Probabilistic Global Robustness Verification of Arbitrary Supervised Machine Learning Models

Max-Lion Schumacher [1] **Marco F. Huber** [1 2]

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. This is applicable to any classification or regression model and any random input perturbation. We then illustrate the resulting bounds on classifiers for the MNIST and CIFAR-10 datasets.

1. Introduction

Neural Networks have demonstrated unprecedented performance for various tasks (Dosovitskiy et al., 2021; Vaswani et al., 2017; Krizhevsky et al., 2017), yet they are susceptible to adversarial perturbations (Goodfellow et al., 2014; Szegedy et al., 2014). Several approaches for assessing and mitigating this risk have been proposed (Madry et al., 2018; Zhang et al., 2019; Gu & Rigazio, 2015; Moosavi-Dezfooli et al., 2016). This work has, to a large degree, focused on formal guarantees for worst-case robustness (Gehr et al., 2018; Wang et al., 2018). 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 (Weng et al., 2019; Pautov et al., 2022; Mangal et al., 2019). The work of (TIT et al., 2021) involves sophisticated sampling methods and statistical tests. The latter will be important for our methods as well.

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. The approach of 'repairing' the model with respect to non-robust areas is pursued by (Fu et al., 2022). A method based on mixed-integer linear programming and over approximation was created by (Wang et al., 2022). Like (Leino et al., 2021; Katz et al., 2017), all of these approaches are based on Lipschitz continuity-type definitions of global robustness and the potential for applying them to classification problems is therefore limited. The reason is that a model with a small Lipschitz constant still cannot 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.

The aforementioned methods cannot deal with randomness, in contrast to the one developed by (Wang et al., 2021). In this work risk measures are used to quantify robustness. Theoretical bounds for the risk measures have been derived but there are no bounds on the probability of robustness. This, instead, is a major focus of our work. 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 illustrate the resulting bounds, both obtained merely by sampling and based on pointwise methods, on classifiers for the MNIST and CIFAR-10 datasets.

[1]Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Stuttgart, Germany. [2]Institute of Industrial Manufacturing and Management IFF, University of Stuttgart, Stuttgart, Germany. Correspondence to: Max-Lion Schumacher <max-lion.schumacher@ipa.fraunhofer.de>.

Presented at the 2^{nd} Workshop on Formal Verification of Machine Learning, co-located with the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA., 2023.

To summarize, our key contributions are as follows:

- 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.

This paper is structured as follows: In Section 2 we give a definition for global probabilistic robustness, derive a method based only on sampling and prove the resulting bounds. In Section 3 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 prove the resulting bounds. In Section 4 we illustrate our methods with several examples. This paper closes with a discussion and an outlook on future work in Section 5.

2. 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 2.1. A classifier model f is said to be globally robust with probability ϵ if the following bound is satisfied:

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

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, in the above definition, the property would be

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

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.

Our goal now, for a given model f, is to find a real number ϵ as small as possible so that f is globally robust with probability ϵ. In order to do this, 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, for any set M, let 1_M be defined as the indicator function of M. We observe that

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

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

$$S := 1_{\{f(X_1) \neq f(T_1(X_1))\}} + \cdots + 1_{\{f(X_n) \neq f(T_n(X_n))\}} \,,$$

that $S \sim \mathrm{Bin}(n, p)$. Now the idea is to use this random variable S to perform a statistical test on the parameter p. This leads to our first of two main results, which we prove in the Appendix, in Section A.

Theorem 2.2. *Let $x_1, \ldots, x_n$ be an input sample, $t_1, \ldots, t_n$ a sample of perturbations and s an observation of S. For any $0 < \psi_0 < 1$ denote by p_0 the result of Algorithm 1. Then the model f is globally robust with probability p_0, i.e., the inequality*

$$P(f(X) \neq f(T(X))) < p_0$$

holds with a false positive error bound $P(S \leq s) \leq \psi_0$ in any case where $p \geq p_0$.

This theorem means that we can accept

$$P(f(X) \neq f(T(X))) < p_0 \tag{1}$$

and the probability of being wrong is at most ψ_0. Or put in other words, we accept the hypothesis of the inequality (1) being true by means of a statistical test with significance level ψ_0.

3. 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 (Pautov et al., 2022; Weng et al., 2019). We consider a general setting, where the nature of the guarantees provided by the underlying pointwise methods can be deterministic

Algorithm 1 Sampling method providing a global robustness bound

Require: Classifier f, set of input samples $\mathrm{input\_data}$, random perturbation function pert, p-value constraint ψ_0, parameter acc specifying maximal distance of the output to the optimal solution
Ensure: Optimal bound for probability of global robustness p_0

$\mathrm{pred} \leftarrow f(\mathrm{input\_data})$
$\mathrm{pred\_pert} \leftarrow f(\mathrm{pert}(\mathrm{input\_data}))$
$s \leftarrow \mathrm{how\_many\_differ}(\mathrm{pred}, \mathrm{pred\_pert})$
$\mathrm{lower} \leftarrow 0$
$\mathrm{upper} \leftarrow 1$
$n \leftarrow \mathrm{length}(\mathrm{input\_data})$
while $\mathrm{upper} - \mathrm{lower} > \mathrm{acc}$ **do**
  $m \leftarrow (\mathrm{lower} + \mathrm{upper})/2$
  if $\mathrm{binomial\_cdf}(s, n, m) > \psi_0$ **then**
    $\mathrm{lower} \leftarrow m$
  else
    $\mathrm{upper} \leftarrow m$
  end if
end while
$p_0 \leftarrow (\mathrm{lower} + \mathrm{upper})/2$

Algorithm 2 Two-stage method providing a global robustness bound

Require: Classifier f, set of input samples $\mathrm{input\_data}$, local robustness assessment method r, local robustness threshold c, p-value constraint ψ_0, parameter acc specifying maximal distance of the output to the optimal solution
Ensure: Optimal bound for probability of global robustness p_0

$s \leftarrow 0$
for d in $\mathrm{input\_data}$ **do**
  if $r(d) > c$ **then**
    $s \leftarrow s + 1$
  end if
end for
$\mathrm{lower} \leftarrow 0$
$\mathrm{upper} \leftarrow 1$
$n \leftarrow \mathrm{length}(\mathrm{input\_data})$
while $\mathrm{upper} - \mathrm{lower} > \mathrm{acc}$ **do**
  $m \leftarrow (lower + upper)/2$
  if $\mathrm{binomial\_cdf}(s, n, m) > \psi_0$ **then**
    $\mathrm{lower} \leftarrow m$
  else
    $\mathrm{upper} \leftarrow m$
  end if
end while
$\mathrm{p_0} \leftarrow (\mathrm{lower} + \mathrm{upper})/2$

or probabilistic. Let the model $f : \mathcal{X} \to \mathcal{Y}$ and it's input X be defined as in Section 2. In order to specify our notion of global robustness in this setting, we require the following definition.

Definition 3.1. 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 example, r could be the result of a method based on mixed-integer linear programming such that $r(x) = 1$, if there is an adversarial example in a neighborhood of x and $r(x) = 0$, otherwise. To give another example, r could be the result of a method, that provides an upper bound on the probability that robustness fails for the input x.

Definition 3.2. 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:

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

The deviation level c can be viewed as the level of local robustness that we are willing to accept. So if a model f satisfies Definition 3.2, this means the probability of encountering an input for which the model is not 'robust enough' is smaller than ϵ.

Denote $p := D(r(f(X)) > c)$ and $S := 1_{\{r(f(X_1))>c\}} + \cdots + 1_{\{r(f(X_n))>c\}}$. The second main result of this paper, which we prove in the Appendix, in Section A, is the following

Theorem 3.3. *Let $x_1, \ldots, x_n$ be an input sample, r a robustness score, s an observation of S and $c > 0$. For any $0 < \psi_0 < 1$ denote by p_0 the result of Algorithm 2. Then the model f is globally robust with probability p_0, with respect to r and c with the false positive error bound*

$$\sup_{D: p \geq p_0} D(S < s) < \psi_0 \, .$$

4. Experiments

In the following, we apply the methods previously introduced to classifiers on the MNIST and CIFAR-10 dataset, which we describe in more detail in the Appendix. For the two-stage method, in terms of the pointwise methods, we use one that provides a deterministic bound and one that provides a probabilistic bound, namely interval bound propagation (IBP) and the CC-Cert algorithm (Pautov et al., 2022). For both methods we used our own implementations.

In all experiments, we make the assumption that the test sets of MNIST and CIFAR-10 are samples from an iid sequence of random variables according to the input distribution and

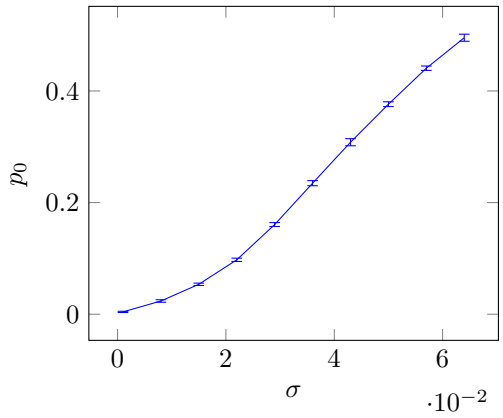

Figure 1. Sampling, CIFAR-10, Gaussian noise

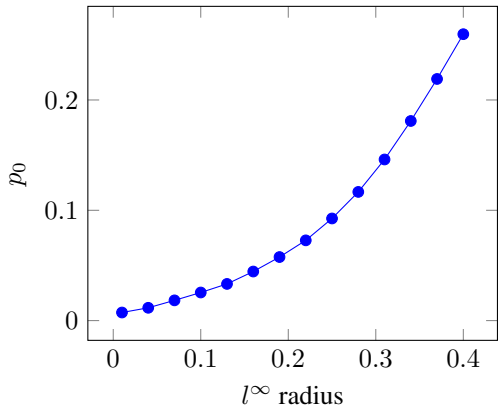

Figure 2. Two-stage, IBP (deterministic, so no confidence intervals), MNIST

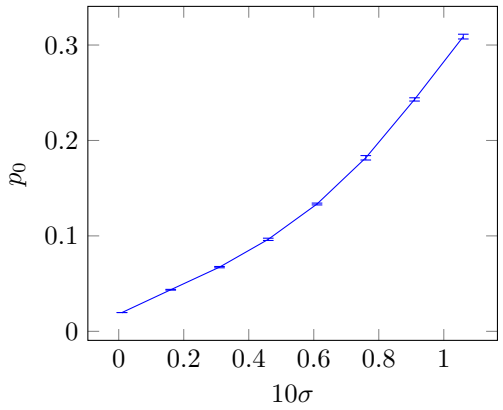

Figure 3. Two-stage, CC-Cert, MNIST, critical level $c = 0.01$, Gaussian noise

we use the entire test sets containing 10^5 images each. We have conducted every random experiment 20 times and we provide confidence intervals in all respective plots. The confidence intervals in the plots represent the 2σ range. All experiments were conducted with the constraint $\psi_0 = 10^{-5}$.

According to the law of large numbers the average number of input points where robustness fails, converges to the true failure probability p. For large enough samples we would expect p_0 to be a good approximation of p.

Depending on the specific robustness problem one is facing in applications, a two-stage approach might be more suitable. For example, in case of self-driving cars, beside the issue of signal noise, one might also face purposefully placed adversarial patches on walls or road signs. In this case an approach based only on sampling might not be accurate enough and a worst-case robustness guarantee is necessary. This can be achieved by using pointwise worst-case methods such as IBP, which we demonstrate in Figure 2 using Algorithm 2. The verification approach IBP of propagating interval bounds through a neural network is described in (Albarghouthi, 2021).

5. 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. We might expect that a two-stage approach based on sophisticated pointwise methods obtains tighter bounds. However, depending on the size of the input space and the performance of the pointwise method, gathering a large enough sample can be computationally demanding. Here lies one of the major advantages of the sampling method. It requires only simple model evaluations and is therefore computationally very efficient. This can be helpful, since high computation times are a known bottleneck for many robustness verification methods. A direct comparison is, however, difficult because the resulting probabilities of the respective approaches refer to different probability measures. Our method can also provide a good balance between taking worst-case robustness into account and making the certification feasible by introducing randomness when we use the two stage approach with a sound deterministic pointwise method. A limitation, however, is that when safety requirements are so high that probabilities are not acceptable, our approach cannot be applied.

In terms of future work, experiments are required to compare our methods to existing ones, even though a direct comparison might be somewhat difficult because of our different setting. Furthermore, two questions questions might be addressed. 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 knowl-

edge of the structure of the underlying machine learning model.

References

Albarghouthi, A. Introduction to neural network verification, 2021.

Brunton, S. L., Kutz, J. N., Manohar, K., Aravkin, A. Y., Morgansen, K., Klemisch, J., Goebel, N., Buttrick, J., Poskin, J., Blom-Schieber, A. W., Hogan, T., and McDonald, D. Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning. *AIAA Journal*, 59 (8):2820—2847, August 2021. doi: 10.2514/1.J060131. URL https://doi.org/10.2514/1.J060131.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Fu, F., Wang, Z., Fan, J., Wang, Y., Huang, C., Chen, X., Zhu, Q., and Li, W. REGLO: Provable neural network repair for global robustness properties. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*, 2022. URL https://openreview.net/forum?id=FRTXdodwsoA.

Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2018. doi: 10.1109/SP.2018.00058.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

Gu, S. and Rigazio, L. Towards deep neural network architectures robust to adversarial examples. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.5068.

Katz, G., Barrett, C., Dill, D., Julian, K., and Kochenderfer, M. Reluplex: An efficient smt solver for verifying deep neural networks, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL https://doi.org/10.1145/3065386.

Leino, K., Wang, Z., and Fredrikson, M. Globally-robust neural networks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6212–6222. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/leino21a.html.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Mangal, R., Nori, A. V., and Orso, A. Robustness of neural networks: A probabilistic and practical approach, 2019.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. 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.

Pautov, M., Tursynbek, N., Munkhoeva, M., Muravev, N., Petiushko, A., and Oseledets, I. Cc-cert: A probabilistic approach to certify general robustness of neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7975–7983, Jun. 2022. doi: 10.1609/aaai.v36i7.20768. URL https://ojs.aaai.org/index.php/AAAI/article/view/20768.

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

TIT, K., Furon, T., and ROUSSET, M. Efficient statistical assessment of neural network corruption robustness. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 9253–9263. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4d215ab7508a3e089af43fb605dd27d1-Paper.pdf.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, B., Webb, S., and Rainforth, T. Statistically robust neural network classification. In de Campos, C. and Maathuis, M. H. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1735–1745. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/wang21b.html.

Wang, S., Pei, K., Whitehouse, J., Yang, J., and Jana, S. Efficient formal safety analysis of neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2ecd2bd94734e5dd392d8678bc64cdab-Paper.pdf.

Wang, Z., Huang, C., and Zhu, Q. Efficient global robustness certification of neural networks via interleaving twin-network encoding, 2022.

Weng, L., Chen, P.-Y., Nguyen, L., Squillante, M., Boopathy, A., Oseledets, I., and Daniel, L. PROVEN: Verifying robustness of neural networks with a probabilistic approach. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6727–6736. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/weng19a.html.

Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. E., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhang19p.html.

A. Proofs

Proof of Theorem 2.2. We consider a random iid sample $X_1, \dots, X_n$, $T_1, \dots, T_n$ 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 $i \in \{1, \dots, n\}$, where $\mathrm{Ber}(p)$ denotes the Bernoulli distribution with parameter p. These random variables are independent, as we considered X_i and T_i to be iid. It follows that

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

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

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

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 $K = \{0, 1, 2, \dots, k\}$, 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 $\psi \leq \psi_0$, with ψ_0 being a user-defined significance level, and determine the smallest p_0, for which H_A can be accepted by this method, given an observation s. Formally, this can be expressed as constrained optimization problem

$$\begin{aligned} &\min p_0 \\ &\text{s.t. } \psi \leq \psi_0 \, . \end{aligned}$$

Since ψ is monotonously decreasing as a function of p_0, for a given s, this optimization problem can be easily solved by bisection. This is described by Algorithm 1, which concludes the proof of Theorem 2.2.

Proof of Theorem 3.3. Let a robustness score r and a critical deviation level c be given. 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 the previous proof, we can use a binomial test. Analogously, for a given observation s of S and a p-value boundary ψ_0 we get an optimal parameter p_0 such that we can statistically prove

$$D(r(f(X)) > c) = p < p_0$$

with a test ϕ having a false positive error bound

$$\sup_{D:p\geq p_0} D(\phi(S) = 1) < \psi_0\,.$$

B. 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} (Brunton et al., 2021). For both Algorithm 1 and 2 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 p-value constraint of $\psi_0 = 10^{-4}$ 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 $F_{\mathrm{Bin}(n,p_0)}$.

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

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

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

$$\begin{aligned} P\left(\frac{Y_n - np}{\sigma\sqrt{n}} \leq s\right) &= P\left(Y_n \leq \sqrt{np(1-p)}s + np\right) \\ &= F_{\mathrm{Bin}(n,p)}\left(\sqrt{np(1-p)}s + np\right)\,. \end{aligned}$$

The Berry-Esseen Theorem provides an error bound for the convergence according to the central limit theorem. Hence we get

$$\left|F_{\mathrm{Bin}(n,p)}\left(\sqrt{np(1-p)}s + np\right) - \Phi(s)\right| \leq \frac{M}{\sqrt{n}} \frac{1-2p(1-p)}{\sqrt{p(1-p)}}\,,$$

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.

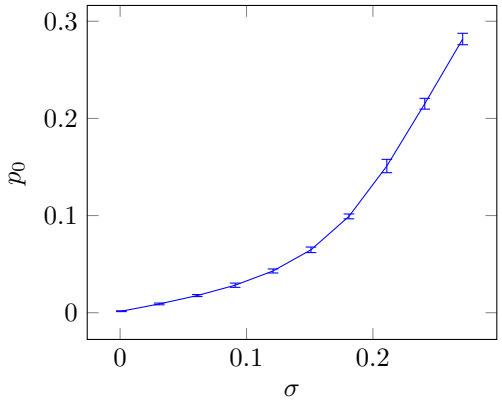

Figure 4. Sampling, MNIST, Gaussian noise

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

$$\sup_{A\subset\mathbb{N}} |\mathrm{Bin}(n,p)(A) - \mathrm{Pois}(np)(A)| \leq p$$

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.

Numerical experiments. 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 $n10^{-4}$. 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 $5 * 10{-}5$. There were no violations of the Berry-Esseen bound or the bound derived from Stein's method, e.g.

$$\left|F_{\mathrm{Bin}(n,p)}\left(\sqrt{np(1-p)}s + np\right) - \Phi(s)\right| \leq \frac{M}{\sqrt{n}} \frac{1-2p(1-p)}{\sqrt{p(1-p)}} ,$$

and

$$|F_{\mathrm{Bin}(n,p)}(s) - F_{\mathrm{Pois}(np)}(s)| < p$$

was satisfied for all specified parameters. Here $F_{\mathrm{Pois}(np)}$ denotes the cumulative distribution function of the Poisson distribution with parameter np.

C. Further Experiments and Training Details

As a classifier for the MNIST dataset we use a neural network with one hidden layer of 500 neurons and ReLu activation functions. We have trained it for six epochs using the Adam optimizer with a learning rate of 10^{-3}. For the CIFAR-10 dataset we have used a convolutional neural network (CNN). We have trained it for 86 epochs using cross entropy loss and stochastic gradient descent with a learning rate of 0.01, momentum of 0.9 and weight decay of 10^{-4}. We performed the following transformations on the training data: Random crop with $\mathrm{padding} = 4$, random horizontal flip, random rotation with parameter $\mathrm{degrees} = 10$. We used the respective implementations of PyTorch. The structure of the CNN is described in Table 1. Again, all experiments were conducted with the constraint $\psi_0 = 10^{-5}$. Training and evaluation was performed on a laptop with an Intel i7 CPU and an NVIDIA Quadro T2000 GPU.

We also did some experiments with different random perturbations as shown in Table 2.

Table 1. CNN Architecture

Layer (type)	Input Channels	Output Channels	Parameters
Conv2D	3	64	kernel_size=3, padding=1
ReLU	-	-	-
BatchNorm2D	64	64	-
Conv2D	64	64	kernel_size=3, padding=1
ReLU	-	-	-
BatchNorm2D	64	64	-
MaxPool2D	-	-	kernel_size=2, stride=2
Dropout	-	-	p=0.2
Conv2D	64	128	kernel_size=3, padding=1
ReLU	-	-	-
BatchNorm2D	128	128	-
Conv2D	128	128	kernel_size=3, padding=1
ReLU	-	-	-
BatchNorm2D	128	128	-
MaxPool2D	-	-	kernel_size=2, stride=2
Dropout	-	-	p=0.2
Conv2D	128	256	kernel_size=3, padding=1
ReLU	-	-	-
BatchNorm2D	256	256	-
Conv2D	256	256	kernel_size=3, padding=1
ReLU	-	-	-
BatchNorm2D	256	256	-
MaxPool2D	-	-	kernel_size=2, stride=2
Dropout	-	-	p=0.2
Flatten	-	-	-
Linear	4096	256	-
ReLU	-	-	-
BatchNorm1D	256	256	-
Dropout	-	-	p=0.5
Linear	256	10	-
Softmax	-	-	dim=1

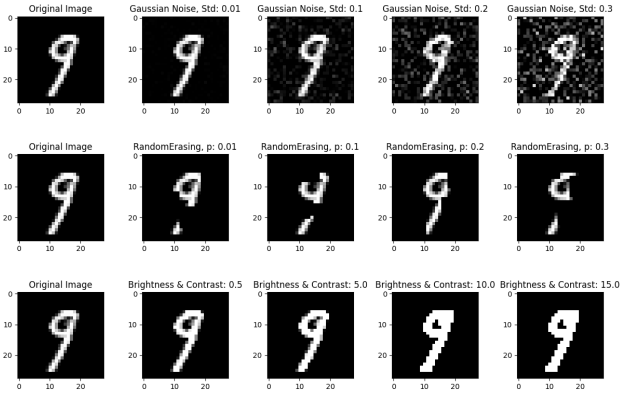

Figure 5. Examples of MNIST image perturbations. First row: additive Gaussian noise. Second row: random deletion. Third row: brightness and contrast adjustment.

Table 2. Evaluation of the sampling method (Algorithm 1) for the deletion as well as the brightness and contrast transformation.

dataset	perturbation	parameters	$p_0 \pm \sigma$ in %
MNIST	deletion	$q = 0.01$	0.54 ± 0.05
	deletion	$q = 0.1$	3.3 ± 0.15
	deletion	$q = 0.3$	8.5 ± 0.29
	brightness & contrast	$q = 0.15$	0.57 ± 0.06
	brightness & contrast	$q = 1$	21 ± 0.47
	brightness & contrast	$q = 3$	12 ± 0.38
CIFAR-10	deletion	$q = 6.7 * 10^{-4}$	0.17 ± 0.02
	deletion	$q = 0.013$	1.5 ± 0.10
	deletion	$q = 0.033$	6.3 ± 0.25
	brightness & contrast	$q = 0.001$	0.17 ± 0.02
	brightness & contrast	$q = 0.067$	1.51 ± 0.12
	brightness & contrast	$q = 0.67$	30 ± 0.38