Probabilistic Global Robustness Verification of Arbitrary Supervised Machine Learning Models

1st Max-Lion Schumacher
Fraunhofer IPA
Stuttgart, Germany
max-lion.schumacher@ipa.fraunhofer.de

2nd Marco F. Huber
Fraunhofer IPA and
Institute of Industrial Manufacturing and Management IFF,
University of Stuttgart
Stuttgart, Germany
marco.huber@ieee.org

***Abstract*—Many works have been devoted to evaluating the robustness of a classifier in the neighborhood of single points of input data. Recently, in particular, probabilistic settings have been considered, where robustness is defined in terms of random perturbations of input data. In this paper, we consider robustness on the entire input domain as opposed to single points of input. For the first time, we provide formal guarantees on the probability of robustness, given a random input and a random perturbation, based only on sampling or in combination with existing pointwise methods. We prove that the error becomes arbitrarily small for enough input data. This is applicable to any classification or regression model and any random input perturbation. We then illustrate the resulting bounds and compare them against the state of the art for models trained on the MNIST, California Housing, and ImageNet datasets.**

***Index Terms*—machine learning, regression, classification, neural networks, global robustness, verification, statistical testing**

I. Introduction

Neural Networks have demonstrated unprecedented performance for various tasks [1]–[3], yet they are susceptible to adversarial perturbations [4], [5]. Several approaches for assessing and mitigating this risk have been proposed [6]–[9]. Previous work has, to a large degree, focused on formal guarantees for worst-case robustness [10], [11]. In many cases we are dealing with random input noise, generated by noisy sensors for example and a worst-case approach is too strict. Proving the absence of adversarial examples in large scale applications is usually not feasible. It is, however, often sufficient to have a guarantee, that the probability of non-robustness is below a certain threshold. Therefore, probabilistic notions of robustness are better suited in these scenarios and corresponding methods have been developed [12], [13]. All of the aforementioned works, however, focus on assessing robustness for single, specific inputs. When faced with the problem of certifying safety of a neural network based method, e.g., an object classifier in a self-driving car, robustness has to be assessed on the entire set of possible inputs. Few works have considered this notion of global robustness [14]. For the first time, we provide formal guarantees on the probability of global robustness. We propose two methods for doing so: the first one being based on sampling and the second one using existing local methods and elevating provided guarantees to a global level. The main ingredient for this is a statistical test. Our method can be applied to any classification or regression model. We demonstrate experimentally that compared to the state of the art, our method provides greatly improved guarantees holding with high probability. Furthermore, we illustrate the strenghts of our methods by applying them to a ResNet classifying ImageNet in a setting of perturbations such as rain, that are of real-world significance.

To summarize, our key contributions are

- We give two novel formal definitions of global probabilistic robustness that can be applied to arbitrary supervised machine learning models, arbitrary random perturbations and, in case of the second definition, to any method providing pointwise guarantees.
- For the first time, we provide methods to derive formal guarantees on the probability of global robustness.
- These methods can be applied to arbitrary supervised machine learning problems and thus, hold for both classification and regression.
- They provide greatly improved guarantees compared to the state of the art, holding with high probability.
- The error of the bounds converges to zero as the sample size goes to infinity.

This paper is structured as follows: In Section II, we give an overview of the state of the art in related areas of work. In Section III, we give a definition for global probabilistic robustness, derive a method based only on sampling and prove the resulting bounds, as well as convergence of the error to zero. In Section IV, we give another definition of global probabilistic robustness that is suited to include existing pointwise methods. We derive a method to provide robustness bounds in this setting and again prove the resulting bounds, as well as convergence of the error to zero. In Section V, we focus on whether the binomial distribution, which results from the methods introduced in Section III and IV, is computable for large parameters. In Section VI, we compare our methods to the state of the art and apply them to an example that is relevant to real-world applications. This paper closes with a discussion and an outlook on future work in Section VII.

This work was partially supported by the Fraunhofer-Gesellschaft within the project ML4Safety and the Federal Ministry for Economic Affairs and Climate Action within the project veoPipe (Grant no. 19A21040B).

II. Related Work

a) Adversarial attacks beyond l^p bounds: Initially, adversarial examples have been considered as l^p-bounded attacks [9]). Among the most notable verification tools for this setting are auto_LiRPA, Marabou and ERAN [15]–[17]. Subsequently, various other types of powerful adversarial attacks have been created. These include semantic manipulations such as rotation and translation [18], [19], color alteration [20], renderer-based light and geometric transformation [21], parametric attribute alteration [22], and 3D scene parameters such as camera positioning, sunlight location, global translation, and rotation [23]. These manifold types of perturbations allow for a more comprehensive understanding and assessment of the notion of robustness against input perturbation.

b) Probabilistic robustness: Recently, some works have taken into account the fact that in many cases probabilistic notions of robustness arise more naturally than worst-case notions, e.g., when dealing with noisy sensor data, and they are important to consider in case of safety critical applications [24]. Verification methods leveraging this notion have been developed in [12], [13], [25]. In [26] a sophisticated sampling method is combined with statistical tests to provide robustness guarantees. These are all strictly pointwise methods, meaning that they give guarantees only for single points of input data. Our goal, however, is to consider robustness on the entire input domain. In particular, in Section IV we will explore how pointwise methods, being probabilistic or otherwise, can be used to derive global bounds.

c) Global robustness: For a comprehensive robustness analysis of a machine learning model it is important to shift the perspective beyond local robustness. Few works, however, have considered notions of global robustness. One of those is [27]. In this work, formal guarantees are derived based on 'repairing' the model for non-robust areas. However, since the definition of robustness is similar to Lipschitz continuity, the approach is not well suited for classification models. The reason is that a model with a small Lipschitz constant can still not be robust, e.g., for input points for which the model is not very certain, i.e., there are small differences between the predicted probabilities of class labels. Probabilistic settings cannot be handled with this approach. Also the model has to be manipulated, which may lead to a decrease in accuracy. Another approach to verify global robustness works by creating twin networks [28], [29]. It includes mixed-integer linear programming and over-approximation. This method suffers from the same limitations, using a definition similar to Lipschitz continuity. The same goes for [30]–[32]. The definition of global robustness given in [14] even considers a probabilistic setting. Robustness is described in terms of risk measures. While being easy to evaluate, it is much harder to interpret any guarantee that might be derived for a risk measure than an actual probability of robustness. Our work provides two definitions of global robustness that are well suited for classification and regression models. It captures arbitrary randomness of the input and input perturbations. Further, it requires no manipulation or modification of the evaluated model.

III. Formal Guarantees based on Sampling

To begin, we define our notion of global robustness. We consider the general setting of a machine learning model $f : \mathcal{X} \to \mathcal{Y}$. In the case of f being, for example, a classifier, $\mathcal{Y}$ might be a finite set. For regression, $\mathcal{Y}$ might be equal to $\mathbb{R}^n$. Let the input $X : \Omega \to \mathcal{X}$ of the model be a random variable defined on the probability space $(\Omega, \mathcal{F}, D)$, so the probability of the input lying in a set $M \subset \mathcal{X}$ is $D(X \in M)$. We consider a random perturbation function $T : \mathcal{X} \to \mathcal{X}$ defined on the probability space $(\mathcal{X}, \mathcal{G}, \mathcal{T})$, so the perturbed input is given by $T(X)$. For example, we could add random noise δ to the input, yielding $T(X) = X + \delta$. Or, we could have a random brightness adjustment $T(X)$ given that the input X is an image, or any other random perturbation. Let $P := D \otimes \mathcal{T}$ denote the product measure of D and $\mathcal{T}$.

Definition 1. *A classifier model f is said to be globally robust with probability ϵ if the following bound is satisfied:*

$$P(f(X) \neq f(T(X))) \leq 1 - \epsilon \ .$$

The idea behind this definition is that we want to bound the probability that the predicted class label changes due to input perturbation, while both the input and the perturbation are random. This is often the case in applications, e.g., when dealing with noisy sensor data.

For regression, the property of the above definition becomes

$$P(\|f(X) - f(T(X))\| > c) \leq 1 - \epsilon$$

for a suitable norm and positive constant c. In the following, we focus on the case of f being a classifier. All of the results, however, apply to regression models as well.

For a given model f the goal is to find a real number ϵ as large as possible so that f is globally robust with probability ϵ. Therefore, we consider a random iid sample $X_1, \ldots, X_n$, $T_1, \ldots, T_n$ of input data and perturbations, respectively. Denote $p := P(f(X) \neq f(T(X))$ and the indicator function

$$1_M(x) := \begin{cases} 1 & x \in M, \\ 0 & \text{otherwise}, \end{cases}$$

for any x, M. We then have

$$1_{\{f(X_i) \neq f(T_i(X_i))\}} \sim \mathrm{Ber}(p)$$

for all $i \in \{1, \ldots, n\}$, where $\mathrm{Ber}(p)$ denotes the Bernoulli distribution with parameter p. These random variables are independent, as X_i and T_i are considered iid. It follows that

$$S := 1_{\{f(X_1) \neq f(T_1(X_1))\}} + \ldots + 1_{\{f(X_n) \neq f(T_n(X_n))\}} \sim \mathrm{Bin}(n, p) \ .$$

Now the idea is to use this random variable S to perform a statistical test on the parameter p. The hypotheses are

$$H_0 : p \geq p_0$$
$$H_A : p < p_0$$

Algorithm 1: Sampling method providing a global robustness bound

Input: Classifier f, set of input samples input_data, random perturbation function pert, significance level α, maximal distance of output to optimal solution acc

Output: Optimal bound for probability of global robustness $p*$

```
pred ← f(input_data)                    // compute predictions for original sample
pred_pert ← f(pert(input_data))         // compute predictions for perturbed sample
s ← how_many_differ(pred, pred_pert)    // Count number of perturbed samples where classification changes
p* ← Bisection(s, α, acc, input_data)   // Call Alg. 2
return p*  // optimal bound approximation
```

for a given p_0. Formally this means we split the set $\mathcal{P}$ of possible probability measures on $\Omega \times \mathcal{X}$ in two sets

$$\mathcal{P}_0 = \{Q \in \mathcal{P} \mid Q(f(X) \neq f(T(X))) \geq p_0\} \ ,$$
$$\mathcal{P}_A = \{Q \in \mathcal{P} \mid Q(f(X) \neq f(T(X))) < p_0\} \ .$$

The binomial distribution $\mathrm{Bin}(n, p)$ describes the probability of a certain number of 'successes' when performing the same experiment independently n times, each having a probability p of 'success'. So when we observe a low enough number of 'successes', we reject the Hypothesis $H_0 : p \geq p_0$. In our case 'success' means misclassification, i.e., $f(X) \neq f(T(X))$. Hence, we choose a set of the form $K = \{0, 1, 2, \ldots, k\}$, so that H_0 is rejected if $S \in K$. Denoting the test by ϕ, we formalize this as $\phi = 1_K(S)$.

The probability of rejecting H_0, even though it is true, is naturally bounded by

$$\begin{aligned}\sup_{P \in \mathcal{P}_0} P(\phi(S) = 1) &= \sup_{P \in \mathcal{P}_0} P(S \leq k) \\ &= \sup_{p \geq p_0} \mathrm{Bin}(n, p)([0, k]) \\ &= \mathrm{Bin}(n, p_0)([0, k]) \ .\end{aligned}$$

The last equality follows from the fact that if the probability of 'success' takes its minimum at p_0, the probability of having at most k 'successes' is maximized.

The p-value ψ of an observation s of S is therefore

$$\psi = \sup_{P \in \mathcal{P}_0} P(S \leq s) = \mathrm{Bin}(n, p_0)([0, s]) \ .$$

In applications, we might want to choose a constraint of the form $\psi \leq \alpha$, with α being a user-defined significance level, and determine the smallest p_0, for which H_A can be accepted

Algorithm 2: Bisection

Input: Number of perturbed samples s, set of input samples input_data, significance level α, maximal distance of output to optimal solution acc

Output: Optimal probability value approximation p

```
lower ← 0      // lower bound of bisection
upper ← 1      // upper bound of bisection
n ← length(input_data)          // input size
while upper − lower > acc do
    m ← (lower + upper)/2       // compute the midpoint
    if binomial_cdf(s, n, m) > α then
        lower ← m   // optimal p value lies in [m, upper]
    else
        upper ← m   // optimal p value lies in [lower, m]
    end
end
return (lower + upper)/2
```

by this method, given an observation s. Formally, this can be expressed as constrained optimization problem

$$\begin{aligned}&\min p_0 \\ &\text{s.t. } \psi \leq \alpha \ .\end{aligned}$$

Since ψ is monotonously decreasing as a function of p_0, for a given s, this optimization problem can be easily solved by bisection. Denote its solution p^*.

Theorem 1. *In the above setting f is globally robust with probability p^*, i.e., the inequality*

$$P(f(X) \neq f(T(X))) < p^*$$

holds with a false positive error bound $P(S \leq s) \leq \alpha$ in any case where $p \geq p^*$.

This theorem is one of the two main results of this work. It means that we can accept

$$P(f(X) \neq f(T(X))) < p^* \tag{1}$$

and the probability of being wrong is at most α. Or put in other words, we accept the hypothesis of the inequality (1) being true by means of a statistical test with significance level α. The method for providing a robustness guarantee according to Theorem 1 is summed up by Algorithm 1.

We conclude this section by showing convergence of the result of our method to the true probability almost surely.

Theorem 2. *Let p_n^* be the result of Algorithm 1 for a sample size of n. Then we have*

$$\lim_{n \to \infty} p_n^* = P(f(X) \neq f(T(X)))$$

almost surely.

Proof. Let $(x_j)_{j\in\mathbf{N}}$ and $(t_j)_{j\in\mathbf{N}}$ be a sequence of iid inputs and perturbations, respectively, and

$$s_n := 1_{\{f(x_1)\neq f(t_1(x_1))\}} + \cdots + 1_{\{f(x_n)\neq f(t_n(x_n))\}} \ .$$

According to the law of large numbers, we have $\frac{s_n}{n} \to p = P(f(x_1) \neq f(t_1(x_1))$ almost surely. Now consider a random variable $Y \sim \mathrm{Bin}(n, q)$ and denote σ the standard deviation of a Bernoulli distribution with parameter q. A binomial test for the null hypothesis $H_0 : p \geq q$ performed on the observation s_n for an arbitrary $q > 0$ has the p-value

$$\begin{aligned} P(Y \leq s_n) &= P\left(\frac{Y - nq}{\sigma\sqrt{n}} \leq \frac{s_n - nq}{\sigma\sqrt{n}}\right) \\ &= P\Bigg(\frac{Y - nq}{\sigma\sqrt{n}} \leq \underbrace{\sqrt{n}\frac{\frac{s_n}{n} - q}{\sqrt{q(1-q)}}}_{=:\ v_n}\Bigg) \ . \end{aligned}$$

According to the uniform convergence of the central limit theorem we have

$$\left| P\left(\frac{Y - nq}{\sigma\sqrt{n}} \leq v_n\right) - \underbrace{\Phi(v_n)}_{=:\ \phi_n} \right| \to 0 \ ,$$

where Φ is the cumulative distribution function of the standard normal distribution. Now, since $\frac{s_n}{n} \to p$, we can see that for every $q > p$ we have $\phi_n \to 0$. This means for a parameter q arbitrarily close to p with a large enough input sample, the binomial test for q has an arbitrarily low p-value for s_n. Since Algorithm 1 finds the optimal parameter p_n^* for a given confidence level, it follows that $p_n^* \to p$. □

IV. Two-Stage Verification

In this section, we are dealing with the question how to use existing methods providing pointwise guarantees to elevate these to a global level. Examples for pointwise methods are [12], [13]. We consider a general setting, where the nature of the guarantees provided by the underlying pointwise methods can be deterministic or probabilistic. Let the model $f : \mathcal{X} \to \mathcal{Y}$ and it's input X be defined as in Section III. In order to specify our notion of global robustness in this setting, we require the following definition.

Definition 2. *A robustness score is a function $r : \mathcal{X} \to \mathbb{R}$.*

A robustness score r can be the output of a method providing rather vague estimates of robustness or formal guarantees of deterministic or probabilistic nature. In any case r assesses robustness only on single inputs $x \in \mathcal{X}$. For a better understanding, we give two examples. They involve the concept of soundness. In the context of verification a method is called sound, if the method certifying a property, e.g., robustness, formally implies that the property holds.

Example 1. *Consider a deterministic pointwise method for assessing worst-case robustness. Denote by r it's output. Soundness can be expressed by*

$$r(x) \geq \sup_{y\in N(x)} 1_{\{f(x)\neq f(y)\}} \ ,$$

for any input $x \in \mathcal{X}$ and its neighborhood $N(x)$. Here, $r(x) = 0$ implies that there is no adversarial example in $N(x)$.

Example 2. *In the case of a method that provides probabilistic bounds, we could define soundness as*

$$r(x) \geq \mathcal{T}(f(x) \neq f(T(x))) \ ,$$

for all $x \in \mathcal{X}$, T being a random perturbation, distributed according to $\mathcal{T}$.

In some of the experiments in Section VI we will see methods satisfying the conditions of these examples. In this setting, our definition of robustness is the following.

Definition 3. *A model f is said to be globally robust with probability ϵ, with respect to the robustness score r and deviation level c, if the following bound is satisfied:*

$$D(r(f(X)) > c) < 1 - \epsilon$$

Let a robustness score r and a critical deviation level c be given. Now we aim to find an ϵ, as large as possible, so that f is globally robust with probability ϵ, with respect to the robustness score r and deviation level c. We start by considering

$$\mathrm{E}\left[1_{\{r(f(X))>c\}}\right] = D(r(f(X)) > c) =: p \ .$$

Let $X_1, \ldots, X_n$ be an iid input sample. Since we have

$$S := 1_{\{r(f(X_1))>c\}} + \cdots + 1_{\{r(f(X_n))>c\}} \sim \mathrm{Bin}(n, p) \ ,$$

as in Section III, we can use a binomial test. Analogously, for a given observation s of S and a p-value boundary α we get an optimal parameter p^* such that we can statistically prove

$$D(r(f(X)) > c) = p < p^*$$

with a test ϕ having a false positive error bound

$$\sup_{D:p\geq p^*} D(\phi(S) = 1) < \alpha \ . \tag{2}$$

To conclude, the following theorem states the second main result of this work.

Theorem 3. *In the above setting the model f is globally robust with probability p^*, with respect to r and c with a false positive error bound given by Equation* (2)*.*

In case of the deterministic Example 1 with $c = 0$ the resulting bound translates to

$$p = D(\exists y \in N(X) : f(X) \neq f(y)) < p^*.$$

In the probabilistic case of Example 2, we obtain

$$p = D(\mathcal{T}(f(X) \neq f(T(X))) < c) < p^* \ .$$

Again, we sum up the introduced two-stage method in Algorithm 3, which is similar to Algorithm 1 with the condition $f(X) \neq f(T(X))$ being replaced by $r(f(X)) > c$. Quite similarly to the proof of Theorem 2, one can show that the output converges to the true probability p.

Algorithm 3: Two-stage method providing a global robustness bound

Input: Classifier f, set of input samples input_data, local robustness assessment method r, local robustness threshold c, significance level α, maximal distance of output to optimal solution acc

Output: Optimal bound for probability of global robustness p^*

```
s ← 0
for d in input_data do
    if r(d) > c then
        s ← s + 1      // count samples with robustness score above threshold
    end
end
p* ← Bisection(s, α, acc, input_data)      // Call Alg. 2
return p*  // optimal bound approximation
```

V. Calculating the Binomial Distribution

In many applications we might encounter large sample sizes and small probabilities. For instance, in safety-critical applications like self-driving cars or aerospace it is common to have probabilities in the order of 10^{-7}–10^{-4} [33]. For Algorithm 1 the question arises whether computing the cumulative distribution function (cdf) $F_{\mathrm{Bin}(n,p_0)}$ of the binomial distribution in Python using the package SciPy is feasible for the given parameters.

As an example of a realistic parameter setting, we consider a sample size of magnitude $n = 10^5$, a significance level of $\alpha = 10^{-4}$ and $s = 1,000$. Since we have

$$F_{\mathrm{Bin}(n,p_0)}(s) = \sum_{k=0}^{s} \binom{n}{k} p_0^k (1-p_0)^{n-k} \ ,$$

we encounter terms such as $n! = 10^5!$ and $p_0^s = 10^{-4000}$.

Using the central limit theorem and Stein's method we give two tests, that strongly suggest correctness of the SciPy implementation of $F_{\mathrm{Bin}(n,p_0)}$.

a) Test based on central limit theorem.: Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of iid $\mathrm{Ber}(p)$ distributed random variables. Define $Y_n := X_1 + \cdots + X_n$. According to the central limit theorem we have

$$P\left(\frac{Y_n - np}{\sigma\sqrt{n}} \le s\right) \to \Phi(s)$$

for all $s \in \mathbb{R}$, $\sigma = \sqrt{p(1-p)}$ being the standard deviation of X_i, and Φ being the cdf of the standard Gaussian distribution. On the other hand, it holds that

$$\begin{aligned} P\left(\frac{Y_n - n\cdot p}{\sigma\sqrt{n}} \le s\right) &= P\left(Y_n \le s\cdot\sigma\sqrt{n} + n\cdot p\right) \\ &= F_{\mathrm{Bin}(n,p)}\left(s\cdot\sigma\sqrt{n} + n\cdot p\right) \ . \end{aligned}$$

The Berry-Esseen Theorem provides the error bound for the convergence

$$\left|F_{\mathrm{Bin}(n,p)}\left(s\cdot\sigma\sqrt{n} + n\cdot p\right) - \Phi(s)\right| \le \frac{M}{\sqrt{n}}\frac{1-2p(1-p)}{\sigma} \ , \tag{3}$$

according to the central limit theorem, with M being a universal constant satisfying $M < 0.41$. Extensive numerical experiments verify that the SciPy implementation of $F_{\mathrm{Bin}(n,p)}$ satisfies this inequality. The results are provided in the Appendix.

b) Test based on Stein's method.: An implication of Stein's method is

$$\sup_{A\subset\mathbb{N}} |\mathrm{Bin}(n,p)(A) - \mathrm{Pois}(n\cdot p)(A)| \le p \tag{4}$$

for all $n \in \mathbb{N}$ and $p > 0$. Here, $\mathrm{Pois}(\lambda)$ denotes the Poisson distribution with parameter λ. Extensive numerical experiments again verify that the SciPy implementation of the binomial distribution satisfies this inequality.

The successful evaluations should give us confidence that the SciPy implementation is reasonably accurate.

VI. Experiments

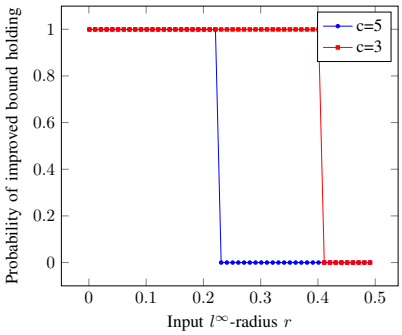
To compare our proposed methods to the state of the art, we conducted two experiments. In our first experiment we compare auto_LiRPA, Marabou, and ERAN against the two-stage method (cf. Section IV) being applied on top of these methods. We selected these methods based on the following two criteria. First, they have proven their performance repeatedly, e.g., in the annual VNN competition [34] and achieved top performance. Second, their code is publicly available, maintained, usable, and requires only one commercial license, which is Gurobi for ERAN and Marabou. We utilize neural network models trained on the California Housing[1] (regression) and MNIST[2] (classification) dataset. We aimed to apply the two-stage method in a way that is most comparable to other existing methods. To achieve this, we chose rather small models with standard operations and compared the size of the over-approximation sets provided by the respective methods. By using our two-stage method on top of other methods we greatly reduced the size of the over-approximation sets and improved the resulting bounds.

In our second experiment we intend to present the strengths of our sampling method, which lie in providing global guarantees on classification labels for large models in a setting of meaningful random perturbation. To do this, we apply the sampling method to a ResNet50 trained on the ImageNet dataset and evaluate its global robustness against meaningful perturbations.

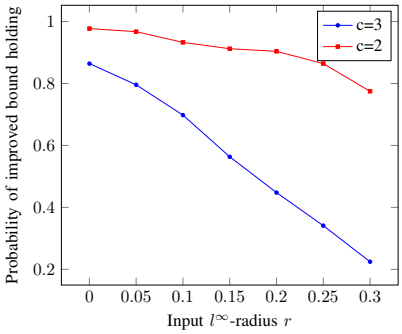
A. Evaluation of Two-Stage Method

We now describe the first experiment in detail. For a neural network f the problem we consider is to determine a set M that is a Cartesian product of intervals, such that

[1]https://keras.io/api/datasets/california_housing
[2]https://keras.io/api/datasets/mnist

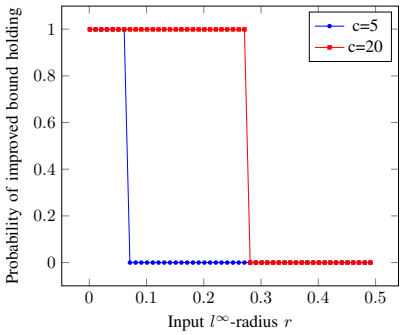


(a) Improving auto_LiRPA bounds

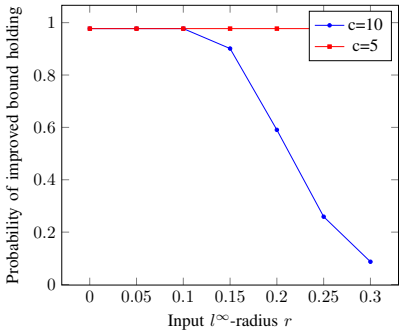


(b) Improving ERAN bounds

Fig. 1: California Housing model



(a) Improving auto_LiRPA bounds



(b) Improving ERAN bounds

Fig. 2: MNIST model

$f(x) - f(y) \in M$ for all x and y with an l^∞ distance smaller than a given number r. In other words we want to find a bound on the output distance for a given input distance. For regression models this can naturally be considered as global robustness and for classification models it is a bound on the volatility of the output vector underlying the classification decision. Our methods can guarantee with high probability output difference sets M that have a fraction of the size of the sets provided by state of the art methods. The methods we use for comparison natively support only local robustness queries. In order to perform global verification on a neural network f, we encoded f twice in parallel into a 'twin-network' g, similar to [31]. The network g acts in the following way: $g(x, \epsilon) = f(x) - f(x+\epsilon)$. We achieved this mainly by giving the weight matrices of g a block diagonal structure, with each block representing the weight matrix of f. Now a global robustness query with input l^∞ distance ϵ can be translated into the following local robustness query around zero: Define $N := Z \times [-\epsilon, \epsilon]^d$, where Z is the input space, e.g., $[0, 1]^d$, and d is the input dimension. Find an over-approximation set $M \supset g(0+N)$. This is because $g(N) = \{f(x) - f(y) : ||x - y||_{l^\infty} < \epsilon\}$.

The neural networks we use, were trained on the California Housing and the MNIST dataset, respectively. They have one hidden layer of 20 and 50 neurons, respectively, and ReLU activation functions. As an optimizer we used Adam. For MNIST we trained the network for six epochs with a learning rate of 10^{-3}. The classification accuracy is 97%. For California Housing we normalized the input data, used an 80/20 training/test split, trained for 100 epochs, and achieved a test mean squared error of 0.31. For auto_LiRPA the entire respective test sets were used, for ERAN a uniformly random subset of 500 inputs was selected from each dataset to make the computation time feasible. The significance level was $\alpha = 10^{-5}$ and the accuracy was $\text{acc} = 10^{-5}$.

During the course of our experiments, we found Marabou to be infeasible for the problem we considered. A single query for the MNIST model did not terminate within three days and many queries would have been required. For a small model with a single hidden layer of 10 neurons trained for MNIST, Marabou gave a guarantee that does not hold. For an input l^∞-radius of 0.015 Marabou gave the guarantee that the output differences are smaller than 10^{-5}. A counterexample, however, had an output difference of 1.52, showing that the provided guarantees can be incorrect. Although Reluplex is described as suitable for our experiment [31] and Marabou is described as building on top of Reluplex [16], for the aforementioned reasons, we do not consider Marabou in the following. The computations involving ERAN took a few hours, whereas the auto_LiRPA part of the experiment terminated within matter of minutes. So the improvement of the bounds comes at a manageable cost for ERAN and rather cheap for auto_LiRPA in terms of computation time.

Let us turn to Figures 1 and 2. We determined two output difference sets M_{LiRPA} and M_{ERAN} using the methods auto_LiRPA and ERAN, respectively. For input l^∞-radii from 0.001 to 0.5 and a given positive constant c the figures show the guaranteed probability that $f(x) - f(y) \in \frac{1}{c} M$ as provided by our method, for all x, y satisfying $||x - y||_{l^\infty} < r$. In

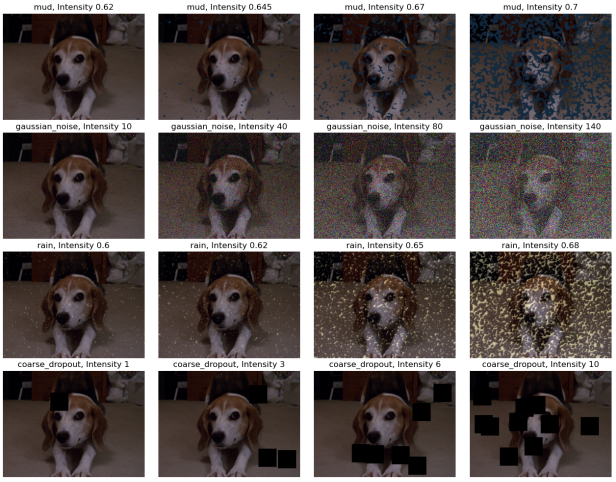
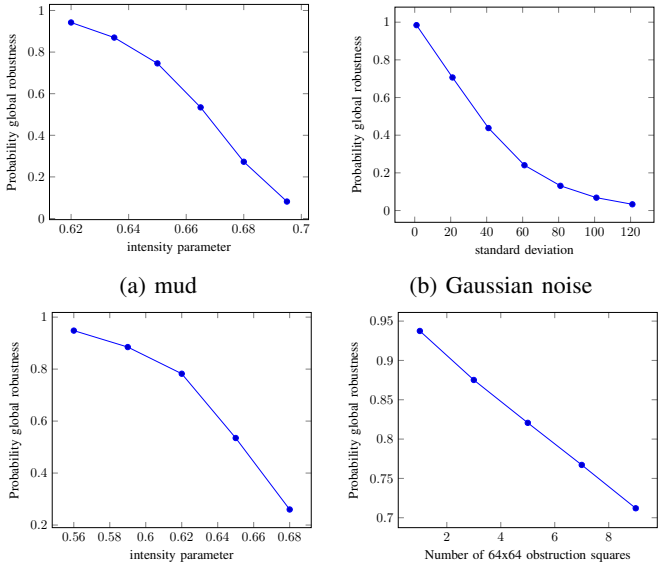


Fig. 3: Examples of ImageNet image perturbations. First row: mud, second row: Gaussian noise, third row: rain, fourth row: random obstruction.



(a) mud (b) Gaussian noise

(c) Rain (d) Random occlusion

Fig. 4: Sampling method, ImageNet

other words, Figures 1 and 2 show the guaranteed probability with which the two-stage method improves the bounds from the other methods by a factor of c. As we can see, for the Figures 1 and 2 the smaller the input radius, the higher is the probability with which our method can guarantee an improvement. We hypothesize that the original bound is a rather rough estimate and that our bound can better capture the intricacies of individual input regions by sampling them. For a very large noise instead, hardly any input is recognizable and therefore no improvement can be expected.

B. Evaluation of Sampling Method

With our second experiment we try to demonstrate the strengths of our sampling method in a use case that is relevant for many real-world applications. While most common classification models such as ResNet, VGG, DenseNet, MobileNet, or EfficientNet contain operations like, for example, max pooling with different kernel size and stride that are not supported by state of the art verification tools, our sampling based method is compatible with any model, since it relies solely on sampling. For the same reason it can handle large models efficiently. Another advantage of our method is that besides l^p-perturbations, it can deal with meaningful random perturbations such as rain or mud on the camera lens, sensor noise, or random occlusion. Therefore, we consider a ResNet50 model pretrained on the ImageNet dataset and evaluate its robustness with regard to the aforementioned perturbations. We used Albumentations as an implementation of the perturbations. As a classifier we used a ResNet50, which we imported from Torchvision including pretrained weights. As the dataset we used the ImageNet validation set. All experiments were conducted on an Nvidia A100 GPU.

Figure 3 shows the progression of each perturbation from slightly corrupted to barely recognizable. Let us consider the example of the mud perturbation. Every image is perturbed in a way that represents a random splatter of mud on a the camera lens, with little mud for low intensity and a lot for high intensity. Figure 4a shows the probability that the classification label of the ResNet50 does not change when a random image is taken from the same distribution as ImageNet and mud is randomly splattered on the camera lens. Figures 4b to 4d show the robustness evaluation for all other considered perturbations. These guarantees could be used for real-world safety evaluations.

In all experiments, we make the assumption that the test sets are iid samples from the input distribution. Theorem 2 implies that in order to tighten the bounds further, one needs larger samples.

It is worth mentioning that an advantage of our method and the notion of robustness we are considering is that we can use unlabeled data for sampling since our definition of robustness involves only changes of the predicted labels, not the ground truth.

VII. Discussion and Future Work

We proposed two notions for global probabilistic robustness of arbitrary supervised machine learning models and two methods for providing corresponding guarantees. Both methods can deal with all sorts of random perturbations on the input data and work with arbitrary models. The advantages of the two-stage approach lie in its flexibility to include any desired kind of robustness by building on top of a corresponding local method. It is therefore applicable to a large variety of robustness problems. The sampling method, on the other hand, is easily compatible with all sorts of large machine learning models and it can deal with semantically meaningful perturbations.

Future work might address two questions. The first one is, having a novel measure of global robustness, how can

training procedures be modified to improve global robustness. The second open question is, how the provided methods can be further improved to tighten the resulting bounds, perhaps including more knowledge of the structure of the underlying machine learning model.

APPENDIX

We performed the test based on the Berry-Esseen bound for $n \in \{10^6, 10^7\}$ and s in the range from 0 to n with a stepsize of $n \cdot 10^{-4}$. For p we chose two ranges. First we set p in the range from 0.05 to 0.95 with a step size from 0.05 and then p in the range from 10^{-5} to 0.05 with a step size of $5 \cdot 10^{-5}$. There were no violations of the Berry-Esseen bound (3) or the bound derived from Stein's method (4) for all specified parameters.

REFERENCES

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, pp. 84—90, May 2017.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv*, vol. abs/1412.6572, 2014.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv*, vol. abs/1312.6199, 2014.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[7] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. Jordan, "Theoretically Principled Trade-off between Robustness and Accuracy," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 97, June 2019, pp. 7472–7482.

[8] S. Gu and L. Rigazio, "Towards Deep Neural Network Architectures Robust to Adversarial Examples," in *3rd International Conference on Learning Representations (ICLR), Workshop Track Proceedings*, May 2015.

[9] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "Ai2: Safety and robustness certification of neural networks with abstract interpretation," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 3–18.

[11] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Efficient Formal Safety Analysis of Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[12] L. Weng, P.-Y. Chen, L. Nguyen, M. Squillante, A. Boopathy, I. Oseledets, and L. Daniel, "PROVEN: Verifying Robustness of Neural Networks with a Probabilistic Approach," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 97, June 2019, pp. 6727–6736.

[13] M. Pautov, N. Tursynbek, M. Munkhoeva, N. Muravev, A. Petiushko, and I. Oseledets, "CC-CERT: A Probabilistic Approach to Certify General Robustness of Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, pp. 7975–7983, June 2022.

[14] B. Wang, S. Webb, and T. Rainforth, "Statistically robust neural network classification," in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*, ser. Proceedings of Machine Learning Research, vol. 161, July 2021, pp. 1735–1745.

[15] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, "Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond," *arXiv*, vol. abs/2002.12920, 2020.

[16] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljić, D. L. Dill, M. J. Kochenderfer, and C. Barrett, "The Marabou Framework for Verification and Analysis of Deep Neural Networks," in *Computer Aided Verification*, I. Dillig and S. Tasiran, Eds. Cham: Springer International Publishing, 2019, pp. 443–452.

[17] W. Ryou, J. Chen, M. Balunovic, G. Singh, A. Dan, and M. Vechev, "Scalable Polyhedral Verification of Recurrent Neural Networks," 2021.

[18] C. Kanbak, S.-M. Moosavi-Dezfooli, and P. Frossard, "Geometric Robustness of Deep Networks: Analysis and Improvement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth, "Unrestricted Adversarial Examples via Semantic Manipulation," 2020.

[20] H. Hosseini and R. Poovendran, "Semantic Adversarial Examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[21] H. D. Liu, M. Tao, C. Li, D. Nowrouzezahrai, and A. Jacobson, "Adversarial Geometry and Lighting using a Differentiable Renderer," *arXiv*, vol. abs/1808.02651, 2018.

[22] A. Joshi, A. Mukherjee, S. Sarkar, and C. Hegde, "Semantic Adversarial Attacks: Parametric Transformations That Fool Deep Classifiers," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4772–4782.

[23] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen, "Differentiable Monte Carlo Ray Tracing through Edge Sampling," *ACM Trans. Graph.*, vol. 37, no. 6, December 2018.

[24] A. Bibi, M. Alfadly, and B. Ghanem, "Analytic Expressions for Probabilistic Moments of PL-DNN With Gaussian Input," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] R. Mangal, A. V. Nori, and A. Orso, "Robustness of Neural Networks: A Probabilistic and Practical Approach," 2019.

[26] K. Tit, T. Furon, and M. Rousset, "Efficient Statistical Assessment of Neural Network Corruption Robustness," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 9253–9263.

[27] F. Fu, Z. Wang, J. Fan, Y. Wang, C. Huang, X. Chen, Q. Zhu, and W. Li, "REGLO: Provable Neural Network Repair for Global Robustness Properties," in *NeurIPS Workshop on Trustworthy and Socially Responsible Machine Learning*, 2022.

[28] Z. Wang, C. Huang, and Q. Zhu, "Efficient Global Robustness Certification of Neural Networks via Interleaving Twin-Network Encoding," *arXiv*, vol. abs/2203.14141, 2022.

[29] Z. Wang, Y. Wang, F. Fu, R. Jiao, C. Huang, W. Li, and Q. Zhu, "A Tool for Neural Network Global Robustness Certification and Training," *arXiv*, vol. abs/2208.07289, 2022.

[30] K. Leino, Z. Wang, and M. Fredrikson, "Globally-Robust Neural Networks," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 139, July 2021, pp. 6212–6222.

[31] G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer, "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks," *arXiv*, vol. abs/1808.02651, 2017.

[32] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[33] S. L. Brunton, J. N. Kutz, K. Manohar, A. Y. Aravkin, K. Morgansen, J. Klemisch, N. Goebel, J. Buttrick, J. Poskin, A. W. Blom-Schieber, T. Hogan, and D. McDonald, "Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning," *AIAA Journal*, vol. 59, no. 8, pp. 2820—2847, August 2021.

[34] C. Brix, M. N. Müller, S. Bak, T. T. Johnson, and C. Liu, "First Three Years of the International Verification of Neural Networks Competition (VNN-COMP)," 2023.