

RESEARCH

Open Access



Over-parameterization and adversarial robustness in neural networks: an overview and empirical analysis

Srishti Gupta^{1,2}, Zhang Chen³, Luca Demetrio⁴, Fabio Brau², Xiaoyi Feng³, Zhaoqiang Xia³, Antonio Emanuele Cinà⁴, Maura Pintor², Luca Oneto⁴, Ambra Demontis^{2*}, Battista Biggio² and Fabio Roli^{2,4}

*Correspondence:

Ambra Demontis
ambra.demontis@unica.it

¹Sapienza University of Rome,
Rome, Italy

²University of Cagliari, Cagliari, Italy

³Northwestern Polytechnical
University, Xi'an, China

⁴University of Genoa, Genoa, Italy

Abstract

Thanks to their extensive capacity, over-parameterized neural networks exhibit superior predictive capabilities and generalization. Nevertheless, having a large parameter space is considered one of the main suspects of the neural network vulnerability to adversarial examples—input samples crafted ad hoc to induce a desired misclassification. The relevant literature has presented contradictory claims regarding the robustness of over-parameterized networks. These contradictory findings might be due to the failure of the attack employed to evaluate the robustness of the networks. Previous research has demonstrated that depending on the considered model, the algorithm employed to generate adversarial examples may not function properly, leading to overestimating its robustness. In this work, we empirically study the robustness of over-parameterized networks against adversarial examples. However, unlike the previous works, we also evaluate the considered attack's reliability, employing the tools at the state of the art, to support the results' veracity. Our results show that over-parameterized networks are more robust to adversarial attacks than their under-parameterized counterparts.

Keywords Over-parameterization, Adversarial robustness, Reliable evaluation

1 Introduction

Neural networks have achieved outstanding performance over the last decade. This advancement has been possible not only thanks to the availability of large datasets but also due to the large number of parameters employed. The networks that have reported better performance are often over-parameterized [1]: their capacity is higher than the one needed to interpolate the training data (to fit them perfectly). In modern Machine Learning (ML) practices, rich function classes like neural networks (NN) are highly capable of training models that *generalizes* well on test samples, meaning giving accurate predictions on samples not seen during the training, provided the test data is sampled from the same distribution as the training data [2, 3].



© The Author(s) 2026. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

However, this phenomenon is at odds with the classical understanding of bias-variance tradeoff that predicts lower bias in increasingly complex¹ models at the expense of higher variance, typically supporting parsimonious networks. Such a mismatch in theoretical and empirical understanding has led the community to study the behavior of over-parameterized networks deeply. Belkin et al.'s theory of *double-descent curve* (Fig. 1) confirms that the variance and, subsequently, the risk of the model decreases as the size of the model goes beyond the interpolation point. It is even considered that the lowest generalization error is expected at the infinitely large model. Large neural networks have shown impressive performance at test time across domains, leading to their widescale applicability. However, surprisingly, these models are vulnerable to *adversarial examples*: input samples carefully perturbed to be classified as the attacker desires [4–6]. Computer vision poses a particularly challenging issue since it has been shown that for the attackers, it is sufficient to apply a tiny perturbation, imperceptible to human eyes, to have a sample misclassified with high confidence [7, 8]. For security-sensitive applications such as autonomous cars and face recognition, it is imperative that the models are robust against adversarial examples while performing well on natural samples. From a classical understanding of the optimal size of the neural network, most neural networks, to minimize the empirical risk, were biased to choose the smallest possible model that fits the training data well, i.e., stay on the left side of the double descent curve. Hence, most adversarial evaluations are performed on these under-parameterized networks. Recently, given the massive performance hike of over-parameterized models and their generalizable properties, their robustness to adversarial examples has been investigated in multiple works. However, the claims about the security of these models are contradictory, and the relationship of over-parameterized models with adversarial robustness is unclear. Recent empirical studies showed their near-optimal results even when the neural network and kernel machines trained to interpolation are corrupted with noise [9, 10]. Adversarial examples, distinct from random noise, are carefully crafted samples designed to intentionally target the vulnerability of the model. One of the major reasons for these contradictory results may be the unreliable evaluation of adversarial robustness. In various recent works where a robustness evaluation was performed, adversarial attacks were found to have failed [11], i.e., the attacks generated were ineffective and hence useless for evaluation, giving a false sense of security [12]. However, the articles assessing the robustness of overparameterized networks with respect to their non-overparameterized counterpart have never investigated the effectiveness of the attack employed to perform the robustness evaluation. We evaluate the adversarial robustness

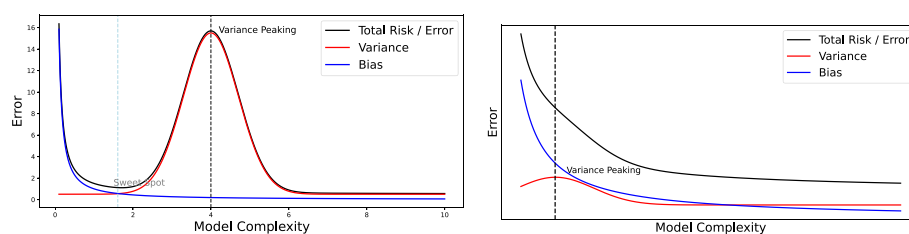


Fig. 1 Risk curves. The bias is represented in blue and the variance is represented in red. Left plot: Double-descent risk curve; Right plot: The risk when, as for deep neural networks, the bias dominates

¹The classical bounds in statistical learning theory point to the size of the model in terms of the number of parameters as a measure of complexity.

of models trained across a spectrum of capacities against Projected Gradient Descent (PGD) attacks. To be consistent with best practices and to reliably evaluate robustness, we test the models using the AutoAttack framework. To avoid known attack failures, we also used tools to verify the attack's effectiveness. The empirical results from the experiments suggest that highly parameterized models are more robust than under-parameterized models. We discuss the background and the notations in Sect. 2, the methodology adopted to conduct experiments in Sect. 3, and experimental setup and results in Sect. 4. We conclude by discussing the major takeaways and future research directions in Sect. 5. The code is available at <https://github.com/pralab/overparam-adv>

2 Over-parameterization and adversarial robustness

In this section, we discuss the need and application of over-parameterized models, their vulnerability to adversarial attacks, and what the existing literature suggests on the robustness of these large models.

2.1 Over-parameterization and double-descent curve

When designing a model architecture to solve an ML problem, from classical understanding, we choose a model that is rich enough to capture the underlying feature space without fitting too closely with the training data (or overfitting) and simple enough to avoid not being able to learn patterns in the data (or underfitting). The balance between underfitting and overfitting is widely known as *bias-variance tradeoff* plotted on a risk capacity curve as in Fig. 1. From this understanding, we would want to find a "sweet spot" where the model has enough capacity to perform well on test data while keeping the training error low. Having a large model with zero training error is conventionally avoided, as stated by Trevor Hastie in [13], "a model with zero training error is overfit to the training data and will typically generalize poorly" which is still a widely accepted notion. However, large NN are highly expressive and can efficiently optimize many parameters to fit the training data. Large number of parameters of the neural network, traditionally considered a case of overfitting, are better suited for generalization [14]. Further, Belkin et al. [1] who coined the term *double-descent curve* (left plot of Fig. 1) suggested that when training a model on the increasing function capacity (referred to the size of network) until the point of zero training error/interpolation, typically have high risk and poor generalization error. Although, as explained in [15] it is not always a double descent curve, especially for deep neural networks, for which the bias usually dominates with respect to the variance and thus the risk is monotonically decreasing (right plot of Fig. 1), unless there is noise in the training labels. Training beyond the interpolation point (in over-parameterized regime), it is observed that the variance of the model decreases, which leads to the overall decrease in the risk, giving improved generalization performance. Further work by Nakkiran et al. [16] showed that this generalization trend is not just a function of model capacity but also a function of the number of epochs in training. In some literature, it is also argued that good generalization performance is due to the inductive bias of gradient-based algorithms, e.g. [17, 18].

2.2 Adversarial robustness

Despite the unprecedented success and wide adoption of large NN in various domains and their outstanding performances, these models are known to be highly vulnerable to

carefully crafted perturbations of the input. For example, consider an input image x , a vector of pixel information, there can exist visually similar images x' with different classifications. Such manipulated input samples are called adversarial examples. These are the results of an optimization problem formulated as:

$$\max_{x'} l(x', y, \theta), \quad (1)$$

$$\text{s.t. } \|x' - x\|_p \leq \epsilon, \quad (2)$$

$$x_{lb} \preceq x' \preceq x_{ub} \quad (3)$$

where $l(\cdot)$ is a loss function, $\|x' - x\|_p$ is the l_p norm constraint bounded by maximum perturbation ϵ . Equation 3 is the box-constraint where $u \preceq v$ means, each element in u has to be smaller than or equal to the corresponding element in v to bound the values of attack sample x' .

To measure the adversarial robustness of the models, we use *Security Evaluation Curves (SEC)*. SEC evaluates the ML models for the maximum error obtained against an increasingly admissible level of perturbations of the input images. As we will see later in the experiments, SEC is used to evaluate all the models considered. Models with a smaller area under the curve suggest higher robustness and vice versa.

2.3 Adversarial robustness of over-parameterized network

While a considerable number of works [19–23] assess the robustness of adversarially trained networks, which is beyond the scope of this article, the findings of related works suggest that overparameterization decreases robustness. In the seminal work by Fu et al. [24], the authors claim that decreasing the number of nodes reduces the sensitivity to small perturbations of the input samples. Sagawa et al. [25] suggested that the over-parameterization can cause the exacerbation of *spurious correlation*, i.e., the relation between the biased features of input like background, texture, or unimportant pixels with the labels. However, in [26], the authors showed that wider neural networks tend to be more robust, as assessed by the PGD attack. The results in this paper were also replicated in [27] and are supported by [28], where the authors claim that achieving robustness may require exponentially more parameters than achieving accuracy on clean test data. The contradiction present in the literature has been highlighted in [29], which provides a theoretical analysis that only encompasses two-layer neural networks, and in [30], which, with an analysis limited to linear models, shows that over-parameterization may be either a source of additional robustness or brittleness. The analysis performed on deep neural networks by the works affirming that over-parameterization increases robustness [26, 27] is empirical and based on the PGD attack, and they have not employed any tool to check the reliability of the obtained results. Therefore, the attack might have failed [31, 32], providing unreliable results that have caused a false sense of robustness for the overparameterized networks. Because gradient-based attacks can fail in many cases [31, 32], e.g., if the employed hyperparameters are incorrect or the decision surface is not smooth enough, they can not achieve convergence, providing a suboptimal point that may not be able to evade the classifier. The goal of this work is to verify, employing the tools available at the state of the art, if a known failure is the reason that led to the abovementioned contradiction between the state-of-the-art works.

Analyzing the reliability of the empirical analysis to assess the robustness of overparametrized networks, we have demonstrated that it does not present any failure known to date; thus, the higher robustness of overparametrized networks is not due to known experimental problems.

3 Study design

In this section, we discuss the assumptions and methodology considered to conduct the experiments. For the evaluation of the over-parameterized models, we consider a simple yet widely used case of supervised learning. In this work, range of models (with different sizes) are trained on training data and then evaluated with benign and adversarial setting at test time. Before diving deep into the considerations of the threat model used in this paper and their evaluations, we first introduce the notations used throughout the paper.

3.1 Notations

We consider a data space $[\mathcal{X}, \mathcal{Y}]$ where $\mathcal{X}$ is a d -dimensional feature space of input samples and $\mathcal{Y}$ is a label space with c -classes such that $\mathcal{Y} \in [c]$. The training data is represented as $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^n$ where n is the size of the training set. We use $L(\mathcal{D}, \theta)$ to represent *loss* incurred by classifier $f \in \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$ where $\mathcal{H}$ is a class of function. Typically the loss is computed by averaging the sample-wise loss function $l(f(\mathbf{x}), y)$ over the training set, i.e., $\frac{1}{n} \sum_{i=1}^n l(f(\mathbf{x}_i), y_i)$. We assume that the classifier f is learned by minimizing the loss function and best function class f^* is selected using *empirical risk minimization*:

$$f^* \in \arg \min_{f \in \mathcal{H}} \sum_i l(f(\mathbf{x}_i), y_i).$$

3.2 Attack approach

The attack approaches considered in our experiments have the following threat model: The attacker's main goal is to cause an integrity violation to evade detection without compromising normal operation using *indiscriminate* attack whose goal is to have the test instance misclassified as any class except the true one. The attacker's capability is limited to influencing data samples at test time to evade the trained model, making it an *evasion* attack. We consider both *white-box* setting, where the attacker has complete access to both model and data distribution, and the *black-box* setting, where the attacker has no access to either data or distribution. To evaluate the security of over-parameterized models, we use the widely used white-box attack approach, which generates attacks on the known trained model parameters. PGD is a widely used evasion attack algorithm to evaluate the robustness of a model and is a good starting point for security evaluation. After evaluating on PGD, as explained in later sections, we perform experiments a on sophisticated diversified ensemble attack: AutoAttack.

3.2.1 Projected gradient descent (PGD)

Proposed by Madry et al. [33], PGD is an iterative optimization-based attack widely used for the evaluation of the models' robustness. This algorithm leverages Gradient Descent to compute a perturbation δ on a sample $\mathbf{x}$ originally labeled as y , that maximizes the output of a loss function ℓ of choice, computed on the target model f . As illustrated

in algorithm 1, at each iteration (line 2), PGD computed the gradient of the loss w.r.t the perturbation δ (line 3), which is added to the previously-computed perturbation to maximize the error. The gradient is post-processed by considering only its sign, modulated by the step size α . Then, the perturbation δ is projected with Π inside the feasible domain Γ (line 4). Such a domain defines both (i) the capability of the attacker, formalized as a bound ϵ on the p -norm of the perturbation itself resulting in $\|\delta\|_p \leq \epsilon$ and (ii) the domain constraints needed to create valid and uncorrupted adversarial examples. At the end of the iterations, the perturbation that mostly maximizes the loss is added to the original sample to produce the adversarial example (line 5).

Input : $\mathbf{x}, y$, the initial sample and its label; f , the target model; ℓ the loss function; Π_Γ , the projection operator enforcing the constraints of the feasible domain Γ ; n , the number of iterations; α , the step size

Output: $\mathbf{x}^*$, the solution found by the algorithm

```

1  $\delta_0 \leftarrow \mathbf{0}$ 
2 for  $i \in [1, n]$  do
3    $\delta_i \leftarrow \alpha \text{sign}(\nabla_{\delta_i} \ell(f(\mathbf{x} + \delta_i), y))$ 
4    $\delta_i \leftarrow \Pi_\Gamma(\mathbf{x}, \delta_i)$ 
5 return  $\mathbf{x}^* \leftarrow \mathbf{x} + \text{best}(\delta_0, \dots, \delta_n)$ 

```

Algorithm 1 Projected gradient descent

3.2.2 AutoAttack

AutoAttack is an ensemble of different typologies of attacks [34]. It leverages three white-box and one black-box attack. The three white-box attacks include Automated PGD with Carlini-Wagner loss $APGD_{CW}$, Automated PGD with Difference of Logits Ratio loss $APGD_{DLR}$, targeted Fast Adaptive Boundary (FAB) attack, and Square Attacks as the black-box attack [35]. The Automated PGD (APGD) is the standard PGD attack with a fixed number of iterations and adaptive step size. It uses Carlini-Wagner Loss and Difference of Logits Ratio loss instead of cross-entropy loss. $APGD_{CW}$ is an untargeted attack whereas $APGD_{DLR}$ is a targeted attack. Namely, it computes the adversarial example ad-hoc to have it misclassified as a specific class. If it does not succeed, it considers a different class as the target class. The third parameter-free white-box attack is the targeted version of the Fast Adaptive Boundary (FAB) attack [36]. Finally, Square Attack is an untargeted black-box attack based on random search and is not affected by gradient masking. Gradient masking is a common adversarial defense [37] against gradient-based attack algorithms that makes the gradient usually computed by these attacks uninformative, which makes the attacker unable to compute an effective adversarial example. However, thanks to the transferability property of adversarial examples [38], these defense techniques can be easily broken by computing an attack against a model owned by the attacker (surrogate). These four attacks are packaged in a single module that serves as a diversified attack and helps evaluate the reliability of the results we obtained from our primary PGD experiments.

3.3 Evaluation procedure

To evaluate the security of neural networks, we take into account different neural network architectures. These architectures are then scaled with different numbers of hidden units in their layers as done in [39]. We consider a model architecture, expanded to ten models ranging from under-parameterized to over-parameterized regime $\theta \in \{\theta_1, \theta_2, \dots, \theta_m\}$. The models are first trained on the training data. Then, for each

model θ_i , its base performance is evaluated using benign test data, i.e., where there is no perturbation ϵ_0 . Consequently, we generate the adversarial examples ad-hoc against the considered model θ_i for varying degrees of attack budgets $\epsilon \in \{\epsilon_1, \epsilon_2, \dots, \epsilon_k\}$ and for each perturbation budget we compute the *Error Rate* (namely the percentage of misclassified samples) of θ_i on them. These error rates are then used to generate the SEC curve that shows how the error rate of the model θ_i increases for a growing perturbation budget of the attacker.

3.4 How reliable are these evaluations?

In an attempt to verify the generalizability of the models, different works carried out analysis under different scenarios, ex., real-world environment, adversarial environment, noisy environment, etc. For safety- and security-sensitive tasks, robustness under an adversarial environment is of utmost importance. Various earlier works [32, 37, 40] have shown that constructing effective adversarial defense is harder than expected. Many works that propose new defenses, to evaluate the system security, use suboptimal attack methods, often leading to a false sense of security. Even the use of simple gradient-based attacks like PGD, as observed in [41, 42] can fail under certain parameter choices leading to over-estimation of robustness. As explained in the previous section, the related works (e.g., [27, 33]), have employed PGD to compare the robustness of networks with different sizes, but they have not verified if the attack was failing due to a known problem. Therefore, it is necessary to verify the reliability of the security evaluations. In this work, we verified the results obtained with the PGD attack using Indicators of Attack Failures (IoAF) presented by Pintor et. al [32], which encompasses all the failures known to date. We also used AutoAttack as an additional ensemble of attacks to measure the reliability of the results shown in Sect. 4. With over decade-long research on devising new attacks and failures to study the robustness of models, many recent works have shown that the attacks may fail to generate viable adversarial samples, for example, depending on the choice of loss function L and the target model θ , some loss functions may exhibit *obfuscated gradients* [37, 43] which prevents gradient-based attacks from generating an optimal attack even when they exist in the feasible domain. Another known issue in constructing attacks is *attack-optimization failures* which are observed at the time of running gradient-based attacks [11, 44], which may be either due to usage of suboptimal parameters or inappropriate implementation. Therefore, we use IoAF [32], which is an automated testing package aimed at detecting *known* failures within the optimization process of gradient-based attacks. We use IoAF to make sure the attacks generated on over-parameterized models do not present any failure known to date. Most model evaluations rely on the results of PGD or its weaker version: FGSM [5]. Models evaluated on single or weak attacks can run into the problem of suboptimal solution found on a local minima that maybe far from the global optimum. Therefore, for reliable evaluation of the over-parameterized models, we run a diverse attack to increase the probability to find better approximation of the global optimum. We use AutoAttack to add diversity in the evaluation. Their parameter-free, computationally cheap and user-independent parameters make it a useful tool for measuring the reliability of the security evaluations of the model (explained in detail in Sect. 3.2.2).

4 Experiments

We perform experiments with base neural network architectures that are expanded width-wise by adding neurons to its layers to move the model to the over-parameterized regime.

4.1 Experimental setup

Datasets. We conduct experiments on two widely used benchmark datasets: MNIST [45] and CIFAR-10 [46]. MNIST consists of 70,000 grayscale images of handwritten digits (0–9), each of spatial resolution 28×28 . The dataset is partitioned into 60,000 training samples and 10,000 test samples. CIFAR-10 consists of 60,000 RGB images of size 32×32 , evenly distributed across ten object categories. The standard split contains 50,000 training samples and 10,000 test samples. For both datasets, we further partition the original training set into an 80:20 train–validation split. Models are trained exclusively on the training portion, while the validation set is used for model selection and early checkpointing. For evaluation, including both clean and adversarial robustness assessment, we use a fixed subset of 1000 test samples. This choice is motivated by computational considerations: the cost of adversarial attacks scales linearly with the number of evaluation samples and significantly with model capacity.

Models. We consider three model families: a) Multi-Layer Perceptron (MLP), b) Convolutional Neural Network (CNN), and c) Residual Network (ResNet). Each architecture is systematically scaled across 10 capacity levels, yielding 10 distinct model sizes per architecture. Scaling is achieved by increasing the width (number of hidden units or channels) while preserving architectural depth and structural design principles. This controlled scaling enables us to isolate the effect of over-parameterization on both clean and adversarial performance.

Formally, let $x \in \mathcal{X} \subset \mathbb{R}^d$ denote the input and $y \in \mathcal{Y} \subset \mathbb{R}^c$ denote the output space. Each model defines a function $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^c$ parameterized by θ , where the total number of parameters varies across capacity levels. For instance, in a two-layered feedforward MLP network:

$$f(x) = h^{(2)}(h^{(1)}(x)) \quad (4)$$

where $h^{(1)} \in \mathbb{R}^{m \times d}$ and $h^{(2)} \in \mathbb{R}^{c \times m}$, and m controls the model capacity. Similarly, for CNNs and ResNets, capacity scaling increases the size of the convolutional channels while preserving the architectural topology. Across all experiments, architectural hyperparameters other than the width are held constant to ensure that the performance differences arise solely from parameter scaling.

Training Procedure. Before training, we do a hyperparameter tuning using a validation split. The selected optimization configurations are: a) for MLP and ResNet, we use Stochastic Gradient Descent (SGD) as the optimizer with 0.01 learning rate and $5e-4$ weight decay, and b) for CNN, we use ADAM as the optimizer with $1e-3$ as the learning rate and $1e-4$ as the weight decay. The models are trained for 1000 epochs. The best-performing checkpoint, defined as the model with the highest validation accuracy, is retained for evaluation. All models are trained using cross-entropy loss. The same training protocol is applied across all capacity levels to ensure comparability.

Evaluation. Evaluation is performed in a controlled setting where clean and adversarial performances are computed. **Benign Performance:** Performance is computed using

the saved checkpoints (across all ten capacity levels per architecture) for unperturbed samples. **Adversarial Performance:** Adversarial robustness is evaluated using two complementary attack frameworks: a) Projected Gradient Descent (PGD-L2) implemented via the SecML library and b) AutoAttack, a parameter-free ensemble attack. For each trained checkpoint, adversarial attacks are performed over a range of perturbation budgets ϵ . Robust accuracy is computed as the fraction of correctly classified adversarial examples at each perturbation level.

Experimentally, the evaluation protocol is as follows: (a) Load the checkpoint for a given capacity, (b) Evaluate clean accuracy on a 1000-sample test subset, (c) For each perturbation budget ϵ , generate adversarial examples and compute robust accuracy. These steps are repeated for all architecture-dataset pairs.

4.2 Experimental results

Despite the controversy over the generalizability of over-parameterized networks, we reliably observe greater robustness in largely-parameterized models than in smaller networks.

4.2.1 Base performance

From Fig. 2 we see the base performance, namely the performances on the clean (unperturbed) test data, of the three configurations considered in this work: CNN trained on MNIST, MLP trained on MNIST, and Resnet18 trained on CIFAR10, respectively. We can observe that:

- The classifier's performance changes when we increase the number of network parameters;
- None of the performance on the test data worsens as would have been expected from the classical theory of overfitting for largely parameterized models. Instead, the error rates continue to decrease for over-parameterized networks, supporting the generalizability claims.

These are well-known results, but they serve to us as a baseline. It's worth noting that the focus of the work is not on acquiring the best possible classification performance on the base model but instead how the robustness of the model varies as we increase the size of the model.

4.2.2 Model robustness

Figure 3 shows the security evaluation curves (SEC) for networks attacked with a PGD attack with an l_2 -norm constraint across an increasing number of parameters

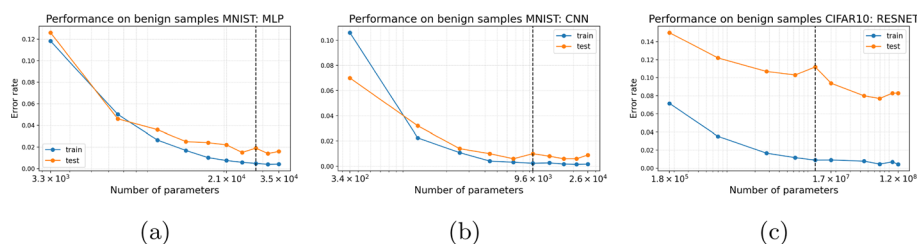


Fig. 2 Base performance. Evaluation of models on clean samples with respect to increasing model size: **a** MLP trained on MNIST, **b** CNN trained on MNIST, and **c** ResNet model trained on CIFAR10

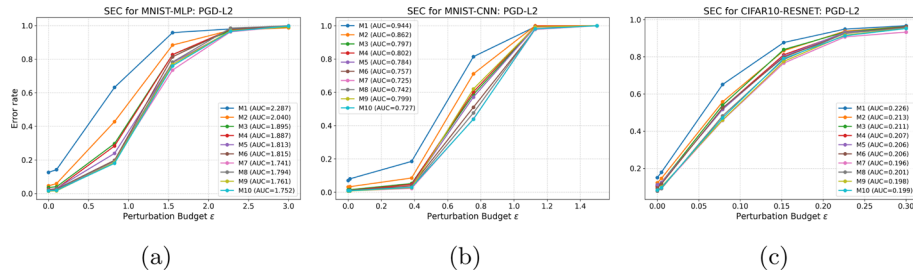


Fig. 3 SEC plots. Evaluation of varying-sized models under **PGD l_2** attack: **(a)** MLP trained on MNIST, **(b)** CNN trained on MNIST, and **(c)** ResNet trained on CIFAR10. Each curve represents the security evaluation curve with the area under the curve specified in the legend

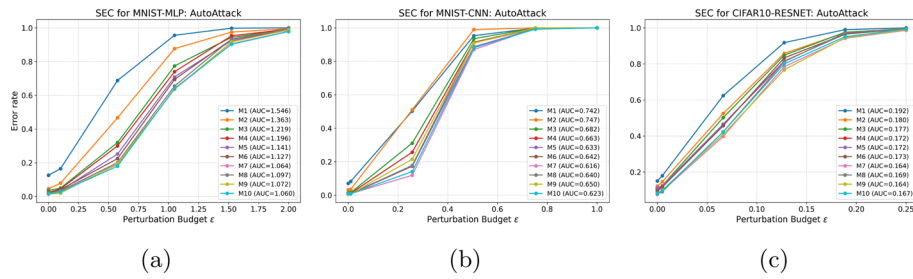


Fig. 4 SEC plots. Evaluation of varying-sized models under **AutoAttack**: **(a)** MLP trained on MNIST, **(b)** CNN trained on MNIST, and **(c)** ResNet trained on CIFAR10. Each curve represents the security evaluation curve with the area under the curve specified in the legend

and various degrees of perturbation. While it is understood that the error rate would increase with increasing degrees of perturbations, it is interesting to note that *error rates for the over-parameterized model are significantly lower than those of the models with fewer parameters*. We see similar trends for all dataset-architecture pairs. This is an important observation, as it suggests that over-parameterized models are not only generalizable but also robust to adversarial samples. However, this outcome may be due to a failure of the PGD attack used for the evaluation.

4.2.3 Reliability of model robustness

This subsection aims to verify the reliability of the analysis made with PGD.

Evaluation of robustness with a standard benchmark. The results in Fig. 4 include all dataset-architecture configurations tested employing AutoAttack. Results:

- Even though one could train a close-to-perfect model on the unperturbed test set using smaller networks since there isn't a significant decline in the benign data performance, over-parameterized models significantly help with the robustness.
- The trends obtained with AutoAttack follow that of PGD from Sect. 4.2.2; moreover, we have verified the affidability of the Sect. 4.2.2 plots performing IoAF and we have observed no triggers for attack failures for any of the six indicators. These results verify that the analysis performed in this paper is conducted employing all the knowledge about the security of machine learning models available so far, providing reliability to the observed higher robustness of overparameterized networks.

Lipschitz Evaluation, For completeness, we conduct a further robustness analysis using the Lipschitz constant as a metric. Recall that the Lipschitz constant of a neural network

is directly related to its robustness: the certifiable robustness radius is inversely proportional to the Lipschitz constant, with the top-2 output margin serving as the proportionality numerator [47]. Consequently, a larger Lipschitz constant generally yields a smaller certifiable robustness radius, modulated by the margin between the top-2 output components. From a generalization perspective, [48] established that a large Lipschitz constant is necessary for interpolating below the noise level in the non-overparameterized regime, linking smoothness to generalization under overparameterization.

To quantify this, we measure the Lipschitz constant of models across varying parameter counts with respect to the Euclidean norm. Given a composed model $f = f^{(L)} \circ \dots \circ f^{(1)}$, we compute both an upper bound $C_u(f)$ and a lower bound estimate $C_l(f)$ of the Lipschitz constant. The upper bound exploits the sub-multiplicativity of the Lipschitz constant under composition:

$$C_u(f) = \prod_i C_u(f^{(i)}), \quad (5)$$

where the Lipschitz constant of convolutional layers is estimated via the singular value method of [49]; residual layers of the form $f(x) = x + h(x)$ are bounded by $C_u(f) = 1 + C_u(h)$ via the triangle inequality; and for fully connected layers with weight matrix W , the Lipschitz constant equals the spectral norm $\|W\|_2$. Finally observe that ReLU activations are 1-Lipschitz, being $\text{Jac}(\text{ReLU})$ diagonal with only 0 or 1 values.

For the lower bound, we exploit the fact that the Lipschitz constant equals the supremum of the spectral norm of the Jacobian over the input space. This quantity is lower-bounded by sampling points x from a dataset $\mathcal{D}$ and augmenting with Gaussian perturbations $\varepsilon \sim \mathcal{N}(0, \sigma)$, yielding

$$C_l(f, x) = \max_{\varepsilon \sim \mathcal{N}(0, \sigma)} \max_{i \neq j} \|\nabla(f_i - f_j)(x + \varepsilon)\|_2, \quad (6)$$

from which $C_l(f) = \max_{x \in \mathcal{D}} C_l(f, x)$. In our experiments, we evaluate over the full test set $\mathcal{D}$, drawing 100 perturbation samples per input with $\sigma = 0.125$.

Results are reported in Appendix C. Consistent with prior literature, the upper bound C_u proves overly loose, exhibiting no meaningful decrease as model size increases. The lower bound C_l , however, aligns well with the robust accuracy results in Sect. 4.2.2 for resnet and mlp models: where a higher robust accuracy aligned lower estimated Lipschitz constants, corroborating the theoretical connection between smoothness and robustness, and hence with over parameterization.

5 Conclusions and future works

In this work, we empirically study how robust over-parameterized models are in adversarial environments as compared to their under-parameterized counterparts. These impressive generalizable properties of large models defy the widely accepted understanding of the model selection method that uses the Empirical Risk Minimization (ERM) principle, which is biased to select the smallest network that can fit the data well, and is often on the left side of the second descent of the double descent curve. From the classical statistical notion of bias-variance tradeoff, the optimal model is neither too large nor too small to avoid overfitting and underfitting, respectively. Yet the performance of large models outperforms that of smaller models, making their usage ubiquitous. However,

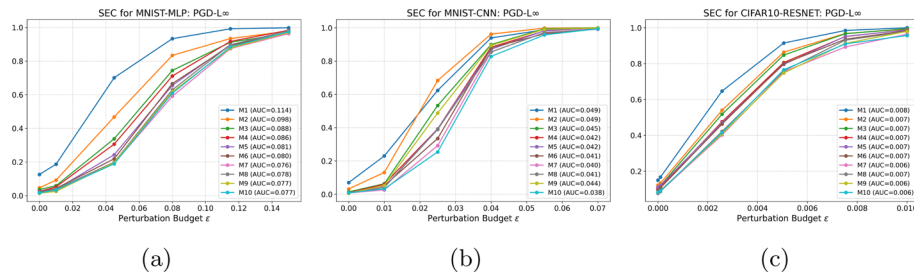


Fig. 5 SEC Plots. Evaluation of varying-sized models under PGD l_∞ attack: **a** MNIST trained on MLP, **b** MNIST trained on CNN, and **c** CIFAR10 trained on ResNet. Each curve represents the security evaluation curve with the area under the curve specified in the legend

when it comes to the security of the large models, the literature has contradictory claims in favor and against the robustness in an adversarial environment. This claims are made without verifying the reliability of the attacks and, for this reason, they are unreliable. Our results conducted with an attack of which we have verified that does not present any known failures, show that over-parameterized models not only generalize well but are also more robust to carefully crafted adversaries in their environment than their under-parameterized counterparts. We provide the empirical evidence to show that *even when increasing the number of parameters is no more beneficial for the accuracy on clean data, it is beneficial for the robustness against adversarial examples*. Future directions of this work may include evaluating over-parameterized models against privacy attacks. It would also be interesting to investigate if it is possible to obtain smaller networks with accuracy and robustness comparable to those of over-parameterized networks to avoid computational burden.

6 Expected impact on future work

While some work about the robustness of over-parameterized deep neural networks was present in the literature, their claim were contradictory, and their analysis were unreliable because the effectiveness of the performed attacks was not verified. The analysis presented in this work is verified with ad-hoc tools to analyze its reliability and shows that over-parameterized networks are more robust. We believe the empirical evidence presented in this manuscript serves as a starting point for theoretical work explaining why over-parametrized networks are more robust. Moreover, we believe that highlighting the reliability of robustness analysis for over-parameterized networks and the incoherence of the previously proposed analysis will lead to more reliable future work on this topic.

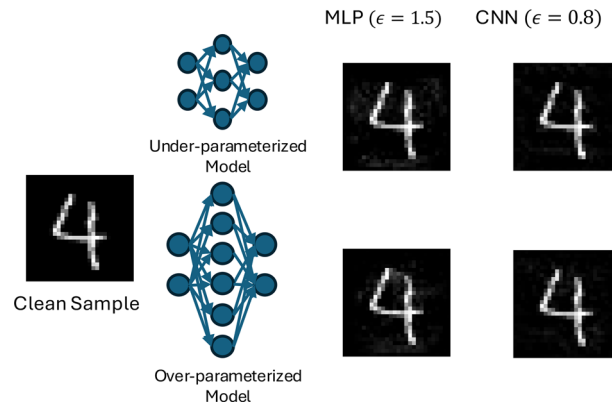


Fig. 6 Adversarial examples. The visual representation of PGD injected adversarial noise to MNIST digit 4 for under-parameterized (smallest) and over-parameterized (largest) models for corresponding architectures: MLP ($\epsilon = 1.5$) and CNN ($\epsilon = 0.8$)

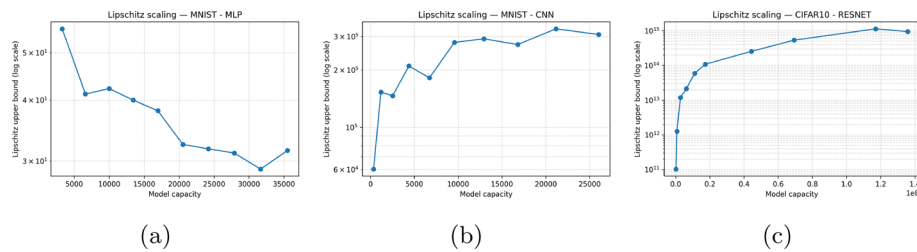


Fig. 7 Upper Lipschitz plots. The upper bound of Lipschitz constant for: (a) MLP, (b) CNN, and (c) ResNet. Each curve represents the Lipschitz value across different model sizes

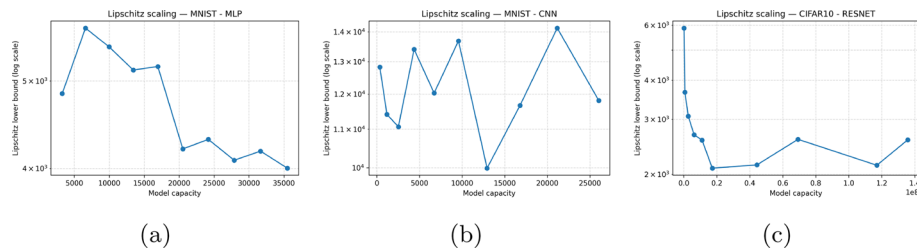


Fig. 8 Lower Lipschitz plots. The lower bound of the Lipschitz constant for: (a) MLP, (b) CNN, and (c) ResNet. Each curve represents the Lipschitz value across different model sizes

Security evaluation curves for PGD l_∞ Norm

In this section, we support our claims by presenting the security evaluation curves for l_∞ attacks on all three architectures: MLP, CNN, and ResNet, by scaling the size of the models (Fig. 5).

Visualization of adversarial examples

In Fig. 6, we have visualized two examples of adversarial examples generated for an under-parameterized and an over-parameterized model for MLP and CNN.

Bound Lipschitz

Overparameterization is a necessary but not sufficient condition for small Lipschitz interpolation (Figs. 7 and 8).

Author contributions

S.G. wrote the manuscript draft, performed the experiments, and made some updates to the code written by L.D. Z.C. made the initial exploratory experiments. L.D. wrote the code for the experiments currently reported in the paper. F.B. analysed the Lipschitz constant of the models. A.D. supervised Z.C. and S.G. B.B., X.F., and F.R. acquired funding. All authors reviewed the manuscript.

Funding

This work has been carried out while Srishti Gupta was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with the University of Cagliari. This work was also partially supported by project FAIR (PE00000013) under the NRRP MUR program funded by the EU – NGEU (CUP: J23C24000090007).

Data availability

All data supporting the findings of this study are publicly available. The dataset MNIST is available here: <https://www.kaggle.com/datasets/hojjatk/mnist-dataset> and the CIFAR10 dataset is available here: <https://www.cs.toronto.edu/~kriz/cifar.html>

Declarations

Ethics approval and consent to participate

Not Applicable.

Consent to publish

Not Applicable.

Competing interests

The authors declare no conflict of interest.

Received: 28 November 2025 / Accepted: 5 June 2026

Published online: 25 June 2026

References

1. Belkin M, Hsu D, Ma S, Mandal S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc Natl Acad Sci*. 2019;116(32):15849–54. <https://doi.org/10.1073/pnas.1903070116>.
2. Neyshabur B, Tomioka R, Srebro N. Search of the real inductive bias: on the role of implicit regularization in deep learning. 2015.
3. Novak R, Bahri Y, Abolafia DA, Pennington J, Sohl-Dickstein J. Sensitivity and generalization in neural networks: an empirical study. In: International conference on learning representations (2018). <https://openreview.net/forum?id=HJC2SzZCW>
4. Biggio B, Corona I, Maiorca D, Nelson B, Srndic N, Laskov P, Giacinto G, Roli F. Evasion attacks against machine learning at test time. *CoRR* **abs/1708.06131** (2017)
5. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: International conference on learning representations. 2015.
6. Nguyen A, Yosinski J, Clune J. Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 427–436 (2015). <https://doi.org/10.1109/CVPR.2015.7298640>
7. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. In: International conference on learning representations (2014). <http://arxiv.org/abs/1312.6199>
8. Moosavi-Dezfooli S-M, Fawzi A, Frossard P. Deepfool: a simple and accurate method to fool deep neural networks. In: IEEE conference computer vision and pattern recognition (CVPR). 2016. p. 2574–82.
9. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning requires rethinking generalization. 2017.
10. Belkin M, Ma S, Mandal S. To understand deep learning we need to understand kernel learning (2018)
11. Tramer F, Carlini N, Brendel W, Madry A. On adaptive attacks to adversarial example defenses. 2020.
12. Popovic N, Paudel DP, Probst T, Gool LV. Gradient obfuscation checklist test gives a false sense of security. 2022.
13. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, New York, NY, USA (2009). <https://doi.org/10.1007/978-0-387-84858-7>
14. Neal B, Mittal S, Baratin A, Tanti V, Scicluna M, Lacoste-Julien S, et al. A modern take on the bias-variance tradeoff in neural networks. 2019.
15. Yang Z, Yu Y, You C, Steinhardt J, Ma Y. Rethinking bias-variance trade-off for generalization of neural networks. 2020.
16. Nakkiran P, Kaplan G, Bansal Y, Yang T, Barak B, Sutskever I. Deep double descent: where bigger models and more data hurt. 2019.
17. Bartlett PL, Montanari A, Rakhlin A. Deep learning: a statistical viewpoint. 2021.
18. Hastie T, Montanari A, Rosset S, Tibshirani RJ. Surprises in high-dimensional ridgeless least squares interpolation. 2020.
19. Wu B, Chen J, Cai D, He X, Gu Q. Do wider neural networks really help adversarial robustness? In: *NeurIPS*. 2021. p. 7054–67.

20. Rice L, Wong E, Kolter JZ. Overfitting in adversarially robust deep learning. In: Proceedings of the 37th international conference on machine learning. ICMML'20. JMLR.org. Austria; 2020.
21. Huang H, Wang Y, Erfani SM, Gu Q, Bailey J, Ma X. Exploring architectural ingredients of adversarially robust deep neural networks. *NeurIPS*; 2021. p. 5545–59.
22. Gao R, Cai T, Li H, Hsieh C, Wang L, Lee JD. Convergence of adversarial training in overparametrized neural networks. In: *NeurIPS*. 2019. p. 13009–20.
23. Hassani H, Javanmard A. The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression. *Ann Stat*. 2024;52(2):441–65. <https://doi.org/10.1214/24-AOS2353>.
24. Fu L, Chen T. Sensitivity analysis for input vector in multilayer feedforward neural networks. In: IEEE international conference on neural networks, pp. 215–2181 (1993). <https://doi.org/10.1109/ICNN.1993.298559>
25. Sagawa S, Raghunathan A, Koh PW, Liang P. An investigation of why overparameterization exacerbates spurious correlations. 2020.
26. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: International conference on learning representations (2018). <https://openreview.net/forum?id=rjzIBfZAb>
27. Borji A. Overparameterization improves robustness against adversarial attacks: a replication study. 2022.
28. Li B, Jin J, Zhong H, Hopcroft JE, Wang L. Why robust generalization in deep learning is difficult: perspective of expressive power. *NeurIPS*; 2022.
29. Zhu Z, Liu F, Chrysos G, Cevher V. Robustness in deep learning: the good (width), the bad (depth), and the ugly (initialization). *NeurIPS*; 2022.
30. Ribeiro AH, Schön TB. Overparameterized linear regression under adversarial attacks. *IEEE Trans Signal Process*. 2023;71:601–14. <https://doi.org/10.1109/TSP.2023.3246228>.
31. Carlini N, Wagner DA, Thuraisingham BM, Biggio B, Freeman DM, Miller B. Adversarial examples are not easily detected: bypassing ten detection methods. In: Sinha A, editor. 10th ACM Workshop on Artificial Intelligence and Security. *AISeC '17*. New York, NY, USA: ACM; 2017. p. 3–14.
32. Pintor M, Demetrio L, Sotgiu A, Demontis A, Carlini N, Biggio B, et al. Indicators of attack failure: debugging and improving optimization of adversarial examples. 2022.
33. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. 2018. p. ICLR.
34. Croce F, Hein M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International conference on machine learning. 2020. p. 2206–16 (**PMLR**).
35. Andriushchenko M, Croce F, Flammarion N, Hein M. Square Attack: a query-efficient black-box adversarial attack via random search. 2020.
36. Croce F, Hein M. Minimally distorted adversarial examples with a fast adaptive boundary attack. 2020.
37. Athalye A, Carlini N, Wagner DA. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In: ICMML. JMLR workshop and conference proceedings, vol. 80, pp. 274–283. JMLR.org, Sweden (2018)
38. Demontis A, Melis M, Pintor M, Jagielski M, Biggio B, Oprea A, et al. Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. In: 28th USENIX Security Symposium (USENIX Security 19). USENIX Association. USA; 2019.
39. Neyshabur B, Li Z, Bhojanapalli S, LeCun Y, Srebro N. Towards understanding the role of over-parametrization in generalization of neural networks (2018)
40. Carlini N, Wagner D. Adversarial examples are not easily detected: bypassing ten detection methods. 2017.
41. Mosbach M, Andriushchenko M, Trost T, Hein M, Klakow D. Logit pairing methods can fool gradient-based attacks. 2019.
42. Croce F, Rauber J, Hein M. Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks. 2019.
43. Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (SP), pp. 39–57 (2017). <https://doi.org/10.1109/SP.2017.49>
44. Athalye A, Engstrom L, Ilyas A, Kwok K. Synthesizing robust adversarial examples. 2018. p. ICLR.
45. LeCun Y, Cortes C. The mnist database of handwritten digits. (2005). <https://api.semanticscholar.org/CorpusID:60282629>
46. Krizhevsky A. Learning multiple layers of features from tiny images. (2009). <https://api.semanticscholar.org/CorpusID:18268744>
47. Hein M, Andriushchenko M. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in neural information processing systems*. 2017. p. 30.
48. Bubeck S, Sellke M. A Universal Law of Robustness via Isoperimetry. 2021.
49. Sedghi H, Gupta V, Long PM. The singular values of convolutional layers. In: International Conference on Learning Representations (2019). <https://openreview.net/forum?id=rJevYoA9Fm>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.