An introduction to adversarially robust deep learning

Jonathan Peck, Bart Goossens and Yvan Saeys

Abstract—The widespread success of deep learning in solving machine learning problems has fueled its adoption in many fields, from speech recognition to drug discovery and medical imaging. However, deep learning systems are extremely fragile: imperceptibly small modifications to their input data can cause the models to produce erroneous output. It is very easy to generate such adversarial perturbations even for state-of-the-art models, yet immunization against them has proven exceptionally challenging. Despite over a decade of research on this problem, our solutions are still far from satisfactory and many open problems remain. In this work, we survey some of the most important contributions in the field of adversarial robustness. We pay particular attention to the reasons why past attempts at improving robustness have been insufficient, and we identify several promising areas for future research.

Index Terms—Deep learning, adversarial machine learning, computer vision

1 Introduction

Artificial intelligence (AI) has been revolutionized by the emergence of *deep learning* (DL), *i.e.*, the use of *deep neural networks* (DNNs) to solve machine learning (ML) problems [1], [2]. Since the publication of AlexNet in 2012 drastically improved the performance of image recognition systems [3], DNNs have been successfully applied to essentially every conceivable AI domain. On the ImageNet Large Scale Visual Recognition Challenge [4], for example, [5] achieve 91% top-1 accuracy, compared to the 60% top-1 accuracy achieved by AlexNet ten years prior. Using reinforcement learning [6], researchers have been able to construct deep learning systems that can achieve remarkable competency in certain video games and board games, ranging from simple Atari games [7] to complex strategy games such as Dota 2 [8]. Recently, major advances in generative modeling using diffusion models [9] for images and Transformers [10] for natural language understanding have made headlines around the world due to their remarkable results. The results obtained by modern DL techniques are sufficiently impressive that DNNs are now also applied to very sensitive and safety-critical problems, such as medical diagnoses [11], [12] and autonomous driving [13]. Deep learning has been an important part of the worldwide response to the COVID-19 pandemic, aiding in the discovery of vaccines and other therapeutic treatments [14], [15] as well as forecasting hospitalizations and case numbers [16].

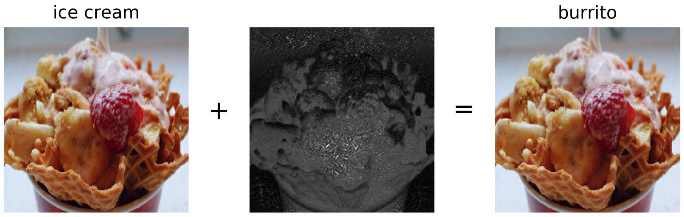


Fig. 1.1: An adversarial example in the image recognition domain.

It is clear that deep learning is rapidly becoming a significant part of modern life, and DNNs are increasingly used to solve problems where accuracy is paramount and mistakes can cause tangible harm. As such, ensuring that these AI-powered systems are sufficiently trustworthy has become a major area of research. In this regard, it has been found that DNNs can in fact be very easily fooled into providing incorrect predictions even when they are highly accurate on the task at hand [17], [18]. Specifically, for almost any deep learning system, it is generally easy to craft inputs to that system for which the output is plainly wrong. Moreover, these inputs are often indistinguishable from "normal" data on which the system tends to perform very well. In fact, most methods for generating such inputs start from regular test data on which the model is highly accurate and perturb it in very small but specific ways to cause erroneous results. These are referred to as *adversarial perturbations*: small alterations to normal data samples which are specifically crafted to fool some target model.

Figure 1.1 shows an example of an adversarial perturbation on a DNN for image recognition. The network initially classifies the image on the left correctly as `ice cream`. Adding the perturbation in the middle pixel-wise to the original image and clipping the values back to the original range results in the image on the right. Despite being visually identical to the original, the classification has now changed to `burrito`. Although adversarial perturbations are easiest to generate for image recognition systems, they have also been shown to exist in natural language processing (NLP) [19] as well as speech recognition systems [20]. As a contemporary example of adversarial perturbations

- *Jonathan Peck and Yvan Saeys are with the Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Ghent, 9000 Belgium as well as the Data Mining and Modeling for Biomedicine group at the VIB Inflammation Research Center, Ghent, 9052 Belgium. Correspondence to:* `Jonathan.Peck@UGent.be`
- *Bart Goossens is with the Department of Telecommunications and Information Processing, IMEC/IPI/Ghent University, Ghent, 9000 Belgium.*

Prompt. Prove that the square of 2 is irrational.
Response. The proof that the square of 2 is irrational is a well-known proof, which can be shown by contradiction. If we assume that the square of 2 is rational, then it can be written in the form of a fraction, where the numerator and denominator are integers. Let's represent the square of 2 as a fraction, where a and b are integers and b is not equal to zero.

$$(2^2) = (a/b)$$

Expanding the left side of the equation:

$$4 = (a/b)$$

We can then multiply both sides of the equation by b:

$$4b = a$$

Since a and b are integers, a must also be divisible by 4. So let's write a as $4x$ for some integer x:

$$4b = 4x$$

Dividing both sides by 4:

$$b = x$$

So, we have $a = 4x$ and $b = x$, which means that $a/b = 4$, and thus the square of 2 can be expressed as a ratio of integers. But this contradicts our assumption that the square of 2 is irrational. Therefore, the square of 2 is irrational.

Fig. 1.2: ChatGPT (version 2023-01-09) is easily tricked into constructing false mathematical proofs.

for NLP, we consider the ChatGPT model by OpenAI,[1] a chatbot based on the Transformer neural network architecture [10]. In figure 1.2, we prompt ChatGPT to "prove" that the square of 2 is irrational. It responds by constructing a proof that resembles that of the irrationality of $\sqrt{2}$, but with the number 4 instead. This is an example of an adversarial perturbation for NLP, where the deletion of a single word ("root" in this case) causes the model to generate nonsense.

Adversarial perturbations were introduced to the deep learning community via the work of [21], who were experimenting with visualizations of class boundaries of deep convolutional neural networks (CNNs). Historically, however, adversarial perturbations had already been an object of study in "classical" ML for many years before this work [18], [22]. The discovery that DNNs also suffered from this problem came as quite a surprise, as many researchers believed DNNs learned smooth representations which would be naturally robust to adversarial perturbations [23]. Indeed, this "smoothness prior" formed the original motivation behind the work of [21]. Specifically, given an input $\boldsymbol{x}$ classified as $f(\boldsymbol{x}) = y$ by the CNN, they considered the problem of finding a closest sample $\boldsymbol{x}'$ such that $f(\boldsymbol{x}') = y_t \neq y$, where y_t is a target class specified beforehand. The idea was that any such $\boldsymbol{x}'$ should be semantically ambiguous under the smoothness prior: if the original sample was classified as `school bus`, for example, and the target class is `ostrich`, then $\boldsymbol{x}'$ should resemble some hybrid interpolation of a school bus and an ostrich. It turns out that this is not what happens: instead, the sample $\boldsymbol{x}'$ is almost identical to $\boldsymbol{x}$, yet the CNN will have high confidence for the target class y_t.

This unintentional finding has severe implications for the trustworthiness of our AI systems. The existence of adversarial perturbations immediately begs the question: what have these models actually learned? How reliable can an AI system be when it can be fooled by tiny perturbations that are barely noticeable to humans, let alone relevant to the task at hand? And, most importantly, can this issue be resolved? The deep learning community has been trying to develop methods for training robust deep learning models that are not vulnerable to adversarial perturbations, but this has proven to be an exceptionally challenging problem. Despite almost a decade of sustained research effort, even our most robust state-of-the-art DNNs cannot (yet) solve it satisfactorily.

Overview

Our goal here is to provide a historical overview of adversarial machine learning (AML), so that the reader understands how the field has evolved and the important lessons the community has learned in the past ten years. As such, it is not our intention to provide an exhaustive overview of the most recent advances; for that, the reader can turn to other surveys [24], [25], [26]. Rather, we have attempted to chronicle the development of the field since its inception in 2013, with emphasis primarily on the major themes underlying the important successes and failures of the field. Therefore, while many recent and state-of-the-art methods will of course be discussed, we also pay significant attention to older ideas, particularly on the defensive side. Many of these have fallen out of favor, either because they did not work or because they were simply forgotten. In the former case, it can be highly instructive to learn why the methods did not work; in the latter case, they may still serve as inspiration for newer, more effective algorithms.

Our focus is specifically on adversarial attacks — also referred to as **evasion attacks** — which are intended to compromise the integrity of a deployed system. Specifically, evasion attacks attempt to manipulate deployed ML systems by corrupting incoming data at inference time. There are many other important classes of attacks which do not fall under this category, but which we cannot adequately cover here. Examples include backdoor or poisoning attacks which aim to insert hidden triggers in the training data in order to influence the behavior of the system, a threat that is particularly relevant today when much of our training data is supplied by third party sources that may not be completely reliable. We refer the interested reader to other surveys on these topics [27], [28].

We begin by formally defining the adversarial robustness problem in section 2. With the mathematical framework established, we survey so-called *adversarial attacks* in section 3; these are algorithms for the efficient generation of adversarial examples under specific threat models. As we shall see, generating adversarial perturbations boils down to a relatively simple optimization problem, and all adversarial attacks essentially just try to solve this same problem under different assumptions. We then proceed in section 4 to survey defenses against adversarial perturbations that have been proposed in the literature. We encourage the reader to pay particular attention to the contents of this section, as the AML community has learned most of its valuable lessons by *failing* to develop effective defenses. It is therefore important

[1] https://openai.com/blog/chatgpt/. Accessed 2023-02-17.

TABLE 2.1: Some common L_p perturbation thresholds

Data set	L_1	L_2	L_∞
MNIST	22	1.5	0.3
Fashion-MNIST		1.5	0.1
CIFAR-10	12	0.5	8/255
CIFAR-100	12	0.5	8/255
ImageNet	60	0.05	4/255

to know beforehand that many of the methods discussed in section 4 have been broken at some point, and offer no real protection against adversarial attacks. Yet we include them nonetheless, because the value of understanding *why* each defense failed cannot be overstated. To this end, we have taken care to elucidate wherever possible the likely reasons behind the failure of every broken defense. In section 5, we move from the more empirical to the more mathematical side of things and survey some theoretical results regarding the robustness of machine learning models. The AML community has of course spent a significant amount of work toward understanding the nature of adversarial perturbations, and these results do provide many valuable insights that can inform future defenses. However, perhaps the most interesting parts of section 5 are those that are missing: specifically, at the time of this writing, we do not yet have a comprehensive understanding of why DNNs are so fragile. Finding an answer to this question is the largest open problem in the field at this time. Section 6 lists what we believe are the most interesting avenues for future research in this field. Finally, section 7 offers some concluding remarks.

2 Adversarial robustness

The basic idea behind adversarial examples as they were conceived by [21] is the following. Starting from a classifier $f : \mathbb{R}^n \to \{1, \dots, k\}$, a "benign" sample $\boldsymbol{x} \in \mathbb{R}^n$ (such as a random sample from the training set), find the minimum set of modifications necessary such that the classification output by f is altered: $f(\boldsymbol{x}) \neq f(\tilde{\boldsymbol{x}})$ where $\tilde{\boldsymbol{x}}$ is the modified "adversarial" sample.

Note that this definition crucially relies on what is considered a "minimal" modification. Indeed, the precise choice of measure is the subject of some debate and controversy within the adversarial ML community. It has become standard practice to use L_p norms for this purpose, *i.e.*, to measure the "size" of the modification as

$$\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |\tilde{x}_i - x_i|^p\right)^{1/p},$$

where p is typically set to 2 or ∞. This particular formalization is known as **the L_p threat model**. Table 2.1 gives an overview of commonly used bounds for popular data sets (assuming the input features are normalized to the range $[0, 1]$). Although these bounds are obviously somewhat arbitrary, the idea behind them is to capture the threshold of "perceptibility." That is, for each data set, these bounds are supposed to guarantee that any additive perturbation of a sample within that radius does not meaningfully alter it and so the predicted label should stay the same.

We now propose the following definition of an *adversarial example*. Given a function $f : \mathcal{X} \to \mathcal{Y}$, constants[2] $\varepsilon_\mathcal{X} > 0$ and $\varepsilon_\mathcal{Y} > 0$ and an element $\boldsymbol{x} \in \mathcal{X}$. An element $\tilde{\boldsymbol{x}} \in \mathcal{X}$ is said to be $(\varepsilon_\mathcal{X}, \varepsilon_\mathcal{Y})$-*adversarial* to f and $\boldsymbol{x}$ if the following properties are satisfied:

$$d_\mathcal{X}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \leq \varepsilon_\mathcal{X} \qquad \text{and} \qquad d_\mathcal{Y}(f(\boldsymbol{x}), f(\tilde{\boldsymbol{x}})) \geq \varepsilon_\mathcal{Y}.$$

Here, $d_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ and $d_\mathcal{Y} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ are similarity measures in the input space $\mathcal{X}$ and output space $\mathcal{Y}$ respectively. These functions do *not* need to satisfy any properties of a metric; they merely need to map pairs of inputs and outputs to non-negative real numbers as a means to quantify their "similarity," however that concept may be defined for the particular task at hand. Mathematically speaking, of course, it would be more convenient to have $d_\mathcal{X}$ and $d_\mathcal{Y}$ satisfy all properties of metrics, as this would facilitate proofs of lower and upper robustness bounds. In practice, however, we may simply not have such a luxury. For example, in the image domain, we may wish to quantify the distance $d_\mathcal{X}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ using some approximation of visual similarity according to the human visual system, which may not yield a valid metric. In the domain of natural language processing, we may be looking at vectorized word embeddings and quantifying their distance according to cosine similarity, which is known not to be a metric.

The above definition easily specializes to the typical L_p threat model used in the vast majority of the literature: we simply take $d_\mathcal{X}(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|_p$, $d_\mathcal{Y}(y, y') = \mathbb{1}[y \neq y']$ and $\varepsilon_\mathcal{Y} = 1$. However, it can also capture much more general classes of adversarial threat models. For instance, if $\mathcal{X}$ is the image manifold corresponding to the data distribution and $d_\mathcal{X}$ is the geodesic distance between two points on this manifold, then an adversarial sample could be a much more complicated transformation of the original sample: we would not be limited to small additive perturbations but could also use rotations, reflections, shearing transforms and other distortions unique to images. It has been known for many years now that L_p norms are a poor proxy for human perception [30], so it is highly desirable to look at the adversarial robustness problem from this more general lens. There have been some papers that attempted to achieve this, such as [31] who introduced **semantic adversarial examples**. These samples can differ greatly from the originals according to L_p metrics, but they are constructed in such a way that the *semantics* of the original samples remain unchanged. Hosseini & Poovendran [31] in particular implement semantic adversarial examples by manipulating the hue and saturation of the images in the HSV color space, rather than working directly with the RGB pixel values. Moving beyond the L_p threat model remains a relatively under-explored area of research, however.

There is also another threat model, known as **patch attacks**, which is becoming increasingly popular [32]. In a patch attack, an adversary is constrained to rectangular patches of fixed dimension. There is no constraint on the per-pixel deviation, so individual pixels may be modified arbitrarily, but the modifications must all be done within a

[2] We may consider an alternative definition where $\varepsilon_\mathcal{X}$ and $\varepsilon_\mathcal{Y}$ are not constant but may depend on $\boldsymbol{x}$ and f. Such input-dependent bounds are sometimes (though rarely) considered in the literature [29].

rectangular patch of fixed size that may appear anywhere in the image. Patch attacks can be more realistic than the L_p threat model in cases where one cannot modify the full image but only localized areas within it, such as when one is holding up a sign in front of a camera.

Based on the above formalization of adversarial examples, we can define the *point-wise adversarial robustness* of a model f at a point $\boldsymbol{x}$ as

$$\rho(f, \boldsymbol{x}) = \inf_{\boldsymbol{x}' \in \mathcal{X}} \{ d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}') \mid d_{\mathcal{Y}}(f(\boldsymbol{x}), f(\boldsymbol{x}')) \geq \varepsilon_{\mathcal{Y}} \}. \tag{2.1}$$

The *expected adversarial robustness* is then the expectation of (2.1) over the data distribution:

$$\rho(f) = \mathbb{E}[\rho(f, \boldsymbol{X})]. \tag{2.2}$$

Clearly, in order for adversarial examples to not exist, it is necessary that $\rho(f) \geq \varepsilon_{\mathcal{X}}$. However, this is not sufficient: even if the expected adversarial robustness is high, there can in principle still exist adversarial examples with very small distance $d_{\mathcal{X}}$ in the input space. A complete lack of adversarial examples around every single data point is typically considered too strong a requirement in the literature, and so adversarial defenses tend to focus on improving the expected robustness $\rho(f)$ rather than the robustness around every single point. Indeed, a high $\rho(f)$ generally implies that whatever adversarial examples still exist will be rare, especially at low distortion.

Aside from the point-wise and expected robustness, the *robust accuracy* is another main quantity of interest in AML. It is simply the accuracy of the model under adversarial perturbations, and it is usually approximated by computing the empirical accuracy on a set of adversarial examples generated by some attack. Of course, such evaluations can only yield an *upper bound* on the "true" robust accuracy. As we shall see, finding good bounds on the robust accuracy of a given model is not trivial, and remains an active area of research.

3 Adversarial attacks

In this section, we survey some of the notable adversarial attacks on DNNs that have been proposed since the work of [21]. These attacks all attempt to efficiently solve the following optimization problem in different ways:

$$\begin{aligned} \tilde{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}' \in \mathcal{X}} \ & d_{\mathcal{Y}}(f(\boldsymbol{x}), f(\boldsymbol{x}')) \\ \text{subject to } & d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}') \leq \varepsilon_{\mathcal{X}}. \end{aligned} \tag{3.1}$$

That is, starting from some initial point $\boldsymbol{x} \in \mathcal{X}$, we wish to find elements $\tilde{\boldsymbol{x}} \in \mathcal{X}$ which maximize the dissimilarity in the output space, $d_{\mathcal{Y}}(f(\boldsymbol{x}), f(\tilde{\boldsymbol{x}}))$, while maintaining high similarity in the input space. This bound is determined by the parameter $\varepsilon_{\mathcal{X}}$, which is often referred to as the "budget" or "strength" of the attack. In the simplest case, adopting the L_p threat model and instantiating $d_{\mathcal{Y}}$ with the cross-entropy loss, we obtain the most commonly used formulation of the adversarial robustness problem:

$$\begin{aligned} \tilde{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}' \in \mathcal{X}} \ & \mathcal{L}(f(\boldsymbol{x}), f(\boldsymbol{x}')) \\ \text{subject to } & \|\boldsymbol{x} - \boldsymbol{x}'\|_p \leq \varepsilon_{\mathcal{X}}. \end{aligned} \tag{3.2}$$

However, even (3.2) is already a complicated non-convex problem when f is a deep neural network, so specialized techniques are required to solve it. As we will see in this section, it is by no means straightforward to design an effective adversarial attack. New adversarial attacks are still proposed on a regular basis, each differing slightly in their underlying assumptions about what causes adversarial vulnerability and how to best exploit it. The importance of strong adversarial attacks in this field cannot be overstated. Indeed, the development of adversarial defenses has been incredibly synergistic with the development of new attacks, a phenomenon often referred to as the *arms race*: researchers will propose a defense in response to a new adversarial attack, and other researchers will subsequently analyze this defense for flaws and publish improved attacks that break it. To obtain an accurate picture of where we stand regarding adversarial robustness, strong attacks are needed which do not over-estimate model robustness. This problem of *over-estimation* has been a staple of the field since the early years, and is still regarded as an open problem [33].

In general, adversarial attacks may be roughly classified according to the amount of information about the target they require. The taxonomy we will use in this work is as follows:

I. **White-box, gradient-based.** The attack requires complete knowledge of the model to be attacked, including its exact architecture and weights as well as the ability to evaluate the gradients of the model on arbitrary inputs.

II. **White-box, gradient-free.** The attack requires knowledge of the architecture and weights of the model used by the victim, but does not need its gradients.

III. **Black-box, surrogate-based.** The attack does not need the original model, but requires access to a *surrogate*, *i.e.*, a model trained on similar data as the target.

IV. **Black-box, score-based.** The attack does not need knowledge of the model, but requires the ability to obtain the probability scores assigned by the model on arbitrary inputs.

V. **Black-box, decision-based.** The attack does not need knowledge of the model, but requires the ability to obtain the predicted class of arbitrary inputs.

Aside from the categories above which quantify the level of information required for the attack to operate, we can further characterize adversarial attacks according to the following properties:

- **Targeted.** The attack requires the user to specify a target class y_t beforehand. The adversarial perturbation will then be optimized to push the sample towards this specific class. The attack is successful only if $f(\tilde{\boldsymbol{x}}) = y_t$.
- **Untargeted.** The attack does not require the user to specify a target class. The adversarial perturbation is merely optimized to induce a difference in classifications, *i.e.*, $f(\tilde{\boldsymbol{x}}) \neq f(\boldsymbol{x})$. The precise output of the model on the adversarial sample is not relevant.

Whether a user prefers a targeted attack or an untargeted one entirely depends on the use case. For example, if a user wishes to impersonate a specific employee to fool a biometric security system, a targeted attack is required since a specific classification result must be reached. On the other hand, if the user wants to cause a computer vision system to not recognize a specific object, an untargeted attack suffices since the exact result is irrelevant as long as it differs from the ground truth. Furthermore, any targeted attack can (in principle) be converted to an untargeted one by simply running the attack for every class except the original and returning one of the successful results. However, attacks that were designed to be untargeted will generally be much more efficient than this.

The earliest adversarial attacks developed against DNNs were category I: white-box and gradient-based. As time went on, of course, researchers crafted increasingly sophisticated algorithms that were able to attack DNNs in ever more restricted settings. We discuss a few of these attacks below.

3.1 L-BFGS

One of the first attacks to be created for DNNs is known as the *L-BFGS attack*. It was proposed by [21] and so-named because it is merely an application of the limited-memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimization algorithm [34] to the following problem:

$$\begin{aligned}&\min_{\boldsymbol{\delta}}\ c\|\boldsymbol{\delta}\| + \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y_t, f)\\&\text{subject to } \boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^n.\end{aligned} \tag{3.3}$$

Here, y_t is the target class which must be specified beforehand. The parameter $c > 0$ is tuned via a line search so that the smallest value is used for which the minimizer $\boldsymbol{\delta}$ of (3.3) satisfies $f(\boldsymbol{x}+\boldsymbol{\delta}) = y_t$. It is a **targeted white-box gradient-based** attack, as we must specify a target class y_t and L-BFGS requires access to the gradients of the objective function, which includes the loss of the model on arbitrary samples.

3.2 Mimicry attack

Concurrently to [21], [35] were also working on adversarial examples in the context of DNNs and support vector machines. They developed a general adversarial attack algorithm that tries to find the smallest additive perturbation to an input such that the classification of the model changes, but the sample also remains within the support of the original data distribution. Formally, they consider the following problem:

$$\begin{aligned}&\operatorname*{argmin}_{\tilde{\boldsymbol{x}}\in\mathcal{X}}\ \hat{f}(\tilde{\boldsymbol{x}}) - \frac{\lambda}{m}\sum_{i:y_i=-1}^{m}\kappa\left(\frac{\tilde{\boldsymbol{x}}-\boldsymbol{x}_i}{h}\right)\\&\text{subject to } d_{\mathcal{X}}(\boldsymbol{x},\tilde{\boldsymbol{x}}) \leq \varepsilon_{\mathcal{X}}.\end{aligned} \tag{3.4}$$

Here, $\hat{f}$ is a surrogate model trained by the adversary, $\lambda > 0$ is a regularization parameter, m is the number of benign samples available to the adversary, κ is a kernel and h is its bandwidth. It is a **targeted black-box surrogate-based** attack. Biggio et al. [35] already considered the possibility that the adversary may not have full access to the model f used by the victim and therefore account for the fact that we may be using a surrogate model that we trained ourselves, with the hopes that adversarials generated for the surrogate will also work on the real target. They also account for the fact that the norm constraint by itself may not be sufficient to guarantee that the adversarial sample will lie close to the original data manifold. Since this might cause the adversarials to be detected by the victim, a regularization term is added to the objective which penalizes the distance of the generated sample $\tilde{\boldsymbol{x}}$ to the data samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ according to a kernel density estimate. They refer to this technique as "mimicry." Although the attack can already cause misclassification with almost imperceptible modifications, allowing the algorithm to optimize to complete convergence produces samples that visibly morph into their target classes. This is in stark contrast to most other attacks, which will never produce samples that resemble their target classes at all.

3.3 Fast gradient sign

The *fast gradient sign* (FGS) attack was proposed by [36] in order to make the generation of adversarial examples much more efficient than the L-BFGS method. It is an **untargeted white-box gradient-based** attack. To design it, [36] start from the following optimization problem:

$$\max_{\boldsymbol{\delta}}\ \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y, f) \text{ subject to } \|\boldsymbol{\delta}\|_\infty \leq \varepsilon_{\mathcal{X}}. \tag{3.5}$$

Note that (3.5) is just (3.2) specialized to the L_∞ norm. They then consider a first-order Taylor approximation of the loss term:

$$\mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y, f) \approx \mathcal{L}(\boldsymbol{x}, y, f) + \boldsymbol{\delta}^\intercal \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y, f).$$

Assuming this linear approximation is accurate, we can choose the perturbation as follows:

$$\boldsymbol{\delta} = \varepsilon_{\mathcal{X}} \operatorname{sgn} \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y, f). \tag{3.6}$$

We then clearly have $\|\boldsymbol{\delta}\|_\infty = \varepsilon_{\mathcal{X}}$, satisfying our norm constraint. Furthermore, if we plug this value into the Taylor approximation, we obtain

$$\mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y, f) \approx \mathcal{L}(\boldsymbol{x}, y, f) + \varepsilon_{\mathcal{X}}\|\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y, f)\|_1.$$

If the original sample $(\boldsymbol{x}, y)$ is not a stationary point of the loss, then the 1-norm of the gradient will likely be proportional to the dimensionality of the data. Therefore, in high dimensions, merely following the sign of the gradient vector can lead to large increases in loss even though the input perturbation is very small. Indeed, experiments using the FGS method were highly successful, achieving very high error rates with relatively small values of $\varepsilon_{\mathcal{X}}$. It was also very fast, since the computation of the perturbation $\boldsymbol{\delta}$ in (3.6) requires only a single backward pass through the network per sample. These factors combined made FGS one of the most popular adversarial attacks for many years.

3.4 Projected gradient descent

Similar to L-BFGS, the *projected gradient descent* (PGD) attack is named after an existing general-purpose optimization algorithm that has simply been specialized to the crafting of adversarial perturbations. Specifically, PGD finds an adversarial example by iterating the following update:

$$\tilde{\boldsymbol{x}}_{t+1} \leftarrow \mathcal{P}_{\boldsymbol{x}+\mathcal{S}}(\tilde{\boldsymbol{x}}_t + \alpha \operatorname{sgn} \nabla_{\boldsymbol{x}_t}\mathcal{L}(\boldsymbol{x}_t, y, \theta)). \tag{3.7}$$

Here, $\alpha > 0$ is a user-specified constant, $\mathcal{S}$ is an appropriate L_p norm ball centered at the origin and $\mathcal{P}_U(\boldsymbol{x})$ denotes the projection of $\boldsymbol{x}$ onto U. Depending on the choice of norm, the projection operator $\mathcal{P}_{\boldsymbol{x}+\mathcal{S}}$ can take very different forms. In general, it is the solution to an optimization problem that finds a point $\boldsymbol{u} \in U$ closest to $\boldsymbol{x}$ according to the particular similarity measure:

$$\mathcal{P}_U(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{u} \in U} \; d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{u}).$$

If $d_{\mathcal{X}}$ is the L_∞ distance, the projection can be accomplished merely by clipping the components of the perturbed sample within the admissible range; for L_2, an orthogonal projection onto the ball $\boldsymbol{x} + \mathcal{S}$ must be carried out. For L_1, the projection can be computed in log-linear time in the input dimensionality [37]. Although the optimization can start from the unmodified original point $\boldsymbol{x}_0 = \boldsymbol{x}$, it is common to use *random restarts* where $\boldsymbol{x}_0 = \mathcal{P}_{\boldsymbol{x}+\mathcal{S}}(\boldsymbol{x} + \xi)$ and ξ is sampled from some tractable distribution such as uniform or Gaussian. The attack can then be run multiple times using different independent samples of ξ, where the best result across all restarts is returned as the final adversarial.

The PGD attack is an **untargeted white-box gradient-based attack**. It was made famous by [38], who performed an extensive analysis of its theoretical and empirical properties. They argued that PGD is essentially an optimal first-order adversary, in the sense that no other *efficient* attack algorithm that uses only gradients of the loss to construct perturbations could outperform it. Stronger adversaries would therefore either have to use higher-order information, which is notoriously expensive to compute for DNNs, or they would need to perform more expensive computations using the first-order gradient information. Either way, they would be considerably less efficient than PGD. Based on this insight, they used PGD to construct robust models for MNIST and CIFAR-10. These models remained remarkably robust for some time, although the CIFAR-10 model was eventually broken.

Croce et al. [39] proposed AutoPGD (APGD) as an improvement on the original PGD formulation. APGD incorporates a momentum term in the update (3.7), which is known to facilitate optimization [40]. APGD also does not use a fixed step size α, but rather dynamically decreases the step size when the loss has seemingly stagnated. In their experiments, [39] used APGD to successfully break many defenses that were recently proposed at major conferences at the time. Crucially, they were able to achieve these results with little to no tuning of the hyperparameters of the APGD algorithm, suggesting that this attack should become a new baseline in adversarial defense going forward. Accordingly, since its initial proposal, APGD has been rapidly adopted as a standard benchmark attack for adversarial defenses. It is included in the AutoAttack suite, a popular library for benchmarking robustness.[3]

3.5 The Carlini-Wagner attacks

Carlini et al. [41] originally proposed three different attacks specifically to target *defensive distillation* [42], which was one of the few promising adversarial defenses at the time. They succeeded in reducing the robust accuracy of defensive distillation to 0%, a result that gained immediate notoriety as it was the first complete "break" of an established adversarial defense. The three attacks were each designed for a specific threat model, in this case L_0, L_2 and L_∞ respectively. The basic insight that led to the C&W attacks is that one must very carefully craft an appropriate objective function in order to create effective adversarial samples. Specifically, [41] propose a general framework where one starts from the following optimization problem:

$$\min_{\boldsymbol{\delta}} \; d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta}) + \lambda g(\boldsymbol{x} + \boldsymbol{\delta}). \tag{3.8}$$

Here, $\lambda > 0$ is a tunable parameter and g is a function with the property that

$$f(\boldsymbol{x} + \boldsymbol{\delta}) = y_t \iff g(\boldsymbol{x} + \boldsymbol{\delta}) \leq 0. \tag{3.9}$$

If g satisfies (3.9), then minimizing g is a consistent proxy for optimizing $\boldsymbol{\delta}$ such that the target classifier f outputs the desired class y_t. The basic C&W attacks are all therefore **targeted white-box gradient-based** types, although [41] proposed untargeted variants as well.

Clearly, the main design consideration for the C&W attacks is the specific choice of g in (3.8). Carlini et al. [41] experiment with many different options, but the most effective one turned out to be

$$g(\boldsymbol{x}') = \max\left\{ \max_{i \neq t} Z_i(\boldsymbol{x}') - Z_t(\boldsymbol{x}'), -\kappa \right\}.$$

Here, $\boldsymbol{Z}(\boldsymbol{x})$ represents the vector of logits of the model on the given sample $\boldsymbol{x}$ and $\kappa \geq 0$ is a constant that controls the confidence of the resulting adversarial. Essentially, the C&W attacks optimize the perturbation $\boldsymbol{\delta}$ such that the resulting adversarial $\boldsymbol{x} + \boldsymbol{\delta}$ has a higher logit value for the target class than any of the other classes, leading to a targeted misclassification. Increasing κ causes the attack to generate adversarial samples with higher confidence in the target class, which can help them transfer to other models.

We can interpret (3.8) as the Lagrangian relaxation of the constrained optimization problem

$$\min_{\boldsymbol{\delta}} d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta}) \text{ subject to } g(\boldsymbol{x} + \boldsymbol{\delta}) \leq 0. \tag{3.10}$$

Due to (3.9), problem (3.10) is itself a relaxation of

$$\min_{\boldsymbol{\delta}} d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta}) \text{ subject to } f(\boldsymbol{x} + \boldsymbol{\delta}) = y_t, \tag{3.11}$$

which is the quintessential optimization problem for targeted adversarial attacks. Now, if $\boldsymbol{\delta}$ is any feasible solution to (3.11), it holds that

$$d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta}) + \lambda g(\boldsymbol{x} + \boldsymbol{\delta}) \leq d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta}).$$

That is, for any feasible solution $\boldsymbol{\delta}$ (*i.e.*, such that $f(\boldsymbol{x} + \boldsymbol{\delta}) = y_t$), the objective value of (3.8) is a lower bound on the objective value of (3.11). To obtain the best possible approximation of an optimal solution to (3.11) via the relaxation (3.8), this bound needs to become as tight as possible. This means that the parameter λ must be minimized, because $g(\boldsymbol{x} + \boldsymbol{\delta}) \leq 0$. Indeed, [41] confirm this empirically, and therefore extend their attack algorithm with a modified binary search procedure to choose the minimal constant λ for which the optimizer can still find feasible solutions.

The Carlini-Wagner attacks were the first attacks that broke an established defense that seemed highly promising: at the time, defensive distillation was able to reduce attack success rates from 95% (*i.e.*, 5% robust accuracy) to 0.5%

[3] https://github.com/fra31/auto-attack. Accessed 2023-02-15.

(99.5% robust accuracy). The C&W attacks, on the other hand, obtained a success rate of 100% against this defense. They are relatively slow, however, requiring many iterations to reach a good solution. Moreover, since they optimize over the logit space, high-confidence adversarials produced by the C&W attacks may lead to "over-optimized" perturbations that can be easily identified via IQR-thresholding of the logit values, since such values are atypical of benign samples [43].

3.6 Sparse attacks

The most popular choices of norm in (3.1) are L_2 and L_∞, which lead to adversarial perturbations that are "dense" (most components are non-zero) but imperceptible. However, there has also been much work on *sparse perturbations* where most components are constrained to be close to zero. For sparse perturbations, we essentially just specialize (3.1) to the L_0 norm. However, this makes the optimization problem NP-hard to solve in general, so approximations are often needed. A common trick is to use the L_1 norm instead, which does allow for tractable optimization but may not always generate sparse perturbations. Su et al. [44], for instance, implement the *one-pixel attack*, a **black-box score-based** attack which modifies only a single pixel. Modas et al. [45] introduce *SparseFool*, an extension of the older *DeepFool* algorithm [46], which works by iteratively approximating the decision boundary of the model using an affine hyperplane and solving a linear program to obtain the optimal sparse perturbation. It is an **untargeted white-box gradient-based** attack. Similarly, [47] propose the *structured adversarial attack* which imposes sparsity on smaller groups of pixels rather than the image as a whole. This method is a **targeted white-box gradient-based** attack with the interesting advantage that the generated perturbations, while still imperceptibly small, appear to mimic the structure of the target class. Additionally, [37] propose an L_1 variant of APGD which is meant to generate sparse perturbations as well. However, as an approximation to the L_0 problem, the perturbations may not always be sparse [48].

3.7 Randomized gradient-free attack

The randomized gradient-free attack [49] is notable for being perhaps the only existing **white-box gradient-free** attack: it requires access to the entire model specification yet does not use gradient information. The algorithm exploits the fact that DNNs with ReLU activation functions divide their input spaces into *linear regions*, *i.e.*, connected subsets where the DNN reduces to a linear function [50]. Croce et al. [49] make use of an explicit construction of these linear regions in order to find minimal perturbations that cause a given sample to cross a decision boundary. The resulting optimization problem is a quadratic program that requires only the parameters of the network, but not its gradients with respect to arbitrary inputs.

One might wonder why it is useful to make the distinction between category I and category II if the latter category is so sparsely populated. The reason is that the vast majority of category I attacks work by optimizing the additive perturbation $\boldsymbol{\delta}$ using gradient descent over some function of the loss of the model. Thus, they crucially rely on the gradients of the loss with respect to the model parameters or input. However, as shown by [51], category I attacks can severely overestimate the true robustness of their targets due to a phenomenon known as *gradient masking*, which we discuss further in section 4. Aside from the problem of gradient masking, the distinction between category I and II attacks is also useful because certain types of neural networks can be shown to be immune to gradient-based attacks. Carbone et al. [52], for example, obtained a very interesting result proving that Bayesian neural networks are immune to gradient-based adversarial attacks in the infinite data limit, because the gradient of the loss with respect to any sample from the data distribution is zero. This result also seems to hold approximately for Bayesian neural networks trained on finite data.

3.8 Black-box attacks

Due to their practical usefulness, many black-box attacks have been proposed over the years, and it is not feasible to survey them all here. We will therefore conclude our discussion of black-box attacks with a few "honorable mentions," to which we will not dedicate an entire subsection.

The *Simple Black-box Attack* (SimBA) is a **black-box score-based** attack proposed by [53]. As its name implies, it is an exceedingly simple attack to implement and carry out: in order to perturb a given sample, SimBA randomly samples a vector from a pre-defined orthonormal basis and either adds or subtracts it to the input. If the user chooses the standard basis in $\mathbb{R}^n$, SimBA will perturb only a single randomly chosen pixel at a time. A more interesting choice of basis is the discrete cosine transform (DCT) basis, which makes the algorithm very effective and query-efficient.

Chen et al. [54] proposed the *Zeroth Order Optimization* (ZOO) attack, which is **black-box, score-based** (both targeted and untargeted). ZOO can be viewed as an attempt to "lift" the problem of attacking a black-box model to the problem of attacking a white-box one. It accomplishes this by using the symmetric difference quotient to numerically estimate the gradient of the target model and optimizing a loss function similar to the Carlini-Wagner attack.

The *square attack* by [55] is notable for being one of the most efficient and strong **untargeted black-box score-based** attacks proposed to date. It has been included in the AutoAttack framework, a comprehensive benchmark for evaluating robustness of models proposed by [39] at ICML 2020. This framework is the foundation of the RobustBench leaderboard,[4] which has become the *de facto* standard reference for recording state of the art robustness results.

Brendel et al. [56] proposed the *boundary attack*, a **black-box decision-based** type (targeted and untargeted). Together with *GeoDA* [57], these attacks are notable for being the only category V types that appear to have been described in the literature to the best of our knowledge.

3.9 Real-world attacks

The attacks discussed in the previous sections are all focused on a "lab setting." That is, they all assume we can perfectly manipulate a given input and this input will be provided to

[4] https://robustbench.github.io. Accessed 2023-02-03.

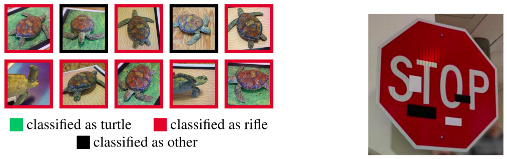


Fig. 3.1: Examples of real-world adversarial attacks. Left: the “adversarial turtle” by [58]. Right: the adversarial stop sign by [59].

the model *exactly* as we crafted it. For many applications of machine learning, this is unrealistic: think, for example, of an autonomous vehicle that uses computer vision to recognize traffic signs. The inputs to this system are images of the road, but these images can vary significantly from moment to moment due to weather, obstacles, physical damage, etc. To address these issues, the adversarial ML community has developed *real-world attacks* which are designed to cope with typical distortions to which the inputs might be subjected. Seminal work in this direction was done by [58], who proposed a technique called *expectation over transformation* (EOT). This technique has since become widely used not only to create real-world adversarial examples but also to increase robustness of models in general (even in the lab setting). The idea behind EOT is to model the distortions to which the input might be subjected as part of the optimization procedure. Formally, [58] define a distribution T of possible transformations. A transformation is simply a $\mathcal{X} \to \mathcal{X}$ function, so T is a distribution on *functions* of the input. Before being processed by the vision system, an input $\boldsymbol{x} \in \mathcal{X}$ will be affected by some random transformation $t \sim T$. Hence, the system will actually observe $t(\boldsymbol{x})$ instead of $\boldsymbol{x}$ itself. The objective of EOT is therefore to solve a slightly different optimization problem:

$$\begin{aligned} \max_{\tilde{\boldsymbol{x}}\in\mathcal{X}} \quad & \mathop{\mathbb{E}}_{t\sim T}[\log \Pr[y_t \mid t(\tilde{\boldsymbol{x}})]] \\ \text{subject to} \quad & \mathop{\mathbb{E}}_{t\sim T}[d_{\mathcal{X}}(t(\tilde{\boldsymbol{x}}), t(\boldsymbol{x}))] \leq \varepsilon. \end{aligned} \tag{3.12}$$

Here, y_t is a chosen target class and $\Pr[y \mid \boldsymbol{x}]$ refers to the probability estimated by the target model for class y given input $\boldsymbol{x}$. In practice, (3.12) is solved using a Lagrangian relaxation, resulting in the stochastic optimization problem

$$\max_{\tilde{\boldsymbol{x}}\in\mathcal{X}} \mathop{\mathbb{E}}_{t\sim T} \left[\log \Pr[y_t \mid t(\tilde{\boldsymbol{x}})] - \lambda d_{\mathcal{X}}(t(\tilde{\boldsymbol{x}}), t(\boldsymbol{x}))\right], \tag{3.13}$$

where $\lambda > 0$ is a hyperparameter. Athalye et al. [58] used projected gradient descent to solve (3.13), so their original attack is a **targeted white-box gradient-based** type.

The EOT algorithm allows for some truly impressive adversarial attacks. For instance, the main contribution of [58] is to specialize EOT for affine transformations, allowing them to craft adversarial textures that can be applied to real-world (3D printed) objects, such as the famous “adversarial turtle” shown in figure 3.1. These textures cause the targeted vision system to misclassify the object even under various common real-world distortions, such as changes in pose, surrounding environment, lighting and camera angles.

Concurrently to the work of [58], [59] proposed real-world adversarial attacks on autonomous vehicle systems using a very similar framework as EOT which they called *robust physical perturbations* (RP$_2$). A typical example of this attack is shown in figure 3.1. The RP$_2$ method allows the creation of adversarial examples using simple black and white stickers that are easily applied to physical objects and which consistently fool computer vision systems under real-world conditions. Like EOT, it is a **targeted white-box gradient-based** attack.

3.10 Universal adversarial perturbations

Bringing the phenomenon of adversarial examples to its logical extreme, [60] introduced *universal adversarial perturbations* (UAPs). A UAP is a *single* perturbation $\boldsymbol{\delta}$ with the property that $\boldsymbol{x} + \boldsymbol{\delta}$ is adversarial for *any* model $f : \mathcal{X} \to \mathcal{Y}$ and *any* sample $\boldsymbol{x} \in \mathcal{X}$. They achieve this by creating adversarial perturbations for individual data samples across a large data set and simply adding them all together with appropriate clipping to stay within the designated budget. To create the individual adversarial perturbations, any existing attack can in principle be used, so the specific category to which this attack belongs can be subject to some debate. Based on the original implementation by [60], one could classify it as **untargeted black-box surrogate-based**. Naturally, the discovery of universal perturbations attracted much research attention, both on the attacking side as well as the defensive side. Since the publication of [60], many new methods have been proposed. We refer the interested reader to other surveys such as [61].

3.11 A note on adversaries

An important distinction that we have neglected so far is that between *static* and *adaptive adversaries* [62]. A **static adversary** (or “oblivious adversary”) is not aware of the particular defense used by their target, and is therefore restricted to using existing adversarial attacks that were not specifically designed to circumvent that defense. An **adaptive adversary** is aware of the defense used by the target, and can adapt their attacks accordingly. Many adversarial defenses in the past failed to provide meaningful robustness and over-stated their results primarily because they only evaluated against static adversaries. As we shall see in the next section, it is often relatively easy to defend against *known* adversarial attacks, since the defense can then be based on violating the assumptions made by the attacks in question. This is why many researchers [63] argue that evaluations on adaptive adversaries are necessary to properly gauge the true robustness of any newly proposed method. That said, we end this note with an important commentary on how the field of AML has historically handled adversarial threat models. Specifically, it seems there has always been considerable confusion about the distinction between a *white-box* (category I) adversarial attack and an *adaptive* adversary. There have been many papers that claimed to perform a robustness evaluation against an adaptive adversary in accordance with the recommendations by [63], but in actuality merely performed an evaluation against a white-box attack. These are not the same thing: a static adversary can employ any category of attacks, including category I; similarly, an adaptive adversary can just as well use category V attacks. There is no inherent connection between the type of adversary (static or adaptive) and the category of attack used. The category is merely an indication of what information is required to conduct the attack. A white-box attack does not necessarily use this information in a manner that appropriately adapts to a specific defense. Conversely, an adaptive attack does not necessarily need white-box access to the model. One particularly problematic consequence of this confusion is now known as *gradient masking*, which we discuss in more detail in section 4.

4 Adversarial defenses

Like adversarial attacks, the defenses described in the literature can be roughly classified into a few distinct groups. The taxonomy we propose here is based on the general strategy used to defend a model:

- **Purification**. The defense attempts to remove the adversarial perturbation from the input and restore the sample to its original form, which can then be processed by the underlying model as usual.
- **Detection**. These defenses focus on detecting adversarially manipulated inputs before they are processed by the model. They do not attempt to correct the prediction of the model in any way; instead, they extend the set of model outputs to include a special "reject" signal which implies that the model is likely not reliable on the given sample.
- **Hardening**. Hardening defenses do not attempt to remove the adversarial noise nor to detect when manipulation might have occurred. Instead, these defenses seek to make the model *inherently invulnerable* to adversarial perturbations.

Apart from these broad categories, adversarial defenses can also be **randomized** or **deterministic**. These distinctions are relevant, since randomized defenses need to be evaluated differently from deterministic ones [39]. In the literature, the distinction is also often made between defenses that are **certified** versus those that are not. A certified adversarial defense is one for which a *certificate of robustness* can be produced. This is a mathematical proof that shows that no perturbation up to a given radius $\varepsilon_{\mathcal{X}}$ could possibly change the outcome of the classifier. This also must be taken into account during evaluation, because there is no point in trying to attack a certified defense using perturbations that lie within its certified radius.

There are exceptions to this, however. In practice, many certificates of robustness are *probabilistic*, in the following sense. They assume that the adversary creates adversarial examples by first sampling a natural sample $\boldsymbol{x}$ from the data distribution on which the model is trained and then finds a perturbation $\boldsymbol{\delta}$ to cause misclassification. Probabilistic certificates of robustness then usually give the guarantee that, under this mode of operation, there is only a small probability (taken over the data distribution) that there exists such a perturbation $\boldsymbol{\delta}$. Although probabilistic certificates are clearly better than no certificates at all, they do not completely rule out the existence of adversarials within their radius of certification. Furthermore, these certificates typically suffer from the same problem as purification-based defenses: their guarantees become meaningless when the adversary does not respect their chosen mode of operation, as with rubbish examples [64].

Whatever their underlying philosophy, all adversarial defenses are direct responses to the adversarial robustness problem (3.1) or specialized versions thereof. As such, the fundamental goal of an adversarial defense is to make a classifier *invariant* to perturbations generated as solutions to (3.1). Typically, this is done by maximizing the expected robustness $\rho(f)$ defined in (2.2). However, it is clear that $\rho(f)$ cannot become too large, as otherwise the classifier may become too invariant and not change its decision when it is supposed to. This implies a trade-off between accuracy and robustness, a question that has indeed become the subject of much research in the field [65]. We discuss this further in section 5.

4.1 Adversarial training

The most successful defense that has been described in the literature as of this writing is *adversarial training* (AT). The basic idea is straightforward: simply include adversarial examples in the training pipeline as a form of data augmentation. In practice, of course, the devil is in the details, and the way we implement AT has changed throughout the years as the field has gained more insights into the factors and design decisions that can make or break the method.

The first variant of AT was proposed by [36], in the same paper where they introduced the FGS attack. Since FGS is so efficient, their AT scheme took the form of a regularized loss function that essentially performs the FGS attack against all samples in the mini-batch and then updates the parameters of the model according to a weighted average of the loss on the original and adversarial samples:

$$\mathcal{L}(\boldsymbol{x}, y, f) = \alpha J(\boldsymbol{x}, y, f) + (1 - \alpha) J(\tilde{\boldsymbol{x}}, y, f),$$

where

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \varepsilon \operatorname{sgn} \nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y, f).$$

Here, $\alpha \in [0, 1]$ and J is the cross-entropy loss. Due to its extremely high efficiency and seemingly high effectiveness, FGS AT was the preferred method for increasing robustness for some time. Later, however, [66] showed that FGS AT suffers from a "label leaking" effect: because the FGS adversarials are computed using the gradient of the loss on the *true* label, statistical artefacts are introduced that encode the true label in the adversarial image in a form that can be easily detected by DNNs. Thus, using FGS AT, adversarially trained models often have much higher accuracy on FGS adversarials than on the original, unaltered samples. Later, [41] also found that FGS is actually a very weak attack, as FGS AT provides no real protection against stronger attacks such as PGD or even an iterated version of FGS. These findings caused FGS to fall out of favor both as an attack as well as the basis for any defense.

The next major innovation in the design of AT schemes came from [38], who proposed a more principled way to implement it. They characterized the problem as one of *robust optimization* (RO) [67]:

$$\theta^\star = \operatorname*{argmin}_{\theta} \; \mathbb{E}\left[\max_{\boldsymbol{\delta} \in B_\varepsilon} \mathcal{L}(\boldsymbol{X} + \boldsymbol{\delta}, Y, \theta)\right], \tag{4.1}$$

where B_ε is the ε ball around the origin in the norm of choice. That is, instead of minimizing the expected loss on a data set of i.i.d. samples, we try to minimize the loss under *worst-case* norm-bounded additive perturbations of the input. The RO view therefore casts AT as a bi-level optimization problem:

1) **Inner maximization**. In this step, the current mini-batch of samples is subjected to an adversarial attack with respect to the current model parameters.
2) **Outer minimization**. After the adversarial examples have been constructed, the original mini-batch is replaced with their adversarial counterparts. These are then used in a subsequent optimizer to minimize the loss.

Madry et al. [38] originally proposed PGD for the inner maximization, using various arguments to support the idea that PGD is an optimal first-order adversary. Using PGD AT, they constructed robust models for MNIST and CIFAR-10 that maintained state of the art robust accuracy in the L_∞ threat model for some time. They publicly launched the MNIST challenge[5] and the CIFAR-10 challenge[6] where researchers were invited to submit adversarial examples against their models, essentially crowd-sourcing their robustness assessments. The CIFAR-10 model was considered broken by December 2017 using a variant of one of the C&W attacks, which reduced robust accuracy at $L_\infty = 8/255$ to 47.76%. As of this writing, however, the MNIST model remains highly robust: even against the best known attack, it still has a robust accuracy of 88% at $L_\infty = 0.3$.

Thanks to the work by [38], by the end of 2017 the field had realized that AT could be an incredibly powerful method of defense. The only real issue was *scale*: training a robust model on a large data set such as ImageNet using the method proposed by [38] severely increased training times compared to standard training. This caused researchers to experiment with more efficient methods, such as generating the adversarials using pre-trained generative models [68], but these approaches tend to be unstable on complex large-scale data sets. Other work has focused on making the base AT algorithm itself more efficient, with some remarkable recent results [69], [70]. The majority of work in AT considers the L_2 and L_∞ threat model, as it has proven to be considerably more difficult to adversarially train models against L_1 perturbations without catastrophic overfitting. Some recent progress has been made in this direction, however, such as the *Fast-EG-L_1* method proposed by [48].

The precise classification of AT according to the taxonomy we proposed here depends on the exact implementation. The FGS AT and PGD AT variants, for example, could be classified as **non-certified deterministic hardening** defenses, since they cannot provide robustness certificates, do not introduce any additional randomness as part of the defense and aim to make the model inherently more robust rather than focusing on detecting or purifying adversarial noise. Other works utilize more advanced ideas from optimization theory in order to implement a training regime that *can* provide robustness certificates, such as [71]. Typically, such work goes beyond the original formulation (4.1) by [38] and instead considers a *distributionally* robust optimization (DRO) problem:

$$\theta^\star = \operatorname*{argmin}_\theta \sup_{Q \in \mathcal{Q}} \mathop{\mathbb{E}}_{Q} \left[\mathcal{L}(\boldsymbol{X}, Y, \theta)\right]. \tag{4.2}$$

Here, $\mathcal{Q}$ is a class of distributions centered around the original data distribution Q_0. The most common form of DRO takes a *Wasserstein ball* around Q_0,

$$\mathcal{Q} = \{Q \mid W_c(Q_0, Q) \leq \varepsilon\},$$

where W_c is the Wasserstein metric

$$W_c(P, Q) = \inf_{M \in \Pi(P,Q)} \mathop{\mathbb{E}}_{M}[c(Z, Z')].$$

The Wasserstein metric $W_c(P, Q)$ between two distributions P and Q is parameterized by a *cost function* c, which takes samples from both distributions P and Q and outputs a non-negative real number. The set $\Pi(P, Q)$ is the collection of all *couplings* of P and Q, *i.e.*, the set of all joint distributions where the respective marginals are P and Q. The Wasserstein metric therefore computes the smallest expected cost $c(z, z')$ when (z, z') is sampled over all possible joint distributions with marginals P and Q. The DRO problem (4.2) then consists of minimizing the worst-case expected loss when the data $\boldsymbol{X}$ can be sampled from *any* distribution Q in a Wasserstein ball around Q_0. This essentially formalizes robustness against small distributional shifts, from which the method derives its name. DRO methods are typically **certified deterministic hardening** defenses.

Another important example of an AT defense is TRADES [72]. Like FGS AT, the concrete implementation of this method takes the form of a regularization added to the loss function. By varying the regularization parameter, the method allows for trade-offs between accuracy and robustness. Although TRADES admits a theoretical upper bound on its robust risk [72, theorem 3.1], this bound is difficult to compute in practice, so we cannot yet classify it as a certified method since it does not produce useful certificates. TRADES gained notoriety for reaching first place in the Adversarial Vision Challenge at NeurIPS 2018.[7] Due to its computational efficiency and impressive robustness results, TRADES has become a popular defense method in recent years.

Adversarial training is not without its issues, however: aside from the often much increased computational complexity, there is the problem of *robust overfitting* described by [73]. There appears to be a consistent and significant gap between the best robust error obtained *during* training and the robust error achieved at the very *end* of training. Moreover, robust overfitting seems to be a general property of all AT methods, as [73] observe it not only for the original formulation by [38] but also in other settings, such as the improved FGS-AT scheme by [69] and even TRADES [72]. After extensive experiments, [73] find that the only effective remedy appears to be early stopping with a validation set. It is therefore important when performing model selection on adversarially trained networks to make use of early stopping on the robust validation loss to mitigate robust overfitting. However, we are only just beginning to understand

[5] https://github.com/MadryLab/mnist_challenge. Accessed 2023-02-03.

[6] https://github.com/MadryLab/cifar10_challenge. Accessed 2023-02-03.

[7] https://www.aicrowd.com/challenges/nips-2018-adversarial-vision-challenge-untargeted-attack-track. Accessed 2023-02-03.

this phenomenon, and some recent works have proposed other techniques to overcome robust overfitting [74], [75].

While some AT methods can be certified, not all certified training methods are variants of AT, because they may not always need to generate actual adversarial examples. To avoid generating adversarial examples, such training schemes usually rely on geometric properties of the networks to compute approximations of regions that are guaranteed to be free of adversarials. Regularization terms are then added to the loss to maximize those regions. This technique is used, for example, by [76], [77] to construct provably robust ReLU networks. Similarly, [78] characterize adversarial-free regions around input samples using polyhedral envelopes and derive a regularization term to maximize these regions. These are all **certified hardening** defenses.

4.2 Gradient masking

In the early days of adversarial machine learning, the FGS attack was very popular because of its efficiency and effectiveness against undefended models. As such, a number of works focused specifically on thwarting the FGS attack and its derivatives such as PGD. For instance, [79] observed that FGS and PGD are only effective because the gradient of the loss is large around natural data samples. It is therefore intuitive to explicitly constrain the magnitude of the gradient around training samples, and to incorporate such a penalty as a regularization term in the loss. Gu et al. [79] achieve this using a *deep contractive network* (DCN), which attempts to directly constrain the matrix norm of the Jacobian of the entire network. They also considered noise injection techniques, *i.e.*, the addition of random noise to the inputs in an effort to defeat adversarial perturbations. This proved to be an extremely popular avenue of research, and many papers tried similar strategies. Well-known examples include pixel deflection [80], random data transformations [81] and stochastic activation pruning [82]. Even JPEG compression has been tried as a defense against adversarial perturbations [83].

The defenses discussed above enjoyed considerable popularity in the early years of AML, but do not really work. If we let ϕ denote some pre-processing function of the data, then the "robust" model takes the form $\hat{f} = f \circ \phi$ where f is the original classifier. Simple preprocessing steps such as JPEG compression were found to be effective against FGS and PGD adversaries. However, these defenses are easily bypassed: if we have access to the original model f without the preprocessing ϕ, then we can perform a **transfer attack** against $\hat{f}$: we generate adversarial examples directly for f and send them to $\hat{f}$. Surprisingly, this strategy will almost always work and the accuracy of $\hat{f}$ gets degraded just as much as if we had not used any defense at all. The problem stems from the fact that many preprocessing operations ϕ are not differentiable, and so gradient-based adversarial attacks cannot evaluate $\nabla\mathcal{L}$ properly for the composition $f \circ \phi$. Yet the transformation ϕ preserves so much structure of the original problem that the shape of $\nabla\mathcal{L}$ (the "loss landscape") barely changes between f and $\hat{f}$. Hence adversarials generated against f will still work against $\hat{f}$.

In a seminal paper, [51] systematically investigated this phenomenon and found that most of the adversarial defenses proposed up to that point suffered from this same problem, which they called *gradient masking*. Gradient masking is illustrated in figure 4.1 using visualizations taken from Nicholas Carlini's ICML 2018 talk about this issue.[8] Figure 4.1a shows the loss landscape of a regular classifier that does not mask gradients. Specifically, what is shown here is a 3D visualization of the loss $\mathcal{L}$ of a model around a point $\boldsymbol{x}$, generated according to

$$L(\alpha_1, \alpha_2) = \mathcal{L}(\boldsymbol{x} + \alpha_1\boldsymbol{v}_1 + \alpha_2\boldsymbol{v}_2).$$

Here, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are orthogonal directions in $\mathbb{R}^n$ and $\alpha_1, \alpha_2 \in \mathbb{R}$ are 2D coordinates that determine how far we move the original point $\boldsymbol{x}$ along these directions. The height of the landscape then corresponds to the value of the loss $L(\alpha_1, \alpha_2)$. The different colors correspond to different predicted labels, so moving from one colored region to another implies a change in the resulting classification. We can see that the loss landscape of a regular classifier is very smooth, and it is plain to see how gradient-based optimization could generate adversarial examples in this case: simply move downward along the slope of the mountain until we cross the boundary between the teal and the green areas. However, when gradient masking occurs, the loss landscape changes to what is shown in figure 4.1b, with figure 4.1c showing a zoomed-in view. At a high level, we see that essentially nothing has changed: we can still follow the slope of the mountain downward to reach different classifications, but we won't be able to do this using first-order gradient optimization methods; the landscape is too discontinuous for that to be possible. Yet it is also clear that this gradient masking classifier $\hat{f}$ is no more robust than the regular one f, since we don't need to cross any larger distance to obtain adversarial examples. In fact, the loss landscape of the gradient masking classifier is so similar to the undefended model that gradient-based methods easily succeed in a transfer attack. The robustness gained from gradient masking is an illusion.

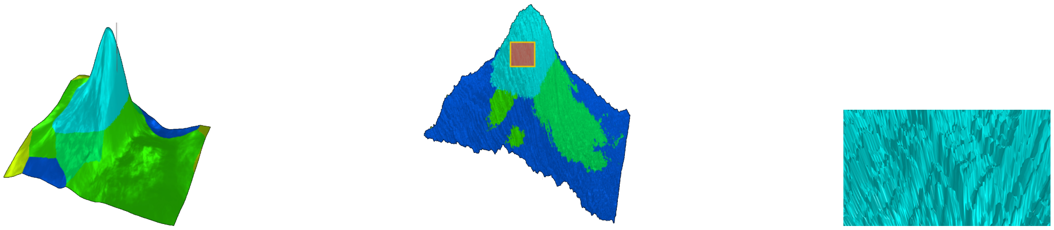
The discovery of gradient masking led to a paradigm shift in how adversarial defenses are evaluated. Specifically, if the defense incorporates any non-differentiable components or randomness, the evaluation must be adapted using specialized techniques such as EOT [58] or backwards-pass differentiable approximation (BPDA) [51].

4.3 Randomized smoothing

Given the relative ease with which researchers were able to break most proposed defenses very quickly after they became known, the community naturally started to focus more on *certified* defenses where one can mathematically prove that no adversarials exist within a certain radius. Perhaps the most well-known such defense is *randomized smoothing* (RS) [84]. RS is both remarkably simple and remarkably effective. To implement it, one takes an existing pre-trained model f and computes the *smoothed model*

$$\hat{f}(\boldsymbol{x}) = \operatorname*{argmax}_{y \in \mathcal{Y}} \Pr_{\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})}[f(\boldsymbol{x} + \boldsymbol{\eta}) = y]. \tag{4.3}$$

[8] https://nicholas.carlini.com/talks/2018_icml_obfuscatedgradients.mp4. Accessed 2023-02-15.



(a) Smooth loss landscape (b) Gradient masking (c) Gradient masking (detail).

Fig. 4.1: Comparison of the loss landscapes of normal and gradient masking classifiers.

By sampling around a given point $\boldsymbol{x}$, the stability of the classifier f under small additive perturbations can be assessed. Despite the fact that RS is randomized, [84] show that one can derive a *deterministic* robustness certificate from it. Specifically, if $\hat{f}$ is defined as in (4.3), then $\hat{f}(\boldsymbol{x}+\boldsymbol{\delta}) = \hat{f}(\boldsymbol{x})$ for all perturbations $\boldsymbol{\delta}$ satisfying

$$\|\boldsymbol{\delta}\| \leq \frac{\sigma}{2}\left(\Phi^{-1}(p_1) - \Phi^{-1}(p_2)\right) \tag{4.4}$$

where Φ is the standard Gaussian cumulative density function (CDF) and p_1, p_2 are probabilities that correspond to a lower and upper bound respectively on the top-2 class probabilities assigned by f to the given sample $\boldsymbol{x}$ under Gaussian perturbations. RS therefore belongs to the class of **certified randomized hardening** defenses, and it is powerful enough to certify the point-wise robustness $\rho(f, \boldsymbol{x})$ at any given point $\boldsymbol{x}$ instead of only providing guarantees on the expected robustness $\rho(f)$.

The robustness bound (4.4) provided by RS is very intuitive to interpret and very simple to optimize: in order to maximize the robustness of the smoothed model $\hat{f}$, one can either increase the variance σ^2 or increase the margin $p_1 - p_2$ between the top two predictions. The former option risks lowering classifier accuracy, as the noise may eventually drown out any meaningful signal. The latter option, however, will be familiar to ML practitioners as maximization of probability margins is known to have all sorts of beneficial effects [85]. It would therefore stand to reason that the combination of RS with other techniques that promote large margins, such as the large-margin softmax loss [86] or temperature scaling [87], would further aid robustness.

The main drawback of randomized smoothing is that the underlying model to which it is applied must be robust to relatively large Gaussian perturbations of the input. This may not be the case unless the model was specifically trained to accommodate such distortions, which is not common practice: Gaussian noise typically used in data augmentations tends to be relatively small. Therefore, although RS can take *any* classifier and turn it into a certifiably robust model, clean and robust accuracy may suffer due to the inability of the model to handle large Gaussian noise.

To compensate for this problem, [88] proposed *denoised smoothing* (DS), where a denoising model is prepended to the classifier before applying RS. Since RS makes no assumptions on the underlying model, it can still make the entire model provably robust despite the addition of a denoiser, and the downstream classifier will achieve better accuracy scores because the denoiser removes the large Gaussian perturbations. Denoised smoothing can be formulated as follows:

$$\hat{f}(\boldsymbol{x}) = \operatorname*{argmax}_{y \in \mathcal{Y}} \Pr_{\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})}[f(D(\boldsymbol{x}+\boldsymbol{\eta})) = y]. \tag{4.5}$$

Note that this procedure is essentially identical to (4.3) except for the addition of the denoising model D. DS therefore belongs to the category of **certified randomized hardening** defenses, as it inherits all the guarantees of RS. DS effectively transforms the hard problem of adversarial purification to the much easier problem of Gaussian denoising.

Taking this idea to its logical conclusion, [89] propose *diffusion denoised smoothing* (DDS), which is essentially just DS where the denoiser is instantiated with a diffusion model [9]. DDS is conceptually very simple, as it merely combines two off-the-shelf models to instantly obtain immensely impressive increases in robust accuracy compared to the previous state of the art. However, it is also very resource-intensive, as it relies on prepending a 552M parameter diffusion model to a 305M parameter classifier. Finding ways to reduce the computational burden of this method while maintaining comparable robustness guarantees would be a very interesting avenue of research.

4.4 Purification methods

Many defense methods have been proposed based on the idea of **purification**, *i.e.*, using a generative model to remove adversarial perturbations. *MagNet* [90], for example, uses a combination of multiple detector networks as well as an auto-encoder to detect and subsequently purify adversarial examples. Its use of multiple detectors instead of just one is motivated by the idea that randomness complicates attacks, and so MagNet will randomly select one of its detector networks to examine each sample. Although an influential work, it was later found to be much less effective than initially thought [91]. Another influential method based on denoising is *Defense-GAN* [92], where a generative adversarial network (GAN) is used to remove the adversarial perturbations. Given a generator G, Defense-GAN purifies an incoming sample $\boldsymbol{x}$ by finding a latent code $\boldsymbol{z}$ such that the reconstruction error $\|G(\boldsymbol{z}) - \boldsymbol{x}\|_2$ is minimized. It then feeds $G(\boldsymbol{z})$ into the downstream model instead of the original input $\boldsymbol{x}$. This method of course relies on the assumption that adversarial examples do not lie on the manifold learned by the generator G. Defense-GAN also requires potentially expensive optimization to find $\boldsymbol{z}$.

Naturally, with the rise of diffusion models came purification defenses that made use of this new powerful class of generative networks. A recent example is *DiffPure* [93], which works by first adding Gaussian noise to the input image via the forward diffusion process and then using the reverse generative process to restore the original. This is similar in spirit to denoised smoothing, but can be computationally much more efficient depending on the architecture of the diffusion model. Similarly, [94] propose *Adaptive Denoising Purification* (ADP), which also first injects Gaussian noise into the images and then uses an energy-based model trained with denoising score matching to purify them. Interestingly, ADP is actually a **certified** defense, and denoising score matching is known to be connected to diffusion models [95]. It may therefore also be possible to certify other diffusion-based defenses such as DiffPure.

4.5 Statistical hypothesis testing

It is natural to use the machinery of statistical hypothesis testing to detect adversarial examples, and this is indeed the basis of many **detection** approaches. Such defenses crucially rely on the idea that adversarial examples come from a distribution that is significantly different from the original data distribution, and that this difference can be detected using efficient statistical tests. Grosse et al. [96] made use of this idea to design an efficient detector method based on the *maximum mean discrepancy* (MMD) test developed by [97]. A problem with such statistical tests is that they can only make decisions about *sets* of samples, not individual samples by themselves. Works such as [98], [99], [100] attempt to address this by using tests which can be pre-trained or calibrated beforehand on collections of data, after which they can be applied sample-wise to individual inputs.

Defenses based on statistical hypothesis testing enjoyed widespread acceptance early on. However, [101] went on to break a number of popular ones, including the MMD-based defense by [96]. As the MMD test is considered one of the strongest statistical tests for detecting distributional shifts, Carlini et al. [101] argued that their break of this defense implies the statistical testing approach to adversarial defense may be fundamentally flawed. Indeed, it is not difficult to imagine why statistical hypothesis testing can fail to provide robustness. For one, it is based on the (controversial) assertion that adversarial examples necessarily come from a different distribution. While this may be true for many adversarial examples generated by popular attacks, it is unclear whether this is an inherent property. Goodfellow [102] has argued against this view, stating that adversarial examples can clearly be found within undersampled regions of the original data distribution. In general, it is more likely that adversarial perturbations simply break the i.i.d. assumption: the idea that incoming data samples form a sequence of independent and identically distributed observations, an assumption that is foundational to almost all ML algorithms. At a higher level, it is difficult to see why statistical tests would offer any real robustness when an adversary is aware of the test being used. In that case, the adversary can simply incorporate the test statistic as an additional regularization term in the objective of (3.1). In order for any particular statistical hypothesis test to make sense as a detection mechanism of adversarial perturbations, a persuasive argument must be made as to why an adversary cannot simply optimize for the test statistic as well and bypass the detector this way. Most off-the-shelf statistical tests cannot support such an argument, because they again assume i.i.d. data samples; they were never designed to be used in a scenario where an adversary is actively trying to fool the test and has intimate knowledge of its inner workings. Statistical tests may therefore work against static adversaries, but it is highly doubtful they are of any use against adaptive ones.

There are exceptions to this, however. One recent example is the work of [103], who combine randomized smoothing with *conformal prediction* [104]. Conformal prediction is a type of frequentist hypothesis testing framework based on a scoring function $S : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Gendler et al. [103] apply randomized smoothing to the scoring function of a conformal predictor in order to construct a certified defense, since the resulting method inherits the provable guarantees of RS. Essentially, they employ a frequentist hypothesis testing framework where RS is applied to the test statistic in order to obtain certified robustness. This strategy is potentially applicable to all manner of hypothesis tests and may therefore lead to new effective statistical tests for detecting adversarial examples. There is also the work by [105], who propose a detection scheme for adversarial examples based on the Mahalanobis distance:

$$M(\boldsymbol{x}) = \max_c -(f(\boldsymbol{x}) - \hat{\boldsymbol{\mu}}_c)^\intercal \boldsymbol{\Sigma}^{-1}(f(\boldsymbol{x}) - \hat{\boldsymbol{\mu}}_c).$$

Here, c ranges over all possible classes, $\hat{\boldsymbol{\mu}}_c$ is the empirical mean of $f(\boldsymbol{x})$ for all samples $\boldsymbol{x}$ belonging to class c and $\boldsymbol{\Sigma}$ is the covariance matrix of the training samples. By thresholding $M(\boldsymbol{x})$ at appropriate levels, one can obtain a remarkably powerful detector that serves as an easy and efficient baseline for adversarial defense and out-of-distribution detection. It is equivalent to fitting class-conditional Gaussian distributions to the output of f with shared covariance across all classes. The quantity $M(\boldsymbol{x})$ then corresponds to the log of the probability density of the test sample.

4.6 Combination therapies

More recent efforts towards improving adversarial robustness have focused on "combination therapies," *i.e.*, the combination of many relatively small changes in model architecture, data curation and training regimes that appear to add up to significantly increased robustness. Gowal et al. [106], for example, showed that one can greatly boost adversarial robustness by combining a number of adjustments, including: a careful choice of loss function with appropriate regularizers, increased model capacity, the use of additional training data (even synthetic samples created by generative models) and modifications to the optimization algorithm such as weight averaging [107]. In a similar vein, [108] show that weight averaging combined with carefully chosen data augmentation strategies can also significantly boost robustness. It has also been noted that Vision Transformers [109] tend to be intrinsically more robust to certain types of adversarial perturbations, such as severe occlusions [110]. However, the dot-product attention mechanism widely used in Transformer architectures suffers from vulnerabilities of its own which compromise robustness [111].

In all, these works seem to suggest that the fragility of our DNNs to adversarial perturbations is likely not caused by any one component of the ML pipeline in isolation. Rather, it is a consequence of the interplay of various elements, from the data curation process to the neural network architecture and specifics of the training algorithm. This can be seen in the results of the works cited above, where considerable improvements in robustness are obtained through the combination of many small tweaks rather than through the construction of an elaborate dedicated defensive component.

4.7 Robustness against multiple threat models

The defenses discussed so far typically focus on one specific threat model, such as L_2 or L_∞. Ideally, of course, we would like our models to be robust against *all* plausible threat models simultaneously. However, Tramèr and Boneh [112] demonstrate that robustness against one threat model in general does not guarantee *any* robustness at all against other threat models. We also cannot simply ensemble models robust to different threat models and expect their robustness properties to add up, nor is it wise to simply mix different perturbation types during adversarial training. Moreover, [112] prove the existence of so-called *mutually exclusive perturbations*, which are pairs of perturbation classes with the property that robustness to one class necessarily implies vulnerability to the other. Under certain assumptions, L_1 and L_∞ perturbations are mutually exclusive, so one cannot in general have robustness against these two threat models simultaneously. Obtaining robustness to multiple threat models remains an active area of research, and specialized techniques have been developed for this purpose [113].

4.8 Evaluation and benchmarking

Before we conclude the section on adversarial defenses, it is important to reflect on current and past practices adopted by the field for the evaluation and benchmarking of proposed methods. This is particularly relevant for adversarial ML, as experience has shown that great care must be taken in order to properly evaluate defense algorithms. Historically, most of the proposed defenses against adversarial examples have failed because of common mistakes in the evaluation methodology. Carlini et al. [63] list a few typical errors:

- Failure to specify a precise threat model.
- Failure to evaluate against an *adaptive* adversary.
- Reporting robust accuracy only for a fixed attack budget.
- Neglecting basic sanity checks. At the very least, one should verify that an *unbounded* attack, *i.e.*, an attack that has no limit on the perturbation budget, reaches 100% success rate. After all, with unbounded perturbations it is possible to transform a given input into any other input and so any target output could be reached. If unbounded attacks fail, it is highly likely that the defense is implemented incorrectly and robustness is over-estimated.
- Failing to tune hyperparameters. Most adversarial attacks and defenses have at least some hyperparameters that should be tuned for the specific task. In any evaluation, failing to tune the hyperparameters of the attacks can easily cause over-estimation of robustness. This is a common and subtle way in which defense evaluations can become overly optimistic.

As discussed by [114], the field has mostly taken these suggestions to heart, and defense evaluations have been getting significantly better in recent years. The introduction of the AutoAttack suite by [39] and the RobustBench leaderboard[9] serves as another important milestone in this regard, as to date this is the most successful attempt at systematically benchmarking existing adversarial defense algorithms. The RobustBench benchmark works by simply applying the AutoAttack suite to a given defense for several different data sets and threat models. For each defense, the leaderboard reports the original publication that proposed it, their standard accuracy, robust accuracy according to the AutoAttack suite, a flag indicating whether the defense uses additional data and a flag indicating whether the evaluation may be unreliable. While definitely a step in the right direction, RobustBench has important shortcomings that researchers need to be aware of. First, RobustBench relies on AutoAttack, which is *not* an adaptive adversary: it is a collection of static adversaries that are deemed to be generally strong against most defenses. It only supports image recognition tasks under L_2 and L_∞ threat models with fixed perturbation budgets on a limited number of data sets (CIFAR-10 and ImageNet). RobustBench does not support detector methods, and considers accuracy as the sole important metric, neglecting computational and memory overhead.

5 Theoretical results

In this section, we survey some of the theoretical results that have been described in the adversarial ML literature. Of particular interest to us here are theorems regarding conditions for the (non-)existence of adversarial examples, bounds on the robustness achievable by a given model and computational hardness results for adversarial defense.

5.1 On the causes of adversarial fragility

Perhaps the most interesting theoretical aspect of adversarial examples, is the reason why they exist at all. Naturally, this question has attracted considerable attention, and various explanations for the existence of adversarial examples have been proposed. One such explanation is offered by [65], who constructed a toy classification task where accurate classifiers provably cannot have *any* robustness. They define the following distribution:

$$\begin{aligned} Y &\sim \mathrm{Unif}(\{-1,+1\}), \\ X_1 &= \begin{cases} +Y & \text{w.p. } p, \\ -Y & \text{w.p.} 1-p, \end{cases} \\ X_2,\dots,X_{n+1} &\sim \mathcal{N}(\eta Y, 1). \end{aligned} \tag{5.1}$$

That is, we have a uniformly distributed binary class label $Y \in \{-1,+1\}$ and observed $(n+1)$-dimensional feature vectors $\boldsymbol{X}$. The first feature X_1 can be equal or opposite to Y depending on some probability p, and the other features $X_2,\dots,X_{n+1}$ are weakly correlated with Y. Tsipras

[9] https://robustbench.github.io. Accessed 2023-02-03.

et al. [65] then show that a classifier that simply returns the sign of the average value of the features $X_2, \ldots, X_{n+1}$ can achieve near-perfect accuracy for large n. However, this classifier is not robust to L_∞ adversarial perturbations for even very small budgets ε: in particular, high standard accuracy implies *zero* robust accuracy for $\varepsilon \geq 2\eta$, where η can be arbitrarily small.

Of course, as damning as the above result may seem, it applies only to (5.1). This is a toy problem that creates an extremely pessimistic scenario for a robust classifier, as there is only one "good" feature and all the other features are essentially rubbish. Nevertheless, it does provide an important insight into why adversarial examples exist in general: it is possible even in realistic classification tasks that our models are over-optimizing for accuracy and depending on large groups of features that individually are only very weakly correlated with the label. Due to these weak correlations, such features can be perturbed significantly without affecting the overall task, but this will fool a classifier that is heavily dependent on them. In this view, adversarial examples merely exploit an over-reliance of classifiers on weak features.

This idea was further elaborated on in a widely celebrated paper by [115]. They proposed the "robust features model," where features are classified along two axes: usefulness and robustness. Here, a "feature" is any $\mathcal{X} \to \mathbb{R}$ function, and classifiers generate predictions by taking the sign of weighted sums of features. A feature f is then called ρ-useful if $\mathbb{E}[Y \cdot f(\boldsymbol{X})] \geq \rho$. It is called γ-robust if $\mathbb{E}\left[\inf_{\boldsymbol{\delta}} Y \cdot f(\boldsymbol{X} + \boldsymbol{\delta})\right] \geq \gamma$. A useful feature correlates well with the label Y under the standard data distribution. A robust feature maintains this correlation even in the presence of adversarial perturbations. Ilyas et al. [115] then argue that classifiers are vulnerable to adversarial perturbations because they rely too much on features that are useful but not robust, and they propose an interesting experiment to verify this. Specifically, they take existing non-robust models f and use them to generate a special data set $\tilde{S}$ which contains adversarial examples for f and the associated *incorrect* predictions given by f on these adversarials. Then, this mislabeled data set $\tilde{S}$ is used to train new models $\tilde{f}$. Paradoxically, these models $\tilde{f}$, which were trained on mislabeled data, generalize to the original test data just as well as the properly trained models f. They repeat this experiment when the original models f have been adversarially trained, and find that this leads to new models $\tilde{f}$ that exhibit higher robustness to adversarial attacks than the baseline non-robust ones.

The idea proposed by [115] is currently one of the most popular explanations for the existence of adversarial examples, but it is by no means the only one. Tanay and Griffin [116] proposed a different hypothesis called *boundary tilting*, which attributes the existence of adversarial examples to a geometric misalignment between the true data manifold and the learned classification boundaries. Other explanations utilizing the geometry of the data manifold were proposed, for example, by [117], [118], who relate adversarial vulnerability to curvature properties of the decision boundaries.

At present, there are multiple distinct hypotheses described in the literature which can all explain the existence of adversarial examples. However, most of these hypotheses do not seem to imply any of the others. We thus have multiple independent but plausible explanations for why adversarial examples exist, suggesting that this phenomenon may not be reducible to a single clearly-defined cause. It could therefore be the case that qualitatively different types of adversarial examples exist which may require distinct approaches for defense. Constructing a relevant taxonomy of adversarial examples according to their underlying cause would be highly informative.

5.2 Robustness certification

In order to assess the robustness of a given DNN in practice, the most straightforward method is to simply attack the network with a strong algorithm and compute its empirical robustness on the generated samples. However, this approach may not be efficient if the attack is slow, since a large number of samples may be required to obtain statistically reliable estimates of robust accuracy. Moreover, in general, there is no guarantee that high robust accuracy against a given attack implies any robustness at all against other attacks. For this reason, there has also been much interest in developing methods that can efficiently and reliably certify robustness of any given model without the need for extensive experiments with suites of existing attacks. The randomized smoothing defense by [84] which we described earlier essentially obtains certification abilities as a side-effect, but other methods exist that focus solely on certification and do not provide any defense themselves. Li et al. [119] is perhaps most similar in spirit to randomized smoothing, as they also rely crucially on additive Gaussian noise to certify robustness. Their method also allows for the derivation of a training algorithm that improves robustness.

The main disadvantage of dedicated certification techniques is that they tend to be highly specific to certain architectures, as the problem of robustness certification for general neural networks is known to be NP-hard even under very simple threat models [120, theorem 3.1]. For example, for tree-based models such as decision trees, random forests and gradient-boosted trees, [121] proposed an iterative algorithm that gives tight lower bounds. The CLEVER score [122] has appeared in many publications as a robustness estimate for general neural networks. It is based on estimation of the local Lipschitz constant, and its results seem to align well with empirical robustness tests. In a similar vein, [123] develop the CROWN score, which has also been widely used. Other works that derive robustness bounds based on efficient estimates of the (local) Lipschitz constants include [124], [125].

Perhaps the most well-studied scenario is the ReLU network, *i.e.*, when the neural network is restricted to using only the ReLU or linear activation functions between layers. Weng et al. [120] introduce Fast-Lin and Fast-Lip to efficiently certify robustness of such networks. Fast-Lin relies on linear approximations of the individual layers whereas Fast-Lip uses Lipschitz estimation techniques similar to [122]. In some cases, researchers cast the robustness certification problem for ReLU networks as a semi-definite program, which is a rich area in the field of optimization theory [126]. Raghunathan et al. [127] provide one such example, and the work of [77] is similar.

5.3 Hardness results

Much work has also gone into the study of the *hardness* of robust learning. Specifically, determining the conditions under which robust learning is possible, and to what extent, is one of the major problems in the field. In this regard, [128] have made several important contributions. They showed that robust learning is impossible in the distribution-free setting even when the adversary is restricted to perturbing just a single *bit* of the input. Therefore, in order to have any hope of creating robust classifiers, certain distributional assumptions *must* be made; no adversarial defense can claim general robustness on arbitrary distributions. However, there are certain concrete distributions that can be robustly learned under milder conditions: the class of monotone conjunctions is robustly learnable if the adversary is limited to perturbing $\mathcal{O}(\log n)$ input bits.

In a similar vein, [129] prove that the Rademacher complexity of robust learning will unavoidably be larger than its natural counterpart under realistic conditions. The Rademacher complexity is a fundamental quantity in statistical learning theory, similar to the VC dimension [130], that measures the complexity of infinite hypothesis classes in a way that can be directly related to the worst-case performance of any finite-sample learner for this class. A higher complexity of the hypothesis class therefore immediately implies worse lower bounds on the error of any learning algorithm. Similarly, [131] prove that hypothesis classes with finite VC dimension can be robustly learned, but only *improperly*. In the case of neural networks, this can imply that additional model capacity is necessary to obtain robustness. Diochnos et al. [132] study the problem of robust learning in the probably approximately correct (PAC) framework [133]. They demonstrate conditions under which robust PAC learning requires either *exponentially* more samples or *polynomially* more samples than standard PAC learning, and give examples where robust PAC learning is impossible altogether.

6 Avenues for future research

In this section, we list some of the research directions we believe are currently most important for the field of AML as a whole.

6.1 Moving beyond the toy problem

As it stands, researchers are much too heavily focused on computer vision and the L_p threat model. Although one of course has to start somewhere, at the time of this writing it has been almost ten years since the work of [21] introduced the deep learning community to this phenomenon, yet we are still studying essentially the same problem despite its known shortcomings. The L_p threat model is a particularly poor fit for computer vision: large L_p perturbations do not necessarily translate to heavily altered images, nor do small L_p perturbations imply small distortion. A small rotation, for instance, could imply a very large L_p perturbation but would not actually change much of what is seen in the image. Furthermore, adversarial robustness is a problem for virtually *all* machine learning models, not just deep neural networks for image recognition. In the case of natural language processing, for example, it is unclear what exactly constitutes a "small" or "imperceptible" perturbation. Most work in this area considers the introduction of typographical errors, where one or more letters of a word are replaced, thus causing slight misspellings in the text. In essence, this is the equivalent of the L_p threat model using the Levenshtein distance as the metric instead of an L_p norm. However, one could also justify replacing words with synonyms or homophones, which could drastically increase the Levenshtein distance but would nevertheless only make a very small difference to human readers.

We believe progress on this front can be made by taking inspiration from differential geometry [134], based on the so-called *manifold hypothesis*. This hypothesis posits that the input data of our ML algorithms, although they are typically given as vectors in $\mathbb{R}^n$ for some potentially large n, in fact almost always lie on an embedded submanifold of much lower dimensionality $d \ll n$. The manifold hypothesis tends to be an unavoidable requirement for most ML algorithms, particularly those dealing with dimensionality reductions [135]. In the simplest case, when the manifold to be learned is smooth, it may be characterized by a single invertible chart $\psi : \mathbb{R}^n \to \mathbb{R}^d$. This chart can be learned using variational auto-encoders [136], normalizing flows [137] or other manifold learning algorithms. We can then use it to rewrite (4.1) as follows:

$$\theta^\star = \min_\theta \ \mathbb{E}\left[\max_{\boldsymbol{\delta}\in B_\varepsilon(\psi(\boldsymbol{X}))} \mathcal{L}(\psi^{-1}(\psi(\boldsymbol{X})+\boldsymbol{\delta}), Y, \theta)\right]. \quad (6.1)$$

In this case, $B_\varepsilon(\boldsymbol{z})$ is the regular L_p norm ball around $\boldsymbol{z} \in \mathbb{R}^d$. In other words, we are constructing adversarial examples by additively perturbing the latent code $\boldsymbol{z} = \psi(\boldsymbol{x})$ of the given input $\boldsymbol{x}$ with vectors $\boldsymbol{\delta}$ of bounded L_p norm and then mapping the resulting perturbed code $\tilde{\boldsymbol{z}} = \boldsymbol{z} + \boldsymbol{\delta}$ back to the input space, obtaining $\tilde{\boldsymbol{x}} = \psi^{-1}(\psi(\boldsymbol{x}) + \boldsymbol{\delta})$. The works of [138], [139] essentially follow the spirit of these ideas, and we expect future work will likely explore this avenue much further. The concept of *semantic adversarial examples* introduced by [31] follows the same spirit, but the concrete implementation would need to be generalized to accommodate more than just images.

6.2 Practical solutions

The question of how to make AML more relevant and accessible to a broader audience is an underestimated but very important open problem in this field. There is a tendency in AML to focus on robust accuracy to the exclusion of other relevant metrics such as efficiency and user-friendliness. This significantly hinders the adoption of robust learning techniques in domains outside of AML proper. If we wish to make all DNNs robust (as well we should), then more research needs to be dedicated to lightweight solutions which can be easily applied in practice with good results without the need for excessive tuning of hyperparameters, mountains of additional data and huge models. This is of course very difficult, and indeed we have discussed some theoretical results which suggest that robust learning may not be possible without significantly more resources. Nevertheless, we should not lose sight of the fact that AML is fundamentally motivated by *trustworthiness*, and so our defenses ought to be constrained not just by the abstract threat model we happen to be interested in but also the requirements and limitations of practitioners who need reliable models to develop viable products. We believe it could greatly benefit the AML community if we started from *real applications* rather than the typical toy problem (3.1). This could not only help us develop more practically useful defenses but also, in doing so, raise much broader awareness of this important issue.

6.3 Improved benchmarks

To facilitate progress towards lightweight solutions and more general threat models, appropriate benchmarks also need to be developed. Currently, the gold standard for robustness benchmarking is the RobustBench library. However, RobustBench is heavily specialized to image classification under the L_p threat model (although the CIFAR-10-C and ImageNet-C data sets have been incorporated into it [140]). It also has little regard for any metric aside from accuracy, limiting the usefulness of the library in resource-constrained environments or applications where large data sets are hard to come by (*e.g.*, medicine). Ideally, our current benchmarks should be extended to support a greater variety of performance metrics and threat models, including patch attacks and other more natural data transformations, as well as different machine learning tasks such as NLP. Even when restricted to the domain of image classification, the field of AML tends to only benchmark robustness on CIFAR-10 and ImageNet, neglecting many other data sets where robustness is clearly desirable. Medical imaging data sets immediately come to mind in this regard. In the case of NLP, the AML community can take inspiration from "red teaming," a popular practice where a language model is probed for harmful or nonsensical outputs [141].

7 Conclusions

Unless proper precautions are taken, DNNs are extremely vulnerable to adversarial perturbations. This problem casts serious doubt on the trustworthiness of our AI systems and can present a major obstacle to the adoption of machine learning in practice. After all, when DNNs are used to assist with medical diagnoses, approving loan applications or screening resumés, tangible harm can result if the predictions drastically change on the basis of imperceptible perturbations. This poses a challenge for high-risk applications of AI to comply with new legislative frameworks such as the European AI Act.[10]

Despite over a decade of sustained research efforts, our current solutions are still unsatisfactory in a number of important ways: they tend to impose a significantly higher computational burden, require much additional data, are complicated and difficult to deploy properly or they simply do not reach the desired level of accuracy for the task at hand. The practical adoption of AML techniques is also hindered by the field's narrow focus on image classification under a limited number of threat models and data sets, as well as its reliance on robust accuracy as the sole important metric.

The L_p threat model is a useful starting point for robustness analysis, but it hits clear limitations when applied to discrete domains such as natural language. For instance, at this time the field of AML has not been able to substantially improve the adversarial robustness of generative large language models [142]. In order to meet the acute and growing demand of society for robust and trustworthy AI systems, the field of AML urgently needs to diversify its benchmarks and motivate its threat models using concrete and realistic use-cases.

[10] https://artificialintelligenceact.eu/. Accessed 2023-03-06.

Acknowledgments

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. [Online]. Available: probml.ai

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[5] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, "CoCa: Contrastive captioners are image-text foundation models," *arXiv preprint arXiv:2205.01917*, 2022.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] J. Fan and C. Xiao, "Generalized data distribution iteration," *arXiv preprint arXiv:2206.03192*, 2022.

[8] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[9] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[10] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[11] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, and P. Xie, "COVID-CT-dataset: a CT scan dataset about COVID-19," *arXiv preprint arXiv:2003.13865*, 2020.

[12] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[13] A. Wong, M. J. Shafiee, and M. S. Jules, "MicronNet: a highly compact deep convolutional neural network architecture for real-time embedded traffic sign classification," *IEEE Access*, vol. 6, pp. 59 803–59 810, 2018.

[14] E. Ong, M. U. Wong, A. Huffman, and Y. He, "COVID-19 coronavirus vaccine design using reverse vaccinology and machine learning," *Frontiers in immunology*, vol. 11, p. 1581, 2020.

[15] M. Al-Emran, M. N. Al-Kabi, and G. Marques, "A survey of using machine learning algorithms during the COVID-19 pandemic," *Emerging technologies during the era of COVID-19 pandemic*, pp. 1–8, 2021.

[16] I. Rahimi, F. Chen, and A. H. Gandomi, "A review on COVID-19 forecasting models," *Neural Computing and Applications*, pp. 1–11, 2021.

[17] Y. Vorobeychik and M. Kantarcioglu, "Adversarial machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–169, 2018.

[18] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[19] M. Sato, J. Suzuki, H. Shindo, and Y. Matsumoto, "Interpretable adversarial perturbation in input embedding space for text," *arXiv preprint arXiv:1805.02917*, 2018.

[20] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE security and privacy workshops (SPW)*. IEEE, 2018, pp. 1–7.

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[22] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 641–647.

[23] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[24] J. Wang, C. Wang, Q. Lin, C. Luo, C. Wu, and J. Li, "Adversarial attacks and defenses in deep learning for image recognition: A survey," *Neurocomputing*, 2022.

[25] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," *IEEE Access*, vol. 9, pp. 155 161–155 196, 2021.

[26] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.

[27] M. Strobel and R. Shokri, "Data privacy and trustworthy machine learning," *IEEE Security & Privacy*, vol. 20, no. 5, pp. 44–49, 2022.

[28] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[29] R. Chen, J. Li, J. Yan, P. Li, and B. Sheng, "Input-specific robustness certification for randomized smoothing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 6295–6303.

[30] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[31] H. Hosseini and R. Poovendran, "Semantic adversarial examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1614–1619.

[32] A. Levine and S. Feizi, "(De)Randomized smoothing for certifiable defense against patch attacks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6465–6475, 2020.

[33] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2196–2205.

[34] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[35] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[36] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[37] F. Croce and M. Hein, "Mind the box: l_1-APGD for sparse adversarial attacks on image classifiers," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2201–2211.

[38] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[39] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2206–2216.

[40] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[41] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*. IEEE, 2017, pp. 39–57.

[42] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[43] U. Ozbulak, A. Van Messem, and W. De Neve, "Not all adversarial examples require a complex defense: Identifying overoptimized adversarial examples with IQR-based logit thresholding," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[44] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, 2019.

[45] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, "SparseFool: a few pixels make a big difference," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9087–9096.

[46] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[47] K. Xu, S. Liu, P. Zhao, P.-Y. Chen, H. Zhang, Q. Fan, D. Erdogmus, Y. Wang, and X. Lin, "Structured adversarial attack: Towards general implementation and better interpretability," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=BkgzniCqY7

[48] Y. Jiang, C. Liu, Z. Huang, M. Salzmann, and S. Süsstrunk, "Towards stable and efficient adversarial training against *l*_1 bounded adversarial attacks," in *40th International Conference on Machine Learning (ICML 2023)*, 2023.

[49] F. Croce, J. Rauber, and M. Hein, "Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks," *International Journal of Computer Vision*, vol. 128, no. 4, pp. 1028–1046, 2020.

[50] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," *arXiv preprint arXiv:1611.01491*, 2016.

[51] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

[52] G. Carbone, M. Wicker, L. Laurenti, A. Patane, L. Bortolussi, and G. Sanguinetti, "Robustness of Bayesian neural networks to gradient-based attacks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 602–15 613, 2020.

[53] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2484–2493.

[54] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[55] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: a query-efficient black-box adversarial attack via random search," in *European Conference on Computer Vision*. Springer, 2020, pp. 484–501.

[56] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.

[57] A. Rahmati, S.-M. Moosavi-Dezfooli, P. Frossard, and H. Dai, "GeoDA: a geometric framework for black-box adversarial attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8446–8455.

[58] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 284–293.

[59] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[60] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1765–1773.

[61] A. Chaubey, N. Agrawal, K. Barnwal, K. K. Guliani, and P. Mehta, "Universal adversarial perturbations: A survey," *arXiv preprint arXiv:2005.08087*, 2020.

[62] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial example defenses: Ensembles of weak defenses are not strong," in

Proceedings of the 11th USENIX Conference on Offensive Technologies, ser. WOOT'17. USENIX Association, 2017, p. 15.

[63] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019.

[64] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[65] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *International Conference on Learning Representations*, 2019.

[66] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[67] A. Ben-Tal and A. Nemirovski, "Robust optimization–methodology and applications," *Mathematical programming*, vol. 92, no. 3, pp. 453–480, 2002.

[68] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," *arXiv preprint arXiv:1703.09387*, 2017.

[69] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," *arXiv preprint arXiv:2001.03994*, 2020.

[70] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *arXiv preprint arXiv:1904.12843*, 2019.

[71] A. Sinha, H. Namkoong, R. Volpi, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," *arXiv preprint arXiv:1710.10571*, 2017.

[72] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7472–7482.

[73] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8093–8104.

[74] L. Li and M. Spratling, "Understanding and combating robust overfitting via input loss landscape analysis and regularization," *Pattern Recognition*, vol. 136, p. 109229, 2023.

[75] H. Liu, Z. Zhong, N. Sebe, and S. Satoh, "Mitigating robust overfitting via self-residual-calibration regularization," *Artificial Intelligence*, p. 103877, 2023.

[76] F. Croce, M. Andriushchenko, and M. Hein, "Provable robustness of ReLU networks via maximization of linear regions," in *the 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 2057–2066.

[77] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5286–5295.

[78] C. Liu, M. Salzmann, and S. Süsstrunk, "Training provably robust models by polyhedral envelope regularization," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[79] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," *arXiv preprint arXiv:1412.5068*, 2014.

[80] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8571–8580.

[81] A. N. Bhagoji, D. Cullina, C. Sitawarin, and P. Mittal, "Enhancing robustness of machine learning systems via data transformations," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2018, pp. 1–5.

[82] G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=H1uR4GZRZ

[83] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[84] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1310–1320.

[85] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[86] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks." in *ICML*, vol. 2, 2016, p. 7.

[87] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[88] H. Salman, M. Sun, G. Yang, A. Kapoor, and J. Z. Kolter, "Denoised smoothing: A provable defense for pretrained classifiers," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 21 945–21 957. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/f9fd2624beefbc7808e4e405d73f57ab-Paper.pdf

[89] N. Carlini, F. Tramer, J. Z. Kolter *et al.*, "(Certified!!) adversarial robustness for free!" *arXiv preprint arXiv:2206.10550*, 2022.

[90] D. Meng and H. Chen, "MagNet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 135–147.

[91] N. Carlini and D. Wagner, "MagNet and "Efficient Defenses Against Adversarial Attacks" are not robust to adversarial examples," *arXiv preprint arXiv:1711.08478*, 2017.

[92] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," *arXiv preprint arXiv:1805.06605*, 2018.

[93] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," in *International Conference on Machine Learning (ICML)*, 2022.

[94] J. Yoon, S. J. Hwang, and J. Lee, "Adversarial purification with score-based generative models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 062–12 072.

[95] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arxiv:2006.11239*, 2020.

[96] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[97] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[98] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[99] J. Peck, B. Goossens, and Y. Saeys, "Calibrated multi-probabilistic prediction as a defense against adversarial attacks," in *Artificial Intelligence and Machine Learning*. Springer, 2019, pp. 85–125.

[100] ——, "Detecting adversarial manipulation using inductive Venn-ABERS predictors," *Neurocomputing*, vol. 416, pp. 202–217, 2020.

[101] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 3–14.

[102] I. Goodfellow, "Defense against the dark arts: An overview of adversarial example security research and future research directions," *arXiv preprint arXiv:1806.04169*, 2018.

[103] A. Gendler, T.-W. Weng, L. Daniel, and Y. Romano, "Adversarially robust conformal prediction," in *International Conference on Learning Representations*, 2022.

[104] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *arXiv preprint arXiv:2107.07511*, 2021.

[105] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[106] S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli, "Uncovering the limits of adversarial training against norm-bounded adversarial examples," *arXiv preprint arXiv:2010.03593*, 2020.

[107] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[108] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. Mann, "Fixing data augmentation to improve adversarial robustness," *arXiv preprint arXiv:2103.01946*, 2021.

[109] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly

et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[110] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. Shahbaz Khan, and M.-H. Yang, "Intriguing properties of vision transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 296–23 308, 2021.

[111] G. Lovisotto, N. Finnie, M. Munoz, C. K. Mummadi, and J. H. Metzen, "Give me your attention: Dot-product attention considered harmful for adversarial patch robustness," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 234–15 243.

[112] F. Tramer and D. Boneh, "Adversarial training and robustness for multiple perturbations," *Advances in neural information processing systems*, vol. 32, 2019.

[113] P. Maini, E. Wong, and Z. Kolter, "Adversarial robustness against the union of multiple perturbation models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6640–6650.

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

[115] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems*, 2019, pp. 125–136.

[116] T. Tanay and L. Griffin, "A boundary tilting persepective on the phenomenon of adversarial examples," *arXiv preprint arXiv:1608.07690*, 2016.

[117] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard, "Robustness via curvature regularization, and vice versa," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[118] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "The robustness of deep networks: A geometrical perspective," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 50–62, 2017.

[119] B. Li, C. Chen, W. Wang, and L. Carin, "Certified adversarial robustness with additive noise," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[120] L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, "Towards fast computation of certified robustness for RELU networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5276–5285.

[121] H. Chen, H. Zhang, S. Si, Y. Li, D. Boning, and C.-J. Hsieh, "Robustness verification of tree-based models," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[122] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," *arXiv preprint arXiv:1801.10578*, 2018.

[123] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[124] J. Peck, J. Roels, B. Goossens, and Y. Saeys, "Lower bounds on the robustness to adversarial perturbations," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/298f95e1bf9136124592c8d4825a06fc-Paper.pdf

[125] M. Hein and M. Andriushchenko, "Formal guarantees on the robustness of a classifier against adversarial manipulation," *arXiv preprint arXiv:1705.08475*, 2017.

[126] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[127] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," *arXiv preprint arXiv:1801.09344*, 2018.

[128] P. Gourdeau, V. Kanade, M. Kwiatkowska, and J. Worrell, "On the hardness of robust classification," in *Advances in Neural Information Processing Systems*, 2019, pp. 7444–7453.

[129] D. Yin, R. Kannan, and P. Bartlett, "Rademacher complexity for adversarially robust generalization," in *International conference on machine learning*. PMLR, 2019, pp. 7085–7094.

[130] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[131] O. Montasser, S. Hanneke, and N. Srebro, "VC classes are adversarially robustly learnable, but only improperly," in *Conference on Learning Theory*. PMLR, 2019, pp. 2512–2530.

[132] D. I. Diochnos, S. Mahloujifar, and M. Mahmoody, "Lower bounds for adversarially robust PAC learning," *arXiv preprint arXiv:1906.05815*, 2019.

[133] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[134] L. W. Tu, *An introduction to manifolds*, 2nd ed., ser. Universitext. New York, NY: Springer, Oct. 2010.

[135] Y. Ma and Y. Fu, *Manifold learning theory and applications*. CRC press Boca Raton, 2012, vol. 434.

[136] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[137] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[138] S. Zhang, K. Huang, J. Zhu, and Y. Liu, "Manifold adversarial learning," *arXiv preprint arXiv:1807.05832*, 2018.

[139] S. Zhang, K. Huang, R. Zhang, and A. Hussain, "Generalized adversarial training in Riemannian space," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 826–835.

[140] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *arXiv preprint arXiv:1903.12261*, 2019.

[141] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse *et al.*, "Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned," *arXiv preprint arXiv:2209.07858*, 2022.

[142] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

Jonathan Peck received the B.Sc. degree in Computer Science at Ghent University in 2015. He obtained the M.Sc. and Ph.D. degrees in Computer Science at Ghent University in 2017 and 2023, respectively. He is currently a post-doctoral researcher at the Department of Applied Mathematics, Computer Science and Statistics as well as the VIB Inflammation Research Center in Ghent, Belgium. His research focuses on improving the robustness of deep learning models to adversarial perturbations.

Bart Goossens is a professor in the Image Processing and Interpretation group of Ghent University, where he currently supervises research on image/video processing, computer vision, AI and heterogeneous platforms mapping tools. He is also a core principal investigator at imec. He earned his master's degree in Computer Science Engineering from Ghent University, Belgium in 2006 and the Ph.D. degree from the same university in 2010. For his research, he received the Barco/FWO prize in 2006 and the annual Scientific Prize IBM Belgium for Informatics in 2011. He is author of more than 100 scientific articles and currently serves as an associate editor for the IEEE Transactions on Image Processing.

Yvan Saeys is a professor in machine learning at Ghent university and a principal investigator at VIB. He obtained a master and PhD degree in Computer Science from Ghent University, respectively in 2000 and 2004. He is heading the Data Mining and Modeling for Biomedicine research group, an interdisciplinary research team that is developing state-of-the-art machine learning methods for biological and medical applications, and is an expert in computational models to analyze high-throughput single-cell data. He has co-authored more than 250 scientific articles and received the FWO/AstraZeneca prize "Patient care in the AI era" for his research.