushchenko and Nicolas Flammarion. *Understanding and Improving Fast Adversarial Training*. 2020. arXiv: 2007.02617 [cs.LG].

[3] Anish Athalye, Nicholas Carlini, and David Wagner. *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*. 2018. arXiv: 1802.00420 [cs.LG].

[4] Dimitri P. Bertsekas. *Nonlinear Programming*. eng. Second edition. Belmont (Mass.): Athena Scientific, C 1999. ISBN: 1-886529-00-0.

[5] Dimitri P. Bertsekas. *Supplementary Chapter 6 on Convex Optimization Algorithms*. eng. Athena Scientific, 2014. ISBN: 978-1-886529-31-1. URL: https://www.mit.edu/~dimitrib/Chapter_6_Web_Posted.pdf.

[6] Arjun Nitin Bhagoji et al. "Practical Black-Box Attacks on Deep Neural Networks Using Efficient Query Mechanisms". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 158–174. ISBN: 978-3-030-01258-8. DOI: 10.1007/978-3-030-01258-8_10.

[7] Battista Biggio, Giorgio Fumera, and Fabio Roli. "Multiple classifier systems for robust classifier design in adversarial environments". In: *J. Mach. Learn. Cybern.* 1 (Dec. 2010), pp. 27–41. DOI: 10.1007/s13042-010-0007-7.

[8] Battista Biggio et al. "Evasion Attacks against Machine Learning at Test Time". In: *Advanced Information Systems Engineering*. Springer Berlin Heidelberg, 2013, pp. 387–402. DOI: 10.1007/978-3-642-40994-3_25.

[9] Nicholas Carlini and David Wagner. *Defensive Distillation is Not Robust to Adversarial Examples*. 2016. arXiv: 1607.04311 [cs.CR].

[10] Nicholas Carlini and David Wagner. *Towards Evaluating the Robustness of Neural Networks*. 2017. arXiv: 1608.04644 [cs.CR].

[11] Yair Carmon et al. *Unlabeled Data Improves Adversarial Robustness*. 2022. arXiv: 1905.13736 [stat.ML].

[12] Coralia Cartis, Lindon Roberts, and Oliver Sheridan-Methven. "Escaping local minima with local derivative-free methods: a numerical investigation". In: *Optimization* 71.8 (2022), pp. 2343–2373. DOI: 10.1080/02331934.2021.1883015.

[13] Coralia Cartis et al. "Improving the Flexibility and Robustness of Model-Based Derivative-Free Optimization Solvers". In: *ACM Trans. Math. Softw.* 45.3 (Aug. 2019). ISSN: 0098-3500. DOI: 10.1145/3338517. URL: https://github.com/numericalalgorithmsgroup/pybobyqa.git.

[14] Pin-Yu Chen et al. *EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples*. 2018. arXiv: 1709.04114 [stat.ML].

[15] Pin-Yu Chen et al. "ZOO". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, Nov. 2017. DOI: 10.48550/arXiv.1708.03999.

[16] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. *Certified Adversarial Robustness via Randomized Smoothing*. 2019. arXiv: 1902.02918 [cs.LG].

[17] Francesco Croce, Jonas Rauber, and Matthias Hein. "Scaling up the Randomized Gradient-Free Adversarial Attack Reveals Overestimation of Robustness Using Established Attacks". In: *International Journal of Computer Vision* 128 (Apr. 2020). DOI: 10.1007/s11263-019-01213-0.

[18] Nilesh Dalvi et al. "Adversarial Classification". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. Seattle, WA, USA: Association for Computing Machinery, 2004, pp. 99–108. ISBN: 1581138881. DOI: 10.1145/1014052.1014066.

[19] John M. Danskin. *The theory of Max-Min and its application to weapons allocation problems [by] John M. Danskin*. English. Springer-Verlag Berlin, New York, 1967, viii, 126 p.

[20] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009.

[21] Yinpeng Dong et al. *Towards Interpretable Deep Neural Networks by Leveraging Adversarial Examples*. 2017. arXiv: 1708.05493 [cs.CV].

[22] Abdullah Al-Dujaili et al. "On the application of Danskin's theorem to derivative-free minimax problems". In: *AIP Conference Proceedings*. Author(s), 2019. DOI: 10.1063/1.5089993.

[23] Logan Engstrom et al. *Robustness (Python Library)*. 2019. URL: https://github.com/MadryLab/robustness.

[24] Kevin Eykholt et al. *Robust Physical-World Attacks on Deep Learning Models*. 2018. arXiv: 1707.08945 [cs.CR].

[25] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. *Analysis of classifiers' robustness to adversarial perturbations*. 2016. arXiv: 1502.02590 [cs.LG].

[26] Jeremy Flynn et al. "Image Quality Assessment Using the SSIM and the Just Noticeable Difference Paradigm". In: vol. 8019. July 2013. ISBN: 978-3-642-39359-4. DOI: 10.1007/978-3-642-39360-0_3.

[27] Nic Ford et al. *Adversarial Examples Are a Natural Consequence of Test Error in Noise*. 2019. arXiv: 1901.10513 [cs.LG].

[28] Ruiqi Gao et al. *Convergence of Adversarial Training in Overparametrized Neural Networks*. 2019. arXiv: 1906.07916 [cs.LG].

[29] Justin Gilmer et al. *Adversarial Spheres*. 2018. arXiv: 1801.02774 [cs.CV].

[30] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML].

[31] Serge Gratton, Annick Sartenaer, and Philippe L. Toint. "Recursive Trust-Region Methods for Multiscale Nonlinear Optimization". In: *SIAM Journal on Optimization* 19.1 (2008), pp. 414–444. DOI: 10.1137/050623012.

[32] Shuyue Guan and Murray Loew. "Analysis of Generalizability of Deep Neural Networks Based on the Complexity of Decision Boundary". In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, Dec. 2020. DOI: 10.1109/icmla51294.2020.00025.

[33] Gregory Gundersen. *From Convolution to Neural Network*. 2017. URL: gregorygundersen.com/blog/2017/02/24/cnns/.

[34] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[35] Guy Katz et al. *Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks*. 2017. arXiv: 1702.01135 [cs.AI].

[36] Marc Khoury and Dylan Hadfield-Menell. *Adversarial Training with Voronoi Constraints*. 2019. arXiv: 1905.01019 [cs.LG].

[37] Hoki Kim, Woojin Lee, and Jaewook Lee. *Understanding Catastrophic Overfitting in Single-step Adversarial Training*. 2020. arXiv: 2010.01799 [cs.LG].

[38] Aleksander Kolcz and Choon Hui Teo. "Feature Weighting for Improved Classifier Robustness". In: *International Conference on Email and Anti-Spam*. 2009. URL: https://api.semanticscholar.org/CorpusID:15223859.

[39] Philip Kolvenbach, Oliver Lass, and Stefan Ulbrich. "An approach for robust PDE-constrained optimization with application to shape optimization of electrical engines and of dynamic elastic structures under uncertainty". In: *Optimization and Engineering* 19 (Sept. 2018), pp. 1–35. DOI: 10.1007/s11081-018-9388-3.

[40] A. Krizhevsky. *The CIFAR-10 and the CIFAR-100 Dataset*. URL: https://www.cs.toronto.edu/~kriz/cifar.html.

[41] A. Krizhevsky and G. Hinton. "Learning multiple layers of features from tiny images". In: (2009). URL: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[42] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. *Adversarial examples in the physical world*. 2017. arXiv: 1607.02533 [cs.CV].

[43] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. *Adversarial Machine Learning at Scale*. 2017. arXiv: 1611.01236 [cs.CV].

[44] Mathias Lecuyer et al. *Certified Robustness to Adversarial Examples with Differential Privacy*. 2019. arXiv: 1802.03471 [stat.ML].

[45] Klas Leino, Zifan Wang, and Matt Fredrikson. *Globally-Robust Neural Networks*. 2021. arXiv: 2102.08452 [cs.LG].

[46] Bai Li et al. *Certified Adversarial Robustness with Additive Noise*. 2019. arXiv: 1809.03113 [cs.LG].

[47] Yanpei Liu et al. *Delving into Transferable Adversarial Examples and Black-box Attacks*. 2017. arXiv: 1611.02770 [cs.LG].

[48] Yu Han Liu. "Feature Extraction and Image Recognition with Convolutional Neural Networks". In: *Journal of Physics: Conference Series* 1087.6 (Sept. 2018), p. 062032. DOI: 10.1088/1742-6596/1087/6/062032. URL: https://dx.doi.org/10.1088/1742-6596/1087/6/062032.

[49] Alessio Lomuscio and Lalit Maganti. *An approach to reachability analysis for feed-forward ReLU neural networks*. 2017. arXiv: 1706.07351 [cs.AI].

[50] Daniel Lowd and Christopher Meek. "Adversarial Learning". In: *Knowledge Discovery and Data Mining*. 2005. URL: https://api.semanticscholar.org/CorpusID:6259400.

[51] Jiajun Lu, Theerasit Issaranon, and David Forsyth. *SafetyNet: Detecting and Rejecting Adversarial Examples Robustly*. 2017. arXiv: 1704.00103 [cs.CV].

[52] Aleksander Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2019. arXiv: 1706.06083 [stat.ML].

[53] Akshay Mehra et al. *On Certifying and Improving Generalization to Unseen Domains*. 2022. arXiv: 2206.12364 [cs.LG].

[54] Dongyu Meng and Hao Chen. *MagNet: a Two-Pronged Defense against Adversarial Examples*. 2017. arXiv: 1705.09064 [cs.CR].

[55] Takeru Miyato et al. *Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning*. 2018. arXiv: 1704.03976 [stat.ML].

[56] Katta G. Murty and Santosh N. Kabadi. "Some NP-complete problems in quadratic and nonlinear programming". In: *Mathematical Programming* 39 (1987), pp. 117–129. DOI: 10.1007/BF02592948.

[57] Yaniv Nemcovsky et al. "Adversarial robustness via noise injection in smoothed models". In: *Applied Intelligence* 53 (Aug. 2022). DOI: 10.1007/s10489-022-03423-5.

[58] Jerzy Neyman, Egon Sharpe Pearson, and Karl Pearson. "IX. On the problem of the most efficient tests of statistical hypotheses". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231.694-706 (1933), pp. 289–337. DOI: 10.1098/rsta.1933.0009.

[59] Anh Nguyen, Jason Yosinski, and Jeff Clune. *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*. 2015. arXiv: 1412.1897 [cs.CV].

[60] Linh Nguyen, Sky Wang, and Arunesh Sinha. *A Learning and Masking Approach to Secure Learning*. 2018. arXiv: 1709.04447 [cs.CR].

[61] Nicolas Papernot and Patrick McDaniel. *Extending Defensive Distillation*. 2017. arXiv: 1705.05264 [cs.LG].

[62] Nicolas Papernot and Patrick McDaniel. *On the Effectiveness of Defensive Distillation*. 2016. arXiv: 1607.05113 [cs.CR].

[63] Nicolas Papernot et al. "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks". In: May 2016, pp. 582–597. DOI: 10.1109/SP.2016.41.

[64] Nicolas Papernot et al. *The Limitations of Deep Learning in Adversarial Settings*. 2015. arXiv: 1511.07528 [cs.CR].

[65] Nicolas Papernot et al. *Towards the Science of Security and Privacy in Machine Learning*. 2016. arXiv: 1611.03814 [cs.CR].

[66] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[67] Kexin Pei et al. "DeepXplore: Automated Whitebox Testing of Deep Learning Systems". In: *Proceedings of the 26th Symposium on Operating Systems Principles*. SOSP '17. Shanghai, China: Association for Computing Machinery, 2017, pp. 1–18. ISBN: 9781450350853. DOI: 10.1145/3132747.3132785.

[68] Marc Pfetsch and Stefan Ulbrich. *Optimization Methods for Machine Learning (Lecture Notes)*. TU Darmstadt. Winter term 2020/21.

[69] Rafael Pinot et al. *On the robustness of randomized classifiers to adversarial examples*. 2021. arXiv: 2102.10875 [cs.LG].

[70] Rafael Pinot et al. *Theoretical evidence for adversarial robustness through randomization*. 2019. arXiv: 1902.01148 [cs.LG].

[71] M. Powell. "DAMTP 2004/NA01 On updating the inverse of a KKT matrix". In: (Feb. 2004). URL: http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2004_01.pdf.

[72] M. Powell. “The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives”. In: *Technical Report, Department of Applied Mathematics and Theoretical Physics* (Jan. 2009). URL: https://optimization-online.org/?p=11137.

[73] M.J.D. Powell. “Least Frobenius Norm Updating of Quadratic Models that Satisfy Interpolation Conditions”. In: *Mathematical Programming* 100 (May 2004), pp. 183–215. DOI: 10.1007/s10107-003-0490-7.

[74] Herbert Robbins. “A Remark on Stirling’s Formula”. In: *The American Mathematical Monthly* 62.1 (1955), pp. 26–29. ISSN: 00029890, 19300972. DOI: 10.2307/2308012. (Visited on 09/27/2023).

[75] Andras Rozsa, Ethan M. Rudd, and Terrance E. Boult. *Adversarial Diversity and Hard Positive Generation*. 2016. arXiv: 1605.01775 [cs.CV].

[76] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y. URL: https://image-net.org/challenges/LSVRC/index.php.

[77] Hadi Salman et al. *Do Adversarially Robust ImageNet Models Transfer Better?* 2020. arXiv: 2007.08489 [cs.CV].

[78] Hadi Salman et al. *Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers*. 2020. arXiv: 1906.04584 [cs.LG].

[79] Sai Prabhakar Pandi Selvaraj, Manuela Veloso, and Stephanie Rosenthal. “Classifier-Based Evaluation of Image Feature Importance”. In: *GCAI-2018. 4th Global Conference on Artificial Intelligence*. Ed. by Daniel Lee, Alexander Steen, and Toby Walsh. Vol. 55. EPiC Series in Computing. EasyChair, 2018, pp. 162–175. DOI: 10.29007/p655.

[80] Uri Shaham, Yutaro Yamada, and Sahand Negahban. “Understanding adversarial training: Increasing local stability of supervised models through robust optimization”. In: *Neurocomputing* 307 (Sept. 2018), pp. 195–204. DOI: 10.1016/j.neucom.2018.04.027.

[81] Mahmood Sharif, Lujo Bauer, and Michael K. Reiter. *On the Suitability of L_p-norms for Creating and Preventing Adversarial Examples*. 2018. arXiv: 1802.09653 [cs.CR].

[82] Yash Sharma and Pin-Yu Chen. *Attacking the Madry Defense Model with L_1-based Adversarial Examples*. 2018. arXiv: 1710.10733 [stat.ML].

[83] Connor Shorten and Taghi Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. In: *Journal of Big Data* 6 (July 2019). DOI: 10.1186/s40537-019-0197-0.

[84] Aman Sinha et al. *Certifying Some Distributional Robustness with Principled Adversarial Training*. 2020. arXiv: 1710.10571 [stat.ML].

[85] Chawin Sitawarin et al. *DARTS: Deceiving Autonomous Cars with Toxic Signs*. 2018. arXiv: 1802.06430 [cs.CR].

[86] Christian Szegedy et al. *Intriguing properties of neural networks*. 2014. arXiv: 1312.6199 [cs.CV].

[87] Simon F. B. Tett et al. “Can Top-of-Atmosphere Radiation Measurements Constrain Climate Predictions? Part I: Tuning”. In: *Journal of Climate* 26.23 (2013), pp. 9348–9366. DOI: 10.1175/JCLI-D-12-00595.1.

[88] Antonio Torralba, Rob Fergus, and William T. Freeman. “80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.11 (2008), pp. 1958–1970. DOI: 10.1109/TPAMI.2008.128.

[89] Florian Tramèr et al. *Ensemble Adversarial Training: Attacks and Defenses*. 2020. arXiv: 1705.07204 [stat.ML].

[90] Florian Tramèr et al. *The Space of Transferable Adversarial Examples*. 2017. arXiv: 1704.03453 [stat.ML].

[91] Jonathan Uesato et al. *Are Labels Required for Improving Adversarial Robustness?* 2019. arXiv: 1905.13725 [cs.LG].

[92] Giuseppe Ughi, Vinayak Abrol, and Jared Tanner. *A Model-Based Derivative-Free Approach to Black-Box Adversarial Examples: BOBYQA*. 2020. arXiv: 2002.10349 [cs.LG].

[93] Giuseppe Ughi, Vinayak Abrol, and Jared Tanner. "An empirical study of derivative-free-optimization algorithms for targeted black-box attacks in deep neural networks". In: *Optimization and Engineering* 23 (Sept. 2022), pp. 1–28. DOI: 10.1007/s11081-021-09652-w.

[94] Liu Wen. "An Analytic Technique to Prove Borel's Strong Law of Large Numbers". In: *The American Mathematical Monthly* 98.2 (1991), pp. 146–148. ISSN: 00029890, 19300972. DOI: 10.2307/2323947.

[95] Eric Wong, Leslie Rice, and J. Zico Kolter. *Fast is better than free: Revisiting adversarial training*. 2020. arXiv: 2001.03994 [cs.LG].

[96] Weilin Xu, David Evans, and Yanjun Qi. *Feature Squeezing Mitigates and Detects Carlini/Wagner Adversarial Examples*. 2017. arXiv: 1705.10686 [cs.CR].

[97] Weilin Xu, David Evans, and Yanjun Qi. "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks". In: *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society, 2018. DOI: 10.14722/ndss.2018.23198.

[98] Zhewei Yao et al. *Hessian-based Analysis of Large Batch Training and Robustness to Adversaries*. 2018. arXiv: 1802.08241 [cs.CV].

[99] Zhewei Yao et al. *Trust Region Based Adversarial Attack on Neural Networks*. 2018. arXiv: 1812.06371 [cs.LG].

[100] Xiaoyong Yuan et al. *Adversarial Examples: Attacks and Defenses for Deep Learning*. 2018. arXiv: 1712.07107 [cs.LG].

[101] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. *Efficient Defenses Against Adversarial Attacks*. 2017. arXiv: 1707.06728 [cs.LG].

[102] Runtian Zhai et al. *Adversarially Robust Generalization Just Requires More Unlabeled Data*. 2019. arXiv: 1906.00555 [cs.LG].

[103] Runtian Zhai et al. *MACER: Attack-free and Scalable Robust Training via Maximizing Certified Radius*. 2022. arXiv: 2001.02378 [cs.LG].

[104] Huan Zhang et al. *The Limitations of Adversarial Training and the Blind-Spot Attack*. 2019. arXiv: 1901.04684 [stat.ML].

[105] Yihua Zhang et al. *Revisiting and Advancing Fast Adversarial Training Through The Lens of Bi-Level Optimization*. 2022. arXiv: 2112.12376 [cs.LG].