Holistic Adversarial Robustness of Deep Learning Models

Pin-Yu Chen[1], **Sijia Liu**[2]

[1] IBM Research
[2] Michigan State University
pin-yu.chen@ibm.com, liusiji5@msu.edu

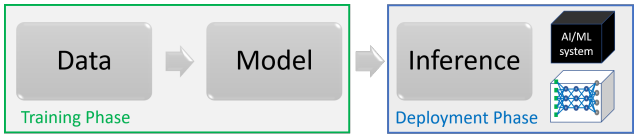
Abstract

Adversarial robustness studies the worst-case performance of a machine learning model to ensure safety and reliability. With the proliferation of deep-learning-based technology, the potential risks associated with model development and deployment can be amplified and become dreadful vulnerabilities. This paper provides a comprehensive overview of research topics and foundational principles of research methods for adversarial robustness of deep learning models, including attacks, defenses, verification, and novel applications.

1 Introduction

Deep learning (LeCun, Bengio, and Hinton 2015) is a core engine that drives recent advances in artificial intelligence (AI) and machine learning (ML), and it has broad impacts on our society and technology. However, there is a growing gap between AI technology's creation and its deployment in the wild. One critical example is the lack of *robustness*, including natural robustness to data distribution shifts, ability in generalization and adaptation to new tasks, and worst-case robustness when facing an adversary (also known as *adversarial robustness*). According to a recent Gartner report[1], 30% of cyberattacks by 2022 will involve data poisoning, model theft or adversarial examples. However, the industry seems underprepared. In a survey of 28 organizations spanning small as well as large organizations, 25 organizations did not know how to secure their AI/ML systems (Kumar et al. 2020). Moreover, various practical vulnerabilities and incidences incurred by AI-empowered systems have been reported in real life, such as Adversarial ML Threat Matrix[2] and AI Incident Database[3].

To prepare deep-learning enabled AI systems for the real world and to familiarize researchers with the error-prone risks hidden in the lifecycle of AI model development and deployment – spanning from data collection and processing, model selection and training, to model deployment and system integration – this paper aims to provide a holistic overview of adversarial robustness for deep learning models. The research themes include (i) attack (risk identification and demonstration), (ii) defense (threat detection and mitigation), (iii) verification (robustness certificate), and (iv) novel applications. In each theme, the fundamental concepts and key research principles will be presented in a unified and organized manner. This paper takes an overarching and holistic approach to introduce adversarial robustness of deep learning models based on the terminology of an AI lifecycle in development and deployment, which differs from existing survey papers that provide an in-depth discussion on a specific threat model. The main goal of this paper is to deliver a primer that provides basic concepts, systematic knowledge, and categorization of this rapidly evolving research field to the general audience and the broad AI/ML research community.



Attack Category / Attacker's Capability	Data	Model / Training Method	Inference
Poisoning Attack	X	X*	
Backdoor Attack	X		
Evasion Attack (adversarial example)		X*	X
Extraction/Inference Attack (model stealing, membership inference, data leakage)			X
Model Injection (model replacement, integrity)		X*	X

*No access to model internal information in the black-box attack setting

Figure 1: Holistic view of adversarial attack categories and capabilities (threat models) in the training and deployment phases. The three types of attacks highlighted in colors (poisoning/backdoor/evasion attack) are the major focus of this paper. In the deployment phase, the target (victim) can be an access-limited black-box system (e.g. a prediction API) or a transparent white-box model.

Figure 1 shows the lifecycle of AI development and deployment and different adversarial threats corresponding to attackers' capabilities (also known as threat models). The lifecycle is further divided into two phases. The *training* phase includes data collection and pre-processing, as well as model selection (e.g. architecture search and design), hyper-parameter tuning, model parameter optimization, and validation. After model training, the model is "frozen" (fixed

[1]https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020

[2]https://github.com/mitre/advmlthreatmatrix

[3]https://incidentdatabase.ai/

Symbol	Meaning
$f_\theta : \mathbb{R}^d \mapsto [0,1]^K$	K-way neural network classification model parameterized by θ
$\mathsf{logit} : \mathbb{R}^d \mapsto \mathbb{R}^K$	logit (pre-softmax) representation
(x, y)	data sample x and its associated groundtruth label y
$\hat{y}_\theta(x) \in [K]$	top-1 label prediction of x by f_θ
x_{adv}	adversarial example of x
δ	adversarial perturbation to x for evasion attack
Δ	universal trigger pattern for backdoor attack
$t \in [K]$	target label for targeted attack
$\mathsf{loss}(f_\theta(x), y)$	classification loss (e.g. cross entropy)
g	attcker's loss function
D_{train} / D_{test}	original training / testing dataset
$\mathcal{T}$	data transformation function

Table 1: Mathematical notation.

Attack	Objective
Poisoning	Design a poisoned dataset D_{poison} such that models trained on D_{poison} fail to generalize on D_{test} (i.e. $\hat{y}_\theta(x_{\text{test}}) \neq y_{\text{test}}$)
Backdoor	Embed a trigger Δ with a target label t to D_{train} such that $\hat{y}_\theta(x_{\text{test}}) = y_{\text{test}}$ but $\hat{y}_\theta(x_{\text{test}} + \Delta) = t$
Evasion (untargeted)	Given f_θ, find x_{adv} such that x_{adv} is similar to x but $\hat{y}_\theta(x_{\text{adv}}) \neq y$
Evasion (targeted)	Given f_θ, find x_{adv} such that x_{adv} is similar to x but $\hat{y}_\theta(x_{\text{adv}}) = t$

Table 2: Objectives of adversarial attacks.

model architecture and parameters) and is ready for deployment. Before deployment, there are possibly some post-hoc model adjustment steps such as model compression and quantification for memory/energy reduction, calibration or risk mitigation. The frozen model providing inference/prediction can be deployed in a white-box or black-box manner. The former means the model details are transparent to a user (e.g. releasing the model architecture and pre-trained weights for neural networks), while the latter means a user can access model predictions but does not know what the model is (i.e., an access-limited model), such as a prediction API. The gray-box setting is a mediocre scenario that assumes a user knows partial information about the deployed model. In some cases, a user may have knowledge of the training data and the deployed model is black-box, such as in the case of an AI automation service that only returns a model prediction portal based on user-provided training data. We also note that these two phases can be recurrent: a deployed model can re-enter the training phase with continuous model/data updates.

Throughout this paper, we focus on adversarial robustness of neural networks for classification tasks. Many principles in classification can be naturally extended to other machine learning tasks, which will be discussed in Section 4. Based on Figure 1, this paper will focus on training-phase and deployment-phase attacks driven by the limitation of current ML techniques. While other adversarial threats concerning model/data privacy and integrity are also crucial, such as model stealing, membership inference, data leakage, and model injection, which will not be covered in this paper. We also note that adversarial robustness of non-deep-learning models such as support vector machines has been investigated. We refer the readers to (Biggio and Roli 2018) for the research evolution in adversarial machine learning.

Table 1 summarizes the main mathematical notations. We use $[K] = \{1, 2, \ldots, K\}$ to denote the set of K class labels. Without loss of generality, we assume the data inputs are vectorized (flattened) as d-dimensional real-valued vectors, and the output (class confidence) of the K-way neural network classifier f_θ is nonnegative and sum to 1 (e.g. softmax as the final layer), that is, $\sum_{k=1}^{K}[f_\theta(\cdot)]_k = 1$. The adversarial robustness of real-valued continuous data modalities such as image, audio, time series, and tabular data can be studied based on a unified methodology. For discrete data modalities such as texts and graphs, one can leverage their real-valued embeddings (Lei et al. 2019), latent representations, or continuous relaxation of the problem formulation (e.g. topology attack in terms of edge addition/deletion in graph neural networks (Xu et al. 2019a)). Unless specified, in what follows we will not further distinguish data modalities.

2 Attacks

This section will cover mainstream adversarial threats that aim to manipulate the prediction and decision-making of an AI model through training-phase or deployment-phase attacks. Table 2 summarizes their attack objectives.

2.1 Training-Phase Attacks

Training-phase attacks assume the ability to modify the training data to achieve malicious attempts on the resulting model, which can be realized through noisy data collection such as crowdsourcing. Specifically, the memorization effect of deep learning models (Zhang et al. 2017; Carlini et al. 2019b) can be leveraged as vulnerabilities. We note that sometimes the term "data poisoning" entails both poisoning and backdoor attacks, though their attack objectives are different.

Poisoning attack aims to design a poisoned dataset D_{poison} such that models trained on D_{poison} will fail to generalize on D_{test} (i.e. $\hat{y}_\theta(x_{\text{test}}) \neq y_{\text{test}}$) while the training loss remains similar to clean data. The poisoned dataset D_{poison} can be created by modifying the original training dataset D_{train}, such as label flipping, data addition/deletion, and feature modification. The rationale is that training on D_{poison} will land on a "bad" local minimum of model parameters.

To control the amount of data modification and reduce the overall accuracy on D_{test} (i.e. test accuracy), poisoning attack often assumes the knowledge of the target model and its training method (Jagielski et al. 2018). (Liu et al. 2020b) proposes black-box poisoning with additional conditions on the training loss function. Targeted poisoning attack aims at manipulating the prediction of a subset of data samples in D_{test}, which can be accomplished by clean-label poisoning (small perturbations to a subset of D_{train} while keeping their labels intact) (Shafahi et al. 2018; Zhu et al. 2019) or gradient-matching poisoning (Geiping et al. 2021).

Backdoor attack is also known as Trojan attack. The central idea is to embed a universal trigger Δ to a subset of data samples in D_{train} with a modified target label t (Gu et al. 2019). Examples of trigger patterns are a small patch in images and a specific text string in sentences. Typically, backdoor attack only assumes access to the training data and does not assume the knowledge of the model and its training. The model f_θ trained on the tampered data is called a backdoored (Trojan) model. Its attack objective has two folds: (i) High standard accuracy in the absence of trigger – the backdoored model should behave like a normal model (same model trained on untampered data), i.e., $\hat{y}_\theta(x_{\text{test}}) = y_{\text{test}}$. (ii) High attack success rate in the presence of trigger – the backdoored model will predict any data input with the trigger as the target label t, i.e., $\hat{y}_\theta(x_{\text{test}} + \Delta) = t$. Therefore, backdoor attack is stealthy and insidious. The trigger pattern can also be made input-aware and dynamic (Nguyen and Tran 2020).

There is a growing concern of backdoor attacks in emerging machine learning systems featuring collaborative model training with local private data, such as federated learning (Bhagoji et al. 2019; Bagdasaryan et al. 2020). Backdoor attacks can be made more insidious by leveraging the innate local model/data heterogeneity and diversity (Zawad et al. 2021). (Xie et al. 2020) proposes distributed backdoor attacks by trigger pattern decomposition among malicious clients to make the attack more stealthy and effective. We also refer the readers to the detailed survey of these two attacks in (Goldblum et al. 2020).

Norm	Meaning
ℓ_0	number of modified features
ℓ_1	total changes in modified features
ℓ_2	Euclidean distance between x and x_{adv}
ℓ_∞	maximal change in modified features

Table 3: ℓ_p norm similarity measures for additive perturbation $\delta = x_{\text{adv}} - x$. The change in each feature (dimension) between x and x_{adv} is measured in absolute value.

2.2 Deployment-Phase Attacks

The objective of deployment-phase attacks is to find a "similar" example $\mathcal{T}(x)$ of x such that the fixed model f_θ will evade its prediction from the original groundtruth label y. The evasion condition can be further separated into two cases: (i) untargeted attack such that $f_\theta(x) = y$ but $f_\theta(\mathcal{T}(x)) \neq y$, or (ii) targeted attack such that $f_\theta(x) = y$ but $f_\theta(\mathcal{T}(x)) = t$, $t \neq y$. Such $\mathcal{T}(x)$ is known as an adversarial example[4] of x (Biggio et al. 2013; Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015), and it can be interpreted as out-of-distribution sample or generalization error (Stutz, Hein, and Schiele 2019).

Data Similarity. Depending on data characteristics, specifying a transformation function $\mathcal{T}(\cdot)$ that preserves data similarity between an original sample x and its transformed sample $\mathcal{T}(x)$ is a core mission for evasion attacks. The transformation can also be a composite function of semantic-preserving changes (Hsiung et al. 2022). A common practice to select $\mathcal{T}(\cdot)$ is through a simple additive perturbation δ such that $x_{\text{adv}} = x + \delta$, or through domain-specific knowledge such as rotation, object translation, and color changes for image data (Hosseini and Poovendran 2018; Engstrom et al. 2019). For additive perturbation (either on data input or parameter(s) simulating semantic changes), the ℓ_p norm ($p \geq 1$) of δ defined as $\|\delta\|_p \triangleq \left(\sum_{i=1}^{d} |\delta_i|^p\right)^{1/p}$ and the pseudo norm ℓ_0 are surrogate metrics for measuring similarity distance. Table 3 summarizes popular choices of ℓ_p norms and their meanings. Take image as an example, ℓ_0 norm is used to design few-pixel (patch) attacks (Su, Vargas, and Sakurai 2019), ℓ_1 norm is used to generate sparse and small perturbations (Chen et al. 2018b), ℓ_∞ norm is used to confine maximal changes in pixel values (Szegedy et al. 2014), and mixed ℓ_p norms can also be used (Xu et al. 2019b).

Evasion Attack Taxonomy. Evasion attacks can be categorized based on attackers' knowledge of the target model. Figure 2 illustrates the taxonomy of different attack types.

White-box attack assumes complete knowledge about the target model, including model architecture and model parameters. Consequently, an attacker can exploit the auto

[4]Sometimes adversarial example may carry a broader meaning of any data sample x_{adv} that leads to incorrect prediction and therefore dropping the dependence to a reference sample x, such as unrestricted adversarial examples (Brown et al. 2018).

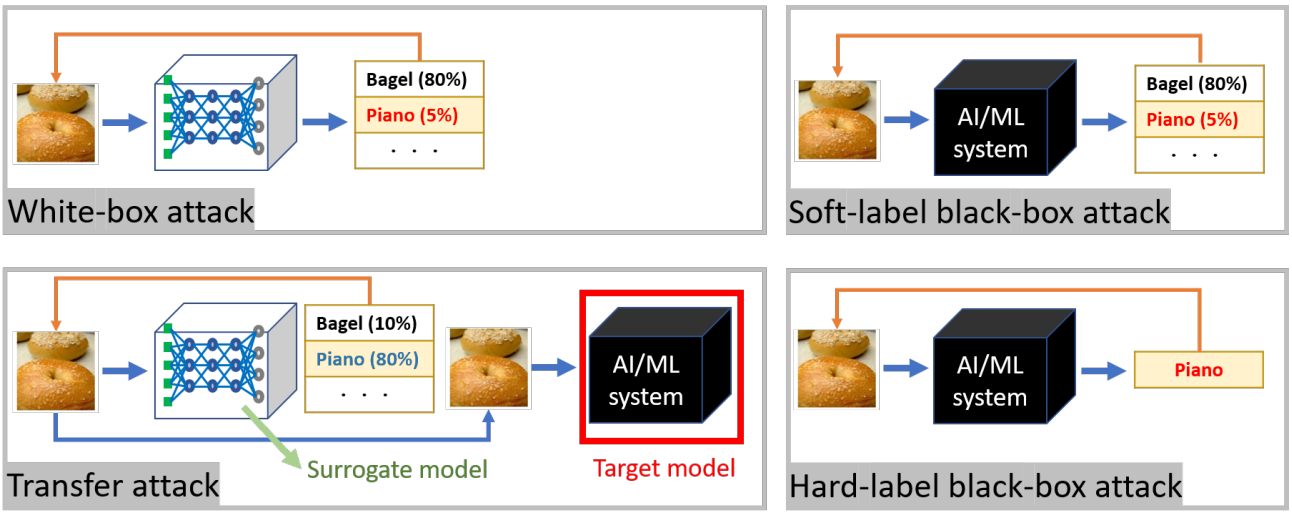


Figure 2: Taxonomy and illustration of evasion attacks.

differentiation function offered by deep learning packages, such as backpropagation (input gradient) from the model output to the model input, to craft adversarial examples.

Black-box attack assumes an attacker can only observe the model prediction of a data input (that is, a query) and does not know any other information. The target model can be viewed as a black-box function and thus backpropagation for computing input gradient is infeasible without knowing the model details. In the *soft-label black-box attack* setting, an attacker can observe (parts of) class predictions and their associated confidence scores. In the *hard-label black-box attack* (decision-based) setting, an attacker can only observe the top-1 label prediction, which is the least information required to be returned to remain the utility of the model. In addition to attack success rate, query efficiency is also an important metric for the performance evaluation of black-box attacks.

Transfer attack is a branch of black-box attack that uses adversarial examples generated from a white-box surrogate model to attack the target model. The surrogate model can be either pre-trained (Liu et al. 2017) or distilled from a set of data samples with soft labels given by the target model for training (Papernot, McDaniel, and Goodfellow 2016; Papernot et al. 2017).

Attack formulation. The process of finding an adversarial perturbation δ can be formulated as a constrained optimization problem with a specified attack loss $g(\delta|x, y, t, \theta)$ reflecting the attack objective (t is omitted for untargeted attack). The variation in problem formulations and solvers will lead to different attack algorithms. We specify three examples below. Without loss of generality, we use ℓ_p norm as the similarity measure (distortion) and untargeted attack as the objective, and assume that all feasible data inputs lie in the scaled space $\mathcal{S} = [0, 1]^d$.

• Minimal-distortion formulation:

$$\text{Minimize}_{\delta: x+\delta\in\mathcal{S}} \ \|\delta\|_p \ \text{ subject to } \hat{y}_\theta(x+\delta) \neq \hat{y}_\theta(x) \quad (1)$$

• Penalty-based formulation:

$$\text{Minimize}_{\delta: x+\delta\in\mathcal{S}} \ \|\delta\|_p + \lambda \cdot g(\delta|x, y, \theta) \quad (2)$$

• Budget-based (norm bounded) formulation:

$$\text{Minimize}_{\delta: x+\delta\in\mathcal{S}} \ g(\delta|x, y, \theta) \ \text{ subject to } \|\delta\|_p \leq \epsilon \quad (3)$$

For untargeted attacks, the attacker's loss can be the negative classification loss $g(\delta|x, y, \theta) = -\mathsf{loss}(f_\theta(x+\delta), y)$ or the truncated class margin loss (using either logit or softmax output) defined as $g(\delta|x, y, \theta) = \max\{[\mathsf{logit}(x+\delta)]_y - \max_{k\in[K],k\neq y}[\mathsf{logit}(x+\delta)]_k + \kappa, 0\}$. The margin loss suggests that $g(\delta|x, y, \theta)$ achieves minimal value (i.e. 0) when the top-1 class confidence score excluding the original class y satisfies $\max_{k\in[K],k\neq y}[\mathsf{logit}(x+\delta)]_k \geq \mathsf{logit}(x+\delta)]_y + \kappa$, where $\kappa \geq 0$ is a tuning parameter governing their confidence gap. Similarly, for targeted attacks, the attacker's loss can be $g(\delta|x, y, t, \theta) = \mathsf{loss}(f_\theta(x+\delta), t)$ or $g(\delta|x, y, t, \theta) = \max\{\max_{k\in[K],k\neq t}[\mathsf{logit}(x+\delta)]_k - [\mathsf{logit}(x+\delta)]_t + \kappa, 0\}$. When implementing black-box attacks, the logit margin loss can be replaced with the observable model output $\log f_\theta$.

The attack formulation can be generalized to the *universal perturbation* setting such that it simultaneously evades all model predictions. The universality can be w.r.t. data samples (Moosavi-Dezfooli et al. 2017), model ensembles (Tramèr et al. 2018), or various data transformations (Athalye and Sutskever 2018). (Wang et al. 2021a) shows that min-max optimization can yield effective universal perturbations.

Selected Attack Algorithms. We show some white-box and black-box attack algorithms driven by the three aforementioned attack formulations. For the minimal-distortion formulation, the attack constraint $\hat{y}_\theta(x+\delta) \neq \hat{y}_\theta(x)$ can be rewritten as $\max_{k\in[K],k\neq y}[f_\theta(x+\delta)]_k \geq [f_\theta(x+\delta)]_y$, which can be used to linearize the local decision boundary around x and allow for efficient projection to the closest linearized decision boundary, leading to white-box attack algorithms such as DeepFool (Moosavi-Dezfooli, Fawzi, and

Frossard 2016) and fast adaptive boundary (FAB) attack (Croce and Hein 2020). For the penalty-based formulation, one can use change-of-variable on δ to convert to an unconstrained optimization problem and then use binary search on λ to find the smallest λ leading to successful attack (i.e., $g = 0$), known as Carlini-Wagner (C&W) white-box attack (Carlini and Wagner 2017b). For the budget-based formulation, one can apply projected gradient descent (PGD), leading to the white-box PGD attack (Madry et al. 2018). Attack algorithms using input gradients of the loss function are called gradient-based attacks.

Black-box attack algorithms often adopt either the penalty-based or budget-based formulation. Since the input gradient of the attacker's loss is unavailable to obtain in the black-box setting, one principal approach is to perform gradient estimation using model queries and then use the estimated gradient to replace the true gradient in white-box attack algorithms, leading to the zeroth-order optimization (ZOO) based black-box attacks (Chen et al. 2017). The choices in gradient estimators (Tu et al. 2019; Nitin Bhagoji et al. 2018; Liu et al. 2018, 2019; Ilyas et al. 2018) and ZOO solvers (Zhao et al. 2019, 2020b; Ilyas, Engstrom, and Madry 2019) will give rise to different attack algorithms. Hard-label black-box attacks can still adopt ZOO principles by spending extra queries to explore local loss landscapes for gradient estimation (Cheng et al. 2019, 2020b; Chen, Jordan, and Wainwright 2020), which is more query-efficient than random exploration (Brendel, Rauber, and Bethge 2018). We refer the readers to (Liu et al. 2020a) for more details on ZOO methods and applications.

Physical adversarial example is a prediction-evasive physical adversarial object. Examples include stop sign (Eykholt et al. 2018), eyeglass (Sharif et al. 2016), physical patch (Brown et al. 2017), 3D printing (Athalye and Sutskever 2018), T-shirt (Xu et al. 2020b), and facial makeup (Lin et al. 2022).

3 Defenses and Verification

Defenses are adversarial threat detection and mitigation strategies, which can be divided into *empirical* and *certified* defenses. We note that the interplay between attack and defense is essentially a cat-and-mouse game. Many seemingly successful empirical defenses were later weakened by advanced attacks that are defense-aware, which gives a false sense of adversarial robustness due to *information obfuscation* (Carlini and Wagner 2017a; Athalye, Carlini, and Wagner 2018). Consequently, defenses are expected to be fully evaluated against the best possible *adaptive attacks* that are defense-aware (Carlini et al. 2019a; Tramer et al. 2020).

While *empirical robustness* refers to the model performance against a set of known attacks, it may fail to serve as a proper robustness indicator against advanced and unseen attacks. To address this issue, *certified robustness* is used to ensure the model is provably robust given a set of attack conditions (threat models). *Verification* can be viewed as a passive certified defense in the sense that its goal is to quantify a given model's (local) robustness with guarantees.

3.1 Empirical Defenses

Empirical defenses are hardening methods applied during the training/deployment phase to improve adversarial robustness without provable guarantees of their effectiveness.

For training-phase attacks, data filtering and model fine-tuning are major approaches. For instance, (Tran, Li, and Madry 2018) shows removing outliers using learned latent representations and retraining the model can reduce the poison effect. To inspect whether a pre-trained model has a backdoor or not, Neural Cleanse (Wang et al. 2019) reverse-engineers potential trigger patterns for detection. (Wang et al. 2020) proposes data-efficient detectors that require only one sample per class and are made data-free for convolutional neural networks. (Zhao et al. 2020a) exploits the mode connectivity in the loss landscape to recover a backdoored model using limited clean data.

For deployment-phase attacks, adversarial input detection schemes that exploit data characteristics such as spatial or temporal correlations are shown to be effective, such as the detection of audio adversarial example using temporal dependency (Yang et al. 2019). For training adversarially robust models, *adversarial training* that aims to minimize the worst-case loss evaluated by perturbed examples generated during training is so far the strongest empirical defense (Madry et al. 2018). Specifically, the standard formulation of adversarial training can be expressed as the following min-max optimization over training samples $\{x_i, y_i\}_{i=1}^n$:

$$\min_{\theta} \sum_{i=1}^{n} \max_{\|\delta_i\|_p \le \epsilon} \mathsf{loss}(f_\theta(x_i + \delta), y) \tag{4}$$

The worst-case loss corresponding to the inner maximization step is often evaluated by gradient-based attacks such as PGD attack (Madry et al. 2018). Variants of adversarial training methods such as TRADES (Zhang et al. 2019) and customized adversarial training (CAT) (Cheng et al. 2020a) have been proposed for improved robustness. (Cheng et al. 2021) proposes attack-independent robust training based on self-progression. On 18 different ImageNet pre-trained models, (Su et al. 2018) unveils an undesirable trade-off between standard accuracy and adversarial robustness. This trade-off can be improved with unlabeled data (Carmon et al. 2019; Stanforth et al. 2019). A similar study on vision transformers is presented in (Shao et al. 2022). (Paul and Chen 2022) extends the analysis to a variety of robustness aspects beyond adversarial robustness.

3.2 Certified Defenses

Certified defenses provide performance guarantees on hardened models. Adversarial attacks are ineffective if their threat models fall within the provably robust conditions.

For training-phase attacks, (Steinhardt, Koh, and Liang 2017) proposes certified data sanitization against poisoning attacks. (Weber et al. 2020) proposes randomized data training for certified defense against backdoor attacks. For deployment-phase attacks, randomized smoothing is an effective and model-agnostic approach that adds random noises to data input to "smooth" the model and perform

majority voting on the model predictions. The certified radius (region) in ℓ_p-norm perturbation ensuring consistent class prediction can be computed by information-theoretical approach (Li et al. 2019), differential privacy (Lecuyer et al. 2019), Neyman-Pearson lemma (Cohen, Rosenfeld, and Kolter 2019), or higher-order certification (Mohapatra et al. 2020a). The certified defense is also recently extended to robustify black-box victim models by leveraging the technique of denoised randomized smoothing (Salman et al. 2020; Zhang et al. 2022b).

3.3 Verification

Verification is often used in certifying local robustness against evasion attacks. Given a neural network f_θ and a data sample x, verification (in its simplest form) aims to maximally certify an ℓ_p-norm bounded radius r on the perturbation δ to ensure the model prediction on the perturbed sample $x + \delta$ is consistent as long as δ is within the certified region. That is, for any δ such that $\|\delta\|_p \leq r$, $\hat{y}_\theta(x + \delta) = \hat{y}_\theta(x)$. The certified radius is a robustness certificate relating to the distance to the closest decision boundary, which is computationally challenging (NP-complete) for neural networks (Katz et al. 2017). However, its estimate (hence not a certificate) can be efficiently computed and used as a model-agnostic robustness metric, such as the CLEVER score (Weng et al. 2018b). To address the non-linearity induced by layer propagation in neural networks, solving for a certified radius is often cast as a relaxed optimization problem. The methods include convex polytope (Wong and Kolter 2018), semidefinite programming (Raghunathan, Steinhardt, and Liang 2018), dual optimization (Dvijotham et al. 2018), layer-wise linear bounds (Weng et al. 2018a), and interval bound propagation (Gowal et al. 2019). The verification tools are also expanded to support general network architectures (Zhang et al. 2018; Boopathy et al. 2019; Xu et al. 2020a) and semantic adversarial examples (Mohapatra et al. 2020b). The intermediate certified results can be used to train a more certifiable model (Wong et al. 2018; Boopathy et al. 2021). However, scalability to large-sized neural networks remains a major challenge in verification.

4 Remarks and Discussion

Here we make several concluding remarks and discussions.

Novel Applications. The insights from studying adversarial robustness have led to several new use cases. Adversarial perturbation and data poisoning are used in generating contrastive explanations (Dhurandhar et al. 2018), personal privacy protection (Shan et al. 2020), data/model watermarking and fingerprinting (Sablayrolles et al. 2020; Aramoon, Chen, and Qu 2021; Wang et al. 2021c), data-limited transfer learning (Tsai, Chen, and Ho 2020; Yang, Tsai, and Chen 2021), and visual prompting (Bahng et al. 2022; Chen et al. 2022b,a). Adversarial examples with proper design are also efficient data augmentation tools to simultaneously improve model generalization and adversarial robustness (Hsu et al. 2022; Lei et al. 2019). Other noteworthy applications include image synthesis (Santurkar et al. 2019) generating contrastive explanations (Dhurandhar et al. 2018), robust text CAPTCHAs (Shao et al. 2021), reverse engineering of deception (Gong et al. 2022), uncertainty calibration (Tang, Chen, and Ho 2022), and molecule discovery (Hoffman et al. 2022).

Adversarial Robustness Beyond Classification and Input Perturbation. The formulations and principles in attacks and defenses for classification can be analogously applied to other machine learning tasks. Examples include sequence-to-sequence translation (Cheng et al. 2020c) and image captioning (Chen et al. 2018a). Beyond input perturbation, the robustness of model parameter perturbation (Tsai et al. 2021) also relates to model quantification (Weng et al. 2020) and energy-efficient inference (Stutz et al. 2020).

Instilling Adversarial Robustness into Foundation Models. As foundation models (Bommasani et al. 2021) adapt task-independent pre-training for general representation learning followed by task-specific fine-tuning for fast adaptation, it is of utmost importance to understand (i) how to incorporate adversarial robustness into foundation model pre-training and (ii) how to maximize adversarial robustness transfer from pre-training to fine-tuning. (Fan et al. 2021; Wang et al. 2021b) show promising results in adversarial robustness preservation and transfer in meta learning and contrastive learning. The rapid growth and intensifying demand on foundation models create a unique opportunity to advocate adversarial robustness as a necessary native property in next-generation trustworthy AI tools and call for novel methods for evaluating *representational robustness*, such as in (Ko et al. 2022).

Practical Adversarial Robustness at Scale. From an industrial viewpoint, current solutions to strengthen adversarial robustness may not be ideal because of the unacceptable performance drop on the original task and the poor scalability of effective defenses to industry-scale large deep learning models and systems. While there are some efforts for enabling adversarial training at scale, such as (Zhang et al. 2022a), the notable tradeoff between standard accuracy and robust accuracy may not be a favorable solution for business adoption. An alternative can be rethinking the evaluation methodology of adversarial robustness. For example, instead of aiming to mitigate the robustness-accuracy tradeoff, we can compare the unilateral robustness gain under the constraint of making minimal (or even zero) harm to the original model utility (e.g. test accuracy). Moreover, an ideal defense should be lightweight and deployable in a plug-and-play manner for any given model, instead of demanding to train a model from scratch for improved robustness.

References

Aramoon, O.; Chen, P.-Y.; and Qu, G. 2021. Don't Forget to Sign the Gradients! *MLSyS*, 3.

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ICML*.

Athalye, A.; and Sutskever, I. 2018. Synthesizing robust adversarial examples. *ICML*.

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How to backdoor federated learning. *AISTATS*.

Bahng, H.; Jahanian, A.; Sankaranarayanan, S.; and Isola, P. 2022. Visual Prompting: Modifying Pixel Space to Adapt Pre-trained Models. *arXiv preprint arXiv:2203.17274*.

Bhagoji, A. N.; Chakraborty, S.; Mittal, P.; and Calo, S. 2019. Analyzing federated learning through an adversarial lens. *ICML*, 634–643.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion attacks against machine learning at test time. *ECML PKDD*.

Biggio, B.; and Roli, F. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84: 317–331.

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

Boopathy, A.; Weng, T.-W.; Chen, P.-Y.; Liu, S.; and Daniel, L. 2019. CNN-cert: An efficient framework for certifying robustness of convolutional neural networks. *AAAI*.

Boopathy, A.; Weng, T.-W.; Liu, S.; Chen, P.-Y.; Zhang, G.; and Daniel, L. 2021. Fast Training of Provably Robust Neural Networks by SingleProp. *AAAI*.

Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. *ICLR*.

Brown, T. B.; Carlini, N.; Zhang, C.; Olsson, C.; Christiano, P.; and Goodfellow, I. 2018. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*.

Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665*.

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

Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; and Song, D. 2019b. The secret sharer: Evaluating and testing unintended memorization in neural networks. *USENIX Security*, 267–284.

Carlini, N.; and Wagner, D. 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods. *ACM Workshop on Artificial Intelligence and Security*, 3–14.

Carlini, N.; and Wagner, D. 2017b. Towards evaluating the robustness of neural networks. *IEEE S&P*, 39–57.

Carmon, Y.; Raghunathan, A.; Schmidt, L.; Liang, P.; and Duchi, J. C. 2019. Unlabeled data improves adversarial robustness. *NeurIPS*.

Chen, A.; Lorenz, P.; Yao, Y.; Chen, P.-Y.; and Liu, S. 2022a. Visual Prompting for Adversarial Robustness. *arXiv preprint arXiv:2210.06284*.

Chen, A.; Yao, Y.; Chen, P.-Y.; Zhang, Y.; and Liu, S. 2022b. Understanding and Improving Visual Prompting: A Label-Mapping Perspective. *arXiv preprint arXiv:2211.11635*.

Chen, H.; Zhang, H.; Chen, P.-Y.; Yi, J.; and Hsieh, C.-J. 2018a. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. *ACL*, 1: 2587–2597.

Chen, J.; Jordan, M. I.; and Wainwright, M. J. 2020. Hopskipjumpattack: A query-efficient decision-based attack. *IEEE S&P*.

Chen, P.-Y.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C.-J. 2018b. EAD: elastic-net attacks to deep neural networks via adversarial examples. *AAAI*, 10–17.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks Without Training Substitute Models. *ACM Workshop on Artificial Intelligence and Security*, 15–26.

Cheng, M.; Chen, P.-Y.; Liu, S.; Chang, S.; Hsieh, C.-J.; and Das, P. 2021. Self-Progressing Robust Training. *AAAI*.

Cheng, M.; Le, T.; Chen, P.-Y.; Yi, J.; Zhang, H.; and Hsieh, C.-J. 2019. Query-efficient hard-label black-box attack: An optimization-based approach. *ICLR*.

Cheng, M.; Lei, Q.; Chen, P.-Y.; Dhillon, I.; and Hsieh, C.-J. 2020a. Cat: Customized adversarial training for improved robustness. *arXiv preprint arXiv:2002.06789*.

Cheng, M.; Singh, S.; Chen, P. H.; Chen, P.-Y.; Liu, S.; and Hsieh, C.-J. 2020b. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack. *ICLR*.

Cheng, M.; Yi, J.; Zhang, H.; Chen, P.-Y.; and Hsieh, C.-J. 2020c. Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples. *AAAI*.

Cohen, J. M.; Rosenfeld, E.; and Kolter, J. Z. 2019. Certified adversarial robustness via randomized smoothing. *ICML*.

Croce, F.; and Hein, M. 2020. Minimally distorted adversarial examples with a fast adaptive boundary attack. *ICML*, 2196–2205.

Dhurandhar, A.; Chen, P.-Y.; Luss, R.; Tu, C.-C.; Ting, P.; Shanmugam, K.; and Das, P. 2018. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *NeurIPS*.

Dvijotham, K.; Stanforth, R.; Gowal, S.; Mann, T. A.; and Kohli, P. 2018. A Dual Approach to Scalable Verification of Deep Networks. *UAI*, 1(2): 3.

Engstrom, L.; Tran, B.; Tsipras, D.; Schmidt, L.; and Madry, A. 2019. Exploring the landscape of spatial robustness. *ICML*, 1802–1811.

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust physical-world attacks on deep learning visual classification. *CVPR*, 1625–1634.

Fan, L.; Liu, S.; Chen, P.-Y.; Zhang, G.; and Gan, C. 2021. When Does Contrastive Learning Preserve Adversarial Robustness from Pretraining to Finetuning? *NeurIPS*, 34.

Geiping, J.; Fowl, L.; Huang, W. R.; Czaja, W.; Taylor, G.; Moeller, M.; and Goldstein, T. 2021. Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching. *ICLR*.

Goldblum, M.; Tsipras, D.; Xie, C.; Chen, X.; Schwarzschild, A.; Song, D.; Madry, A.; Li, B.; and Goldstein, T. 2020. Data Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses. *arXiv preprint arXiv:2012.10544*.

Gong, Y.; Yao, Y.; Li, Y.; Zhang, Y.; Liu, X.; Lin, X.; and Liu, S. 2022. Reverse Engineering of Imperceptible Adversarial Image Perturbations. *arXiv preprint arXiv:2203.14145*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. *ICLR*.

Gowal, S.; Dvijotham, K. D.; Stanforth, R.; Bunel, R.; Qin, C.; Uesato, J.; Arandjelovic, R.; Mann, T.; and Kohli, P. 2019. Scalable verified training for provably robust image classification. *ICCV*, 4842–4851.

Gu, T.; Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2019. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access*, 7: 47230–47244.

Hoffman, S. C.; Chenthamarakshan, V.; Wadhawan, K.; Chen, P.-Y.; and Das, P. 2022. Optimizing molecules using efficient queries from property evaluations. *Nature Machine Intelligence*, 4(1): 21–31.

Hosseini, H.; and Poovendran, R. 2018. Semantic adversarial examples. *CVPR Workshops*, 1614–1619.

Hsiung, L.; Tsai, Y.-Y.; Chen, P.-Y.; and Ho, T.-Y. 2022. Towards Compositional Adversarial Robustness: Generalizing Adversarial Training to Composite Semantic Perturbations. *arXiv preprint arXiv:2202.04235*.

Hsu, C.-Y.; Chen, P.-Y.; Lu, S.; Liu, S.; and Yu, C.-M. 2022. Adversarial Examples can be Effective Data Augmentation for Unsupervised Machine Learning. In *AAAI*.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box Adversarial Attacks with Limited Queries and Information. *ICML*.

Ilyas, A.; Engstrom, L.; and Madry, A. 2019. Prior convictions: Black-box adversarial attacks with bandits and priors. *ICLR*.

Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; and Li, B. 2018. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. *IEEE S&P*, 19–35.

Katz, G.; Barrett, C.; Dill, D. L.; Julian, K.; and Kochenderfer, M. J. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. *International Conference on Computer Aided Verification*, 97–117.

Ko, C.-Y.; Chen, P.-Y.; Mohapatra, J.; Das, P.; and Daniel, L. 2022. SynBench: Task-Agnostic Benchmarking of Pretrained Representations using Synthetic Data. *arXiv preprint arXiv:2210.02989*.

Kumar, R. S. S.; Nyström, M.; Lambert, J.; Marshall, A.; Goertzel, M.; Comissoneru, A.; Swann, M.; and Xia, S. 2020. Adversarial machine learning-industry perspectives. *IEEE S&P Workshops*, 69–75.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature*.

Lecuyer, M.; Atlidakis, V.; Geambasu, R.; Hsu, D.; and Jana, S. 2019. Certified robustness to adversarial examples with differential privacy. *IEEE S&P*, 656–672.

Lei, Q.; Wu, L.; Chen, P.-Y.; Dimakis, A. G.; Dhillon, I. S.; and Witbrock, M. 2019. Discrete Adversarial Attacks and Submodular Optimization with Applications to Text Classification. *SysML*.

Li, B.; Chen, C.; Wang, W.; and Carin, L. 2019. Certified adversarial robustness with additive noise. *NeurIPS*.

Lin, C.-S.; Hsu, C.-Y.; Chen, P.-Y.; and Yu, C.-M. 2022. Real-World Adversarial Examples Via Makeup. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2854–2858.

Liu, S.; Chen, P.-Y.; Chen, X.; and Hong, M. 2019. signSGD via Zeroth-Order Oracle. *ICLR*.

Liu, S.; Chen, P.-Y.; Kailkhura, B.; Zhang, G.; Hero, A.; and Varshney, P. K. 2020a. A Primer on Zeroth-Order Optimization in Signal Processing and Machine Learning. *IEEE Signal Processing Magazine*.

Liu, S.; Kailkhura, B.; Chen, P.-Y.; Ting, P.; Chang, S.; and Amini, L. 2018. Zeroth-order stochastic variance reduction for nonconvex optimization. *NeurIPS*, 3731–3741.

Liu, S.; Lu, S.; Chen, X.; Feng, Y.; Xu, K.; Al-Dujaili, A.; Hong, M.; and O'Reilly, U.-M. 2020b. Min-max optimization without gradients: Convergence and applications to black-box evasion and poisoning attacks. *ICML*, 6282–6293.

Liu, Y.; Chen, X.; Liu, C.; and Song, D. 2017. Delving into transferable adversarial examples and black-box attacks. *ICLR*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. *ICLR*.

Mohapatra, J.; Ko, C.-Y.; Weng, T.-W.; Chen, P.-Y.; Liu, S.; and Daniel, L. 2020a. Higher-Order Certification for Randomized Smoothing. *NeurIPS*.

Mohapatra, J.; Weng, T.-W.; Chen, P.-Y.; Liu, S.; and Daniel, L. 2020b. Towards verifying robustness of neural networks against a family of semantic perturbations. *CVPR*, 244–252.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal Adversarial Perturbations. *CVPR*, 86–94.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. *CVPR*, 2574–2582.

Nguyen, A.; and Tran, A. 2020. Input-aware dynamic backdoor attack. *NeurIPS*.

Nitin Bhagoji, A.; He, W.; Li, B.; and Song, D. 2018. Practical Black-box Attacks on Deep Neural Networks using Efficient Query Mechanisms. *ECCV*, 154–169.

Papernot, N.; McDaniel, P.; and Goodfellow, I. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. *ACM Asia Conference on Computer and Communications Security*, 506–519.

Paul, S.; and Chen, P.-Y. 2022. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2071–2081.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018. Certified a against adversarial examples. *ICLR*.

Sablayrolles, A.; Douze, M.; Schmid, C.; and Jégou, H. 2020. Radioactive data: tracing through training. *ICML*.

Salman, H.; Sun, M.; Yang, G.; Kapoor, A.; and Kolter, J. Z. 2020. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems*, 33: 21945–21957.

Santurkar, S.; Tsipras, D.; Tran, B.; Ilyas, A.; Engstrom, L.; and Madry, A. 2019. Image synthesis with a single (robust) classifier. *arXiv preprint arXiv:1906.09453*.

Shafahi, A.; Huang, W. R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; and Goldstein, T. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. *NeurIPS*, 6103–6113.

Shan, S.; Wenger, E.; Zhang, J.; Li, H.; Zheng, H.; and Zhao, B. Y. 2020. Fawkes: Protecting privacy against unauthorized deep learning models. *USENIX Security*.

Shao, R.; Shi, Z.; Yi, J.; Chen, P.-Y.; and Hsieh, C.-J. 2021. Robust Text CAPTCHAs Using Adversarial Examples. *arXiv preprint arXiv:2101.02483*.

Shao, R.; Shi, Z.; Yi, J.; Chen, P.-Y.; and Hsieh, C.-J. 2022. On the Adversarial Robustness of Vision Transformers. *Transactions on Machine Learning Research*.

Sharif, M.; Bhagavatula, S.; Bauer, L.; and Reiter, M. K. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. *ACM CCS*, 1528–1540.

Stanforth, R.; Fawzi, A.; Kohli, P.; et al. 2019. Are Labels Required for Improving Adversarial Robustness? *NeurIPS*.

Steinhardt, J.; Koh, P. W.; and Liang, P. 2017. Certified defenses for data poisoning attacks. *NeurIPS*.

Stutz, D.; Chandramoorthy, N.; Hein, M.; and Schiele, B. 2020. Bit Error Robustness for Energy-Efficient DNN Accelerators. *arXiv preprint arXiv:2006.13977*.

Stutz, D.; Hein, M.; and Schiele, B. 2019. Disentangling adversarial robustness and generalization. *CVPR*, 6976–6987.

Su, D.; Zhang, H.; Chen, H.; Yi, J.; Chen, P.-Y.; and Gao, Y. 2018. Is robustness the cost of accuracy? A comprehensive study on the robustness of 18 deep image classification models. *ECCV*.

Su, J.; Vargas, D. V.; and Sakurai, K. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5): 828–841.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. *ICLR*.

Tang, Y.-C.; Chen, P.-Y.; and Ho, T.-Y. 2022. Neural Clamping: Joint Input Perturbation and Temperature Scaling for Neural Network Calibration. *arXiv preprint arXiv:2209.11604*.

Tramer, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On adaptive attacks to adversarial example defenses. *NeurIPS*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Boneh, D.; and McDaniel, P. 2018. Ensemble Adversarial Training: Attacks and Defenses. *ICLR*.

Tran, B.; Li, J.; and Madry, A. 2018. Spectral signatures in backdoor attacks. *NeurIPS*, 8000–8010.

Tsai, Y.-L.; Hsu, C.-Y.; Yu, C.-M.; and Chen, P.-Y. 2021. Formalizing Generalization and Adversarial Robustness of Neural Networks to Weight Perturbations. *NeurIPs*, 34.

Tsai, Y.-Y.; Chen, P.-Y.; and Ho, T.-Y. 2020. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. *ICML*, 9614–9624.

Tu, C.-C.; Ting, P.; Chen, P.-Y.; Liu, S.; Zhang, H.; Yi, J.; Hsieh, C.-J.; and Cheng, S.-M. 2019. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *AAAI*, 33: 742–749.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. *IEEE S&P*, 707–723.

Wang, J.; Zhang, T.; Liu, S.; Chen, P.-Y.; Xu, J.; Fardad, M.; and Li, B. 2021a. Adversarial attack generation empowered by min-max optimization. *NeurIPS*, 34.

Wang, R.; Xu, K.; Liu, S.; Chen, P.-Y.; Weng, T.-W.; Gan, C.; and Wang, M. 2021b. On Fast Adversarial Robustness Adaptation in Model-Agnostic Meta-Learning. *ICLR*.

Wang, R.; Zhang, G.; Liu, S.; Chen, P.-Y.; Xiong, J.; and Wang, M. 2020. Practical detection of trojan neural networks: Data-limited and data-free cases. *ECCV*, 222–238.

Wang, S.; Wang, X.; Chen, P. Y.; Zhao, P.; and Lin, X. 2021c. Characteristic Examples: High-Robustness, Low-Transferability Fingerprinting of Neural Networks. *IJCAI*.

Weber, M.; Xu, X.; Karlas, B.; Zhang, C.; and Li, B. 2020. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904*.

Weng, T.-W.; Zhang, H.; Chen, H.; Song, Z.; Hsieh, C.-J.; Boning, D.; Dhillon, I. S.; and Daniel, L. 2018a. Towards Fast Computation of Certified Robustness for ReLU Networks. *International Coference on ICML*.

Weng, T.-W.; Zhang, H.; Chen, P.-Y.; Yi, J.; Su, D.; Gao, Y.; Hsieh, C.-J.; and Daniel, L. 2018b. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach. *ICLR*.

Weng, T.-W.; Zhao, P.; Liu, S.; Chen, P.-Y.; Lin, X.; and Daniel, L. 2020. Towards Certificated Model Robustness Against Weight Perturbations. *AAAI*, 6356–6363.

Wong, E.; and Kolter, Z. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. *ICML*.

Wong, E.; Schmidt, F. R.; Metzen, J. H.; and Kolter, J. Z. 2018. Scaling provable adversarial defenses. *NeurIPS*.

Xie, C.; Huang, K.; Chen, P.-Y.; and Li, B. 2020. DBA: Distributed Backdoor Attacks against Federated Learning. *ICLR*.

Xu, K.; Chen, H.; Liu, S.; Chen, P.-Y.; Weng, T.-W.; Hong, M.; and Lin, X. 2019a. Topology attack and defense for graph neural networks: An optimization perspective. *IJCAI*.

Xu, K.; Liu, S.; Zhao, P.; Chen, P.-Y.; Zhang, H.; Fan, Q.; Erdogmus, D.; Wang, Y.; and Lin, X. 2019b. Structured adversarial attack: Towards general implementation and better interpretability. *ICLR*.

Xu, K.; Shi, Z.; Zhang, H.; Huang, M.; Chang, K.-W.; Kailkhura, B.; Lin, X.; and Hsieh, C.-J. 2020a. Automatic perturbation analysis on general computational graphs. *NeurIPS*.

Xu, K.; Zhang, G.; Liu, S.; Fan, Q.; Sun, M.; Chen, H.; Chen, P.-Y.; Wang, Y.; and Lin, X. 2020b. Adversarial t-shirt! evading person detectors in a physical world. *ECCV*.

Yang, C.-H. H.; Tsai, Y.-Y.; and Chen, P.-Y. 2021. Voice2Series: Reprogramming Acoustic Models for Time Series Classification. In *ICML*.

Yang, Z.; Li, B.; Chen, P.-Y.; and Song, D. 2019. Characterizing Audio Adversarial Examples Using Temporal Dependency. *ICLR*.

Zawad, S.; Ali, A.; Chen, P.-Y.; Anwar, A.; Zhou, Y.; Baracaldo, N.; Tian, Y.; and Yan, F. 2021. Curse or Redemption? How Data Heterogeneity Affects the Robustness of Federated Learning. *AAAI*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. *ICLR*.

Zhang, G.; Lu, S.; Zhang, Y.; Chen, X.; Chen, P.-Y.; Fan, Q.; Martie, L.; Horesh, L.; Hong, M.; and Liu, S. 2022a. Distributed adversarial training to robustify deep neural networks at scale. In *Uncertainty in Artificial Intelligence*, 2353–2363. PMLR.

Zhang, H.; Weng, T.-W.; Chen, P.-Y.; Hsieh, C.-J.; and Daniel, L. 2018. Efficient neural network robustness certification with general activation functions. *NeurIPS*, 4944–4953.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. *ICML*, 7472–7482.

Zhang, Y.; Yao, Y.; Jia, J.; Yi, J.; Hong, M.; Chang, S.; and Liu, S. 2022b. How to Robustify Black-Box ML Models? A Zeroth-Order Optimization Perspective. In *International Conference on Learning Representations*.

Zhao, P.; Chen, P.-Y.; Das, P.; Ramamurthy, K. N.; and Lin, X. 2020a. Bridging Mode Connectivity in Loss Landscapes and Adversarial Robustness. *ICLR*.

Zhao, P.; Chen, P.-Y.; Wang, S.; and Lin, X. 2020b. Towards Query-Efficient Black-Box Adversary with Zeroth-Order Natural Gradient Descent. *AAAI*.

Zhao, P.; Liu, S.; Chen, P.-Y.; Hoang, N.; Xu, K.; Kailkhura, B.; and Lin, X. 2019. On the Design of Black-box Adversarial Examples by Leveraging Gradient-free Optimization and Operator Splitting Method. *ICCV*, 121–130.

Zhu, C.; Huang, W. R.; Li, H.; Taylor, G.; Studer, C.; and Goldstein, T. 2019. Transferable clean-label poisoning attacks on deep neural nets. *ICML*, 7614–7623.