PRoA: A Probabilistic Robustness Assessment against Functional Perturbations[*]

Tianle Zhang[1][0000−0003−4881−2406], Wenjie Ruan ✉[1][0000−0002−8311−8738], and Jonathan E. Fieldsend[1][0000−0002−0683−2583]

University of Exeter, Exeter EX4 4PY, UK
{tz294, W.Ruan, J.E.Fieldsend}@exeter.ac.uk

Abstract. In safety-critical deep learning applications robustness measurement is a vital pre-deployment phase. However, existing robustness verification methods are not sufficiently practical for deploying machine learning systems in the real world. On the one hand, these methods attempt to claim that no perturbations can "fool" deep neural networks (DNNs), which may be too stringent in practice. On the other hand, existing works rigorously consider L_p bounded additive perturbations on the pixel space, although perturbations, such as colour shifting and geometric transformations, are more practically and frequently occurring in the real world. Thus, from the practical standpoint, we present a novel and general *probabilistic robustness assessment method* (PRoA) based on the adaptive concentration, and it can measure the robustness of deep learning models against functional perturbations. PRoA can provide statistical guarantees on the probabilistic robustness of a model, *i.e.*, the probability of failure encountered by the trained model after deployment. Our experiments demonstrate the effectiveness and flexibility of PRoA in terms of evaluating the probabilistic robustness against a broad range of functional perturbations, and PRoA can scale well to various large-scale deep neural networks compared to existing state-of-the-art baselines. For the purpose of reproducibility, we release our tool on GitHub: `https://github.com/TrustAI/PRoA`.

Keywords: Verification · Probabilistic Robustness · Functional Perturbations · Neural Networks.

1 Introduction

With the phenomenal success of Deep Neural Networks (DNNs), there is a growing and pressing need for reliable and trustworthy neural network components, particularly in safety-critical applications. Neural networks' inherent vulnerability to adversarial attacks has been receiving considerable attention from the

---

research community [21]. Numerous empirical defence approaches, including adversarial training [14], have been developed recently in response to diverse adversarial attacks. Such defence strategies, however, are subsequently overwhelmed by elaborate and advanced adversarial attacks [10].

Therefore, in order to construct safe and trustworthy deep learning models with a certain confidence, a challenge has emerged: *how can we verify or certify our models under adversarial perturbations with guarantees*? Various earlier works have attempted to quantify the deterministic robustness of a given input x concerning a specific neural network; they seek to state that no adversarial examples exist within a neighbourhood of x [9]. However, such safety requirements are not always satisfied and applicable in practice. For instance, as ISO/IEC Guide 51 [6] suggests, "*safety risks and dangers are unavoidable; residual risks persist even after risk reduction measures have been implemented*". Thus, in comparison to those ensuring deterministic robustness, it is a more practical assessment of robustness to properly confine the possibility of a failure event occurring. For example, no communication networks can guarantee that no message will be lost over a wireless communication route, and messages might be lost owing to collisions or noise contamination even with proper functioning network hardware. Occasional message loss is tolerated if the occurrence chance is within an acceptable level. However, it is still unexplored for such probabilistic robustness verification.

In the meantime, the majority of existing verification methods consider a narrow threat model with additive perturbations, *i.e.* adversarial examples are produced by adding slight tweaks (measured in L_p distance) to every single feature of normal inputs (e.g. counterexamples are generated by adding minor changes to every single pixel in an image classification task). While the additive threat model implies that the divergence between generated adversarial instances and original instances does not surpass a modest positive constant ϵ measured by L_p norm, other sorts of perturbations undetectable to humans are overlooked. For instance, cameras installed in self-driving cars may be vibrated on bumpy roads, leading to rotating or blurring photos. Resultant rotated and blurry photographs are likely to be misidentified by neural networks, even if they do not "hoodwink" human perception. Such risky and frequent scenarios motivate the robustness assessment against various general perturbations, e.g. geometric transformation like rotation and translation, and common corruptions.

In this paper, we propose a novel and scalable method called PRoA that can provide statistical guarantees on the probabilistic robustness of a large blackbox neural network against functional threat models. Specifically, in this approach, we introduce functional perturbations, including random noise, image transformations and recolouring, which occur naturally and generally, and additive perturbation would be a specific instance in which perturbation functions add a modest adjustment to each feature of inputs. Instead of worst-case based verification, this method measures the probabilistic robustness, *i.e.* accurately bounds the tolerated risk of encountering counterexamples via adaptively randomly sampling perturbations. This robustness property is more appropriate in

real-world circumstances. Furthermore, the proposed method makes no assumptions about the neural network, e.g. activation functions, layers, and neurons, etc. This grants our probabilistic robustness assessment method (PRoA) the scalability to evaluate state-of-the-art and large-scale DNNs. Our main contributions are threefold as follows:

– We propose a randomised algorithm-based framework for evaluating the probabilistic robustness of deep learning models using adaptive concentration inequality. This method is well-scalable and applicable to large and state-of-the-art black-box neural networks.
– The method is attack-agnostic and capable of providing a theoretical guarantee on the likelihood of encountering an adversarial example under parametric functional perturbations.
– Experimentally, we validate our certification method and demonstrate its practical applicability with different trained neural networks for various natural functional perturbations, e.g. geometric transformations, colour-shifted functions, and Gaussian blurring.

2   Related Work

Reachability based approaches. For a given input and a specified perturbation, reachability-based algorithms endeavour to determine the lower and upper bounds of the output. Thus, robustness can be evaluated by solving an output range analysis problem. Some reachability-based approaches employ layer-by-layer analysis to obtain the reachable range of outputs [13, 22, 19, 20, 26, 27, 18]. ExactReach [26] estimates a DNN's reachable set as a union of polytopes by setting the outputs of each layer with Relu activation to a union of polytopes. Yang et al. [27] present an exact reachability verification method utilising a facet-vertex incidence matrix. Additionally, another research approach is to employ global optimisation techniques to generate a reachable output interval. GeepGo [18] uses a global optimisation technique to find the upper and lower bounds of the outputs of Lipschitz-continuous networks. This algorithm is capable of operating on black-box DNNs. Reachability analysis can be used to address the challenge of safety verification; however, these methods often require that target networks be Lipschitz continuous over outputs, which limits their application.

Constraint based approaches. Constraint-based techniques generally transform a verification problem into a set of constraints, which can then be solved by a variety of programme solvers. In recent papers [8, 1], Katz et al. [8] introduce an SMT-based technique called Reluplex for solving queries on DNNs with Relu activation by extending a simplex algorithm, while Amir et al. [1] propose another SMT-based method by splitting constraints into easier-to-solve linear constraints. For constraint-based techniques, all types of solvers can produce a deterministic answer with guarantees, *i.e.*, they can either satisfy or violate robustness conditions. However, these techniques suffer from a scalability issue

and need to access the internal structure and parameters of the targeted DNN (in a white-box setting).

These deterministic verification approaches might be unduly pessimistic in realistic applications since they only account for the worst scenario. In contrast, PRoA focuses on the tail probability of the average case, which is more realistic in a wide range of real-world applications, and worst-case analysis can be a special case of tail risks when we take the tail probability (0%) of the most extreme performance into consideration.

Statistical approaches. Unlike the above deterministic verification methods, statistics-based techniques aim to quantify the likelihood of finding a counterexample. For example, random sampling has lately emerged as an effective statistical strategy for providing certified adversarial robustness, e.g. randomised smoothing [3, 28], cc-cert [16], and SRC [5], among others. Additionally, Webb et al. [24] propose an adaptive Monte Carlo approach, *i.e.* multi-level splitting, to estimate the probability of safety unsatisfiability, where failure occurs as an extremely rare occurrence in real-world circumstances. However, these statistics-based analyses focus on the pixel-level additive perturbations and always require assumptions upon target neural networks or distributions of input, which limits their applicability.

In contrast, we introduce a general adversarial threat model, *i.e.* functional perturbations, and PRoA aims to bound the failure chance with confidence under the functional threat model. Moreover, PRoA is able to provide rigorous robustness guarantees on black-box DNNs without any assumptions and scale to large-scale networks.

3   Preliminary

Classification program. Given a training set with N distinct samples $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ where $x_i \in \mathcal{X} = \mathbb{R}^n$ are i.i.d. samples with dimension n drawn from an unknown data distribution and $y_i \in \mathcal{R} = \{1, \ldots, K\}$ are corresponding labels. We consider a deterministic neural network $f : \mathcal{X} \to [0, 1]^K$ that maps any input to its associated output vector, and $f_k(\cdot) : \mathbb{R}^n \to [0, 1]$ is a deterministic function, representing the output confidence on label k. Our verification procedure solely requires blackbox assess to f, thus, it can obtain the corresponding output probability vector $f(x)$ for each input $x \in \mathcal{X}$.

Additive Perturbation. Given a neural network f and an input $x \in \mathcal{X}$, an adversarial example $\widetilde{x}$ of x is crafted with a slight modification to the original input such that $\arg\max_{k \in \{1,\ldots,K\}} f_k(\widetilde{x}) \neq \arg\max_{k \in \{1,\ldots,K\}} f_k(x)$; this means that the classifier assigns an incorrect label to $\widetilde{x}$ but $\widetilde{x}$ is perceptually indistinguishable from the original input x. Intuitively, slight perturbations $\delta \in \mathbb{R}^n$ can be added directly to x to yield adversarial examples $\widetilde{x} = x + \delta$, in the meantime, a L_p norm bound is normally imposed on such additive perturbations, constraining $\widetilde{x}$ to be fairly

close to x. The relevant definition is as follows:

$$\widetilde{x} = x + \delta \ and \ \|\delta\|_p \le \epsilon \ \ s.t. \ \underset{k \in \{1,...,K\}}{\arg\max} f_k(\widetilde{x}) \neq \underset{k \in \{1,...,K\}}{\arg\max} f_k(x).$$

Functional Perturbation. Unlike the additive perturbation, a normal input x is transformed using a perturbation function $\mathcal{F} : \mathcal{X} \to \mathcal{X}$ parameterised with $\theta \in \Theta$. That is to say, $x_{\mathcal{F}_\theta} = \mathcal{F}_\theta(x)$. It is worth noting that functional perturbation allows for a substantially larger pixel-based distance, which may be imperceptible to humans as well, since the perturbed version $x_{\mathcal{F}_\theta}$ consistently preserves semantic information underlying images, such as shape, boundary, and texture. Unfortunately, such perturbations may confuse the classifier $f(\cdot)$, which is capable of outputting the proper label to an undistorted image, *i.e.* $\underset{k \in \{1,...,K\}}{\arg\max} f_k(x) \neq \underset{k \in \{1,...,K\}}{\arg\max} f_k(x_{\mathcal{F}_\theta})$.

Prior literature on functional perturbations is surprisingly sparse. To our best knowledge, only one work involves a term *functional perturbations* [11], in which a functional threat model is proposed to produce adversarial examples by employing a single function to perturb all input features simultaneously. In contrast, we introduce a flexible and generalised functional threat model by removing the constraint of global uniform changes in images. Obviously, the additive threat model is a particular case of the functional threat model, when the perturbation function $\mathcal{F}_\theta$ manipulates pixels of an image by adding slight L_p bounded distortions.

Verification. The purpose of this paper is to verify the resilience of the classifier $f(\cdot)$ against perturbation functions $\mathcal{F}$ parameterised with $\theta \in \Theta$ while functional perturbations $\mathcal{F}_\theta$ would not change the oracle label from human perception if θ within parameter space Θ, or, more precisely, to provide guarantees that the classifier $f(\cdot)$ is probabilistically robust with regard to an input x when exposed to a particular functional perturbation $\mathcal{F}_\theta$. To this end, let k^* denote the ground truth class of the input sample. Assume that $\mathbb{S}_{\mathcal{F}}(x)$ is the space of all images $x_{\mathcal{F}_\theta}$ of x under perturbations induced by a perturbation function $\mathcal{F}_\theta$ and $\mathcal{P}$ is the probability measure on this space $\mathbb{S}_{\mathcal{F}}(x)$. This leads to the following robustness definitions:

Definition 1 (Deterministic robustness). *Let $\mathcal{F}_\theta$ be a specific perturbation function parametrized by θ, and Θ denotes a parameter space of a given perturbation function. Assume that $x_{\mathcal{F}_\theta} = \mathcal{F}_\theta(x)$ is the perturbed version of x given $\theta \in \Theta$, and $\mathbb{S}_{\mathcal{F}}(x)$ is the space of all images $x_{\mathcal{F}_\theta}$ of x under perturbation function $\mathcal{F}_\theta$. Given a K-class DNN f, an input x and a specific perturbation function $\mathcal{F}_\theta$ with $\theta \in \Theta$, we can say that f is deterministically robust w.r.t. the image x, i.e. x is correctly classified with probability one, if*

$$\underset{k \in \{1,...,K\}}{\arg\max} f_k(x_{\mathcal{F}}) = k^*, for \ all \ x_{\mathcal{F}} \sim \mathbb{S}_{\mathcal{F}}(x).$$

Definition 2 (Probabilistic Robustness). *Let $\mathcal{F}_\theta$ be a specific perturbation function parametrized by θ, and Θ denotes a parameter space of a given perturbation function. Assume that $x_{\mathcal{F}_\theta} = \mathcal{F}_\theta(x)$ is the perturbed version of x given*

$\theta \in \Theta$, and $\mathbb{S}_{\mathcal{F}}(x)$ is the space of all images $x_{\mathcal{F}_\theta}$ of x under perturbation function $\mathcal{F}_\theta$. Given a K-class DNN f, an input x, a specific perturbation function $\mathcal{F}_\theta$ with $\theta \in \Theta$, and a tolerated error rate τ, the K-class DNN f is said to be probabilistically robust with probability at least $1 - \delta$, if

$$\mathbb{P}_{x_\mathcal{F} \sim \mathbb{S}_\mathcal{F}(x)} \left(p \left(\underset{k \in \{1,...,K\}}{\arg\max} f_k(x_\mathcal{F}) \neq k^* \right) < \tau \right) \geq 1 - \delta. \tag{1}$$

Verifying deterministic robustness has been widely studied in the context of pixel-level additive perturbations and worst-case adversarial training; however, deterministic robustness is always too stringent to hold, and deterministic robustness and probabilistic robustness are "equivalent" to each other when we choose $\tau = 0$.

4   Verification of Probabilistic Robustness

We now present our proposed method, named PRoA, for verifying the probabilistic robustness of black-box classifiers against functional perturbations. A schematic overview of PRoA is illustrated in *Appendix A*.

4.1   Formulating Verification Problem

Our goal is to verify probabilistic robustness properties for a neural network classifier f, providing the classifier with probabilistic guarantees of its stability under functional perturbations. We formalise the robustness properties by examining substantial discrepancies of outputs w.r.t. input transformations [16]. Next, we describe how to formalise the robustness property using both original and perturbed images.

   We have a deterministic neural network $f : \mathbb{R}^n \rightarrow [0, 1]^K$. Assume that a given input x and its perturbed image $x_\mathcal{F}$ are assigned by f with the output probability vectors $\mathbf{p} = f(x)$ and $\mathbf{p}_\mathcal{F} = f(x_{\mathcal{F}_\theta})$, respectively. Let $k^* = \arg\max \mathbf{p}$ and $\tilde{k} = \arg\max \mathbf{p}_\mathcal{F}$ denote the output labels assigned to original image x and perturbed version $x_{\mathcal{F}_\theta}$ and $d = \frac{p_1 - p_2}{2}$ be the half of the difference between two largest components of $\mathbf{p}$.

   Then, the certain perturbations would not change the label, *i.e.* $\tilde{c} = c$, if

$$\|\mathbf{p} - \mathbf{p}_\mathcal{F}\|_\infty < d. \tag{2}$$

where $\|\mathbf{p} - \mathbf{p}_\mathcal{F}\|_\infty = \max(|\mathbf{p}_1 - \mathbf{p}_{\mathcal{F}_1}|, \ldots, |\mathbf{p}_K - \mathbf{p}_{\mathcal{F}K}|)$.

   That means, if the maximum change caused by functional perturbations amongst all classes w.r.t. the output probability vectors, does not exceed half of the maximum difference d between the two largest components of $\mathbf{p}$, the classifier will retain the category to an input x. Thus, it is straightforward to provide the probabilistic guarantees that the class label assigned to an input x by a classifier f would not change under the transformation functions $\mathcal{F}_\theta$ by bounding the probability of the event $\|\mathbf{p} - \mathbf{p}_\mathcal{F}\|_\infty < d$ occurring.

Subsequently, we suggest applying adaptive concentration inequalities, which enable our algorithm iteratively to take more and more samples until the estimated probability of event occurrence is sufficiently accurate to be used to compute the probability satisfying Eq. (2). We establish some notation for the verification process that follows. For a random variable $Z \sim P_Z$ following any probability distribution P_Z, $\mu_Z = \mathbb{E}_{Z \sim P_Z}[Z]$ donates the expectation of Z. To fit the context of probabilistic robustness verification, we let

$$Z = \mathbb{1}\left[\|\mathbf{p} - \mathbf{p}_{\mathcal{F}}\|_{\infty} < d\right] \tag{3}$$

where $\mathbb{1}[x]$ is an indicator function that returns 1 if x is true and 0 otherwise. In this case, μ_Z represents certified stable probability of a data instance x under functional transformations $\mathcal{F}_{\theta}$ parameterised by θ, i.e.,

$$\mu_Z = \mathbb{E}_{Z \sim P_Z}[Z] = P_{Z \sim P_Z}\left[Z = 1\right]. \tag{4}$$

4.2   Adaptive Concentration Inequalities

Concentration inequalities [2], e.g. Chernoff inequality, Azuma's bound and Hoeffding's inequality, are fundamental statistical analytic techniques, widely applied to reliable decision-making with probabilistic guarantees. Hoeffding inequality is utilised to bound the probability of an event or the sum of bounded variables.

Let Z be a random variable with distribution P_Z, and $Z_1, Z_2, \ldots, Z_n$ are independent and identically distributed samples drawn from P_Z, then we can estimate μ_Z, which represents the expected value of Z using

$$\hat{\mu}_Z = \frac{1}{n}\sum_{i=1}^{n} Z_i. \tag{5}$$

Note that, regardless of the number of samples used, there must be some error ϵ between the estimated value $\hat{\mu}_Z$ and true expected value μ_Z. However, we can derive high-probability bounds on this error using Hoeffding inequality [4].

Definition 3 (Hoeffding Inequality [4]). *For any $\delta > 0$,*

$$\mathrm{P}_{Z_1,\ldots,Z_n \sim P_Z}\left[|\hat{\mu}_Z - \mu_Z| \leq \varepsilon\right] \geq 1 - \delta \tag{6}$$

holds for $\delta = 2e^{-2n\varepsilon^2}$, equivalently, $\varepsilon = \sqrt{\frac{1}{2n}\log\frac{2}{\delta}}$.

The number of samples n, on the other hand, must be independent of the underlying process and determined in advance, yet in most circumstances, we generally have no idea how many samples we will need to validate the robustness specification. Consequently, we would like the number of samples used during the verification procedure to be a random variable. We decide to incorporate *adaptive concentration inequality* into our algorithm, enabling our verification algorithm to take samples iteratively. Upon termination, n becomes a stopping

time J, where J is a random variable, depending on the ongoing process. Then, the following adaptive Hoeffding inequality is utilised to guarantee the bound of the aforementioned probability since traditional concentration inequalities do not hold when the number of samples is stochastic.

Theorem 1 (Adaptive Hoeffding Inequality [30]). *Let Z_i be $1/2$-subgaussian random variables, and let $\hat{\mu}_Z^{(n)} = \frac{1}{n} \sum_{i=1}^{n} Z_i$, also let J be a random variable on $\mathbb{N} \cup \{\infty\}$ and let $\varepsilon(n) = \sqrt{\frac{a \log(\log_c n + 1) + b}{n}}$ where $c > 1, a > c/2, b > 0$, and ζ is the Riemann-ζ function. Then, we have*

$$\mathrm{P}\left[J < \infty \wedge \left(\left| \hat{\mu}_Z^{(J)} \right| \geq \varepsilon(J) \right) \right] \leq \delta_b \tag{7}$$

where $\delta_b = \zeta(2a/c)e^{-2b/c}$.

4.3   Verification Algorithm

In this section, we will describe how to verify the probabilistic robustness of a given classifier, deriving from adaptive Hoeffding inequality. To begin, we can derive a corollary from Theorem 1. Note that the values of a and c do not have a significant effect on the quality of the bound in practice [30] and we fix a and c with the recommended values in [30], 0.6 and 1.1, respectively.

Theorem 2. *Given a random variable Z as shown in Eq. (3) with unknown probability distribution P_Z, let $\{Z_i \sim P_{\mathcal{Z}}\}_{i \in \mathbb{N}}$ be independent and identically distributed samples of Z. Let $\hat{\mu}_Z^{(n)} = \frac{1}{n} \sum_{i=1}^{n} Z_i$ be estimate of true value μ_Z, and let stopping time J be a random variable on $\mathbb{N} \cup \{\infty\}$ such that $P[J < \infty] = 1$.*
Then, for a given $\delta \in \mathbb{R}_+$,

$$\mathrm{P}\left[\left| \hat{\mu}_Z^{(J)} - \mu_Z \right| \leq \varepsilon(\delta, J) \right] \geq 1 - \delta \tag{8}$$

holds, where $\varepsilon(\delta, n) = \sqrt{\frac{0.6 \cdot \log(\log_{1.1} n + 1) + 1.8^{-1} \cdot \log(24/\delta)}{n}}$.

We give a proof in *Appendix B*.

In the context of probabilistic robustness verification, we can certify the probabilistic robustness of a black-box neural network against functional threat models. Specifically, certified probability, μ_Z, is calculated by computing the proportion of the event $(Z < d)$ occurring through sampling the perturbed images surrounding an input x. For example, given a target neural network, we would like to verify whether there are at most τ (e.g. 1%) adversarial examples within a specific neighbouring area around an image x with greater than $1 - \delta$ (e.g. 99.9%) confidence. This means we would like to have more than 99.9% confidence in asserting that the proportion of the adversarial examples is fewer than 1%.

Building upon this idea, the key of this statistical robustness verification is to prove the robustness specification of form $\mu_Z \geq 1 - \tau$ holds. If μ_Z is quite

close to $1 - \tau$, then more additional samples are required to make ϵ to be small enough to ensure that $\hat{\mu}_Z$ is close to μ_Z. We use a hypothesis test parameterized by a given modest probability τ of accepted violation predefined by users.

- $\circ$ $\mathcal{H}_0$: The probability of robustness satisfaction $\mu \geq 1 - \tau$. Thus, the classifier can be certified.
- $\circ$ $\mathcal{H}_1$: The probability of robustness satisfaction $\mu < 1 - \tau$. Thus, the classifier should not be certified.

Alternatively, consider the hypothesis testing with two following conditions

$$
\begin{aligned}
\mathcal{H}_0 &: \hat{\mu}_Z + \tau - \epsilon - 1 \geq 0 \\
\mathcal{H}_1 &: \hat{\mu}_Z + \tau + \epsilon - 1 < 0.
\end{aligned}
\tag{9}
$$

If H_0 holds, then together with $\mathrm{P}\left[\left|\hat{\mu}_Z^{(J)} - \mu_Z\right| \leq \epsilon\right] \geq 1 - \delta$, we can assert that $\mu_Z \geq \hat{\mu}_Z - \epsilon \geq 1 - \tau$ with high confidence. Likewise, we can conclude that $\mu_Z \leq \hat{\mu}_Z + \epsilon < 1 - \tau$, if H_1 holds.

The full algorithm is summarized in Algorithm 1 in *Appendix A*.

5   Experiments

In order to evaluate the proposed method, an assessment is conducted involving various trained neural networks on public data sets CIFAR-10 and ImageNet.

Specifically, for neural networks certified on the CIFAR-10 dataset, we have trained three neural networks based on ResNet18 architecture: a naturally trained network (plain), an adversarial trained network augmented with adversarial examples generated by l_2 PGD attack (AT), and a perceptual adversarial trained network (PAT) against a perceptual attack [12]. In addition, four state-of-the-art neural networks, *i.e.* resnetv2_50, mobilenetv2_100, efficientnet_b0 and vit_base_patch16_244 are introduced for ImageNet dataset; all pre-trained models are available on a PyTorch library. For our models, selected details are described in Table 6 in *Appendix C.1*.

We provide the details about considered functional perturbations in the following subsection, and the results follow. *Nota bene*, we choose $\tau = 5\%$ for certifying the robustness of all models, as this is a widely accepted level in most practice. All the experiments are run on a desktop computer (i7-10700K CPU, GeForce RTX 3090 GPU).

5.1   Baseline setting

To demonstrate the effectiveness and efficiency of PRoA[1], it is natural to compare the estimated probability of the event, *i.e.* a target model will not fail when encountering functional perturbations, obtained by PRoA with the lower limit

---

[1] Our code is released via `https://github.com/TrustAI/PRoA`.

Table 1. Comparison with related work in different aspects.

	SRC [5]	AMLS based [24]	Randomized Smoothing [3, 28]	DeepGo [18]	Reachability based [26]	Semantify-NN [15]	FVIM based [27]	CROWN [29, 25, 23]	PRoA
Deterministic Robustness	✗	✗	✗	✓	✓	✓	✓	✓	✗
Probabilistic Robustness	✓	✓	✓	✗	✗	✗	✗	✗	✓
Verifying Robustness on Functional Perturbation	✗	✗	✗	✗	✗	✗	✗	✗	✓
Black-box Model	✓	✓	✓	✓	✗	✓	✗	✗	✓

of the corresponding confidence interval, *i.e.* Agresti–Coull confidence interval (A-C CI), see *Appendix C.2.*

We list the relevant existing works in Table 1 and compare our method with these typical methods from five aspects. Specifically, DeepGo [18], Reachability based [26], Semantify-NN [15], FVIM based [27] and CROWN [29, 25, 23] only can evaluate deterministic robustness of neural networks. Although SRC [5], AMLS based [24], Randomized Smoothing [3, 28] are able to certify probabilistic robustness, our work extensively consider models' probabilistic robustness under functional threat models.

To the best of our knowledge, there is no existing study in terms of certifying the probabilistic robustness of neural networks involving a functional threat model. Since [5] is the closest approach in spirit to our method amongst recent works, we use SRC [5] as our baseline algorithm. The proposal of SRC is to measure the probabilistic robustness of neural networks by finding the maximum perturbation radius using random sampling, and we extend it to be a baseline algorithm for computing the certified accuracy under functional perturbations.

5.2   Considered Functional Perturbations

PRoA is a general framework that is able to assess the robustness under any functional perturbations. In our experiments, we specifically study geometric transformation, colour-shifted function, and Gaussian blur in terms of verifying probabilistic robustness.

Gaussian Blur. Gaussian blurring is used to blur an image in order to reduce image noise and detail involving a Gaussian function

$$G_{\theta_g}(k) = \frac{1}{\sqrt{2\pi\theta_g}} \exp\left(-k^2/(2\theta_g)\right) \tag{10}$$

where θ_g is the squared kernel radius. For $x \in \mathcal{X}$, we define

$$\mathcal{F}_G(x) = x * G_{\theta_g} \tag{11}$$

as the corresponding function parameterised by θ_g where $*$ denotes the convolution operator.

Geometric Transformation. For geometric transformation, we consider three basic geometric transformations: rotation, translation and scaling. We implement the corresponding geometric functions in a unified manner using a spatial

transformer block with a set of parameters of affine transformation, *i.e.* $\mathcal{T}(x, \theta)$, in [7], where

$$\theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \tag{12}$$

is an affine matrix determined by θ_r, θ_t as well as θ_s.

Colour-Shifted Function. Regarding colour shifting, we change the colour of images based on HSB (Hue, Saturation and Brightness) space instead of RGB space since HSB give us a more intuitive and semantic sense for understanding the perceptual effect of the colour transformation. We also consider a combination attack using brightness and contrast.

All mathematical expressions of these functional perturbations as well as their parameter ranges are presented in *Appendix C.3*.

5.3   Quantitative Results of Experiments on CIFAR-10

To evaluate our method, we calculate the probabilistically certified accuracy of 1,000 images randomly from a test set for various functional perturbations by PRoA and SRC, in dependence on the user-defined confidence level. Furthermore, empirical robust accuracy is computed against random and grid search adversaries as well.

To begin, we validate the effectiveness of our method over three ResNet18 models (plain, AT and PAT) trained with different training protocols against all considered functional perturbations on the CIFAR-10 dataset as mentioned previously. As a result of the experiments, we present considered perturbation functions, accompanying parameters, and quantitative results in terms of probabilistically certified accuracy (Cert. Acc), empirical accuracy (Rand.) and empirical robust accuracy (Grid) in Table 2. Clearly, the results of the proposed method align well with the validation results obtained by exhaustive search and random perturbation, and PRoA is able to achieve higher certified accuracy than SRC in almost all scenarios. Thus, the effectiveness of PRoA can be demonstrated.

An illustration of model verification using A-C CI, SRC and PRoA with various confidence levels ($90\% \sim 1 - 10^{-30}$) against the picture scaling function on CIFAR-10 is depicted in Fig. 1. For instance, according to Fig. 1(a), we have 90% confidence ($\delta = 10^{-1}$) that this considered trained model will correctly identify roughly 71 percent of images in CIFAR-10 after a no more than 30% image scaling with a greater than 95% chance ($\tau = 5\%$). In contrast, we have $1 - 10^{-30}$ ($\delta = 10^{-30}$) confidence that the proportion of images with a misclassification probability below our accepted level 5% would be 67%. Clearly, accuracy certified by PRoA reduces along with the growth of confidence, but it is not significantly changed for SRC and progressively diminishes for A-C CI. In addition, as compared with baselines, the proposed method achieves remarkable higher certified robust accuracy and a narrower gap to empirical robust accuracy along with our confidence increasing, see Fig. 1. Moreover, Grid is an approximated accuracy to the extreme case with zero tolerance ($\tau = 0$) to perturbations. However, certified accuracy with a 5% tolerance level obtained by SRC and A-C

Table 2. CIFAR-10 - Comparison of empirical robust accuracy (Grid and Rand.) and probabilistically certified accuracy (Cert. Acc) with respect to a specific model (ResNet18) with three training methods, shown in Table 6. Moreover, probabilistically certified accuracy is presented with three confidence levels $(1-\delta)$, low level of confidence $(\delta = 10^{-4})$, middle level of confidence $(\delta = 10^{-15})$ and high level of confidence $(\delta = 10^{-30})$, respectively.

Transformation	Parameters	Training type	Grid	Rand.	SRC Cert. Acc			PRoA Cert. Acc		
					$\delta = 10^{-30}$	$\delta = 10^{-15}$	$\delta = 10^{-4}$	$\delta = 10^{-30}$	$\delta = 10^{-15}$	$\delta = 10^{-4}$
Rotation	$\theta_r \in [-35°, 35°]$	plain	26.9%	76.8%	24.7%	24.8%	24.8%	**30.3%**	**31.5%**	**32.0%**
		PAT	16.7%	55.9%	8.1%	8.1%	8.1%	**10.8%**	**12.4%**	**12.9%**
		AT	16.5%	74.5%	11.2%	11.2%	11.2%	**14.7%**	**15.2%**	**15.4%**
Translation	$\theta_t \in [-30\%, 30\%]$	plain	62.8%	89.6%	64.9%	65.1%	66.6%	**77.5%**	**78.8%**	**79.4%**
		PAT	50.1%	77.7%	31.1%	31.7%	32.4%	**47.5%**	**48.6%**	**49.5%**
		AT	56.5%	79.3%	45.3%	45.7%	46.1%	**58.9%**	**60.1%**	**61.7%**
Scale	$\theta_s \in [-70\%, 130\%]$	plain	45.4 %	86.9%	48.7%	49.0%	49.7%	**63.3%**	**65.2%**	**67.1%**
		PAT	23.5%	73.1%	8.4%	8.7%	9.6%	**20.9%**	**22.8%**	**24.7%**
		AT	34.4%	74.4%	19.2%	19.4%	20.3%	**32.4%**	**34.5%**	**35.9%**
Hue	$\theta_h \in [-\frac{\pi}{2}, \frac{\pi}{2}]$	plain	76.9 %	89.9%	75.0%	75.0%	75.0%	**79.5%**	**79.8%**	**79.6%**
		PAT	63.0%	77.5%	53.6%	53.6%	53.6%	**56.7%**	**57.6%**	**57.9%**
		AT	57.6%	55.1%	**54.7%**	**54.7%**	54.7%	54.1%	54.5%	**55.7%**
Saturation	$\theta_s \in [-30\%, 30\%]$	plain	92.3 %	93.9%	95.3%	95.3%	95.3%	**95.6%**	**95.6%**	**96.4%**
		PAT	77.1%	80.8%	72.3%	72.3%	72.4%	**75.6%**	**76.0%**	**77.3%**
		AT	74.5%	76.4%	76.8%	77.0%	77.0%	**79.4%**	**79.6%**	**80.3%**
Brightness+Contrast	$\theta_b \in [-30\%, 30\%]$ $\theta_c \in [-30\%, 30\%]$	plain	72.6 %	92.7%	75.5%	75.7%	76.2%	**83.8%**	**84.8%**	**84.1%**
		PAT	36.1%	76.2%	20.5%	20.9%	21.6%	**37.4%**	**38.2%**	**37.0%**
		AT	31.5%	73.9%	17.8%	17.9%	18.4%	**34.7%**	**38.1%**	**35.7%**
Gaussian Blurring	$\theta_g \in [0, 9]$	plain	1.0%	18.1%	3.1%	3.1%	3.3%	**3.6%**	**3.7%**	**3.4%**
		PAT	2.9%	39.7%	11.0%	11.0%	11.0%	**13.5%**	**13.7%**	**12.9%**
		AT	3.7%	42.9%	18.7%	18.9%	**18.9%**	**19.2%**	**19.3%**	18.8%

CI always tends to be below the Grid without tolerance as the confidence level increases, which causes underestimation of the probabilistic robustness.

(a) plain   (b) PAT   (c) AT
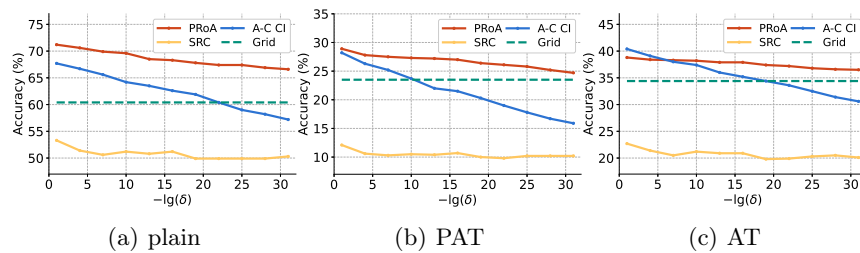
Fig. 1. CIFAR-10 - An illustration of evaluating robustness of trained neural networks using PRoA, SRC and A-C CI with different confidence parameter δ against one specific perturbation, image scaling.

We apply SRC and PRoA for verifying the robustness of 500 images, which are randomly chosen from the test set on CIFAR-10. The corresponding confusion matrix is shown in Table 3, which takes into account the cases in which the SRC outputs an "infeasible" status when it fails to obtain a deterministic certification result, and PRoA reaches sample limitation (set to 10,000) as a termination condition. Unsurprisingly, our method can take a certification decision in most cases when SRC returns an "infeasible", even though 14 images obtain a "termination" status due to adaptive sampling reaching sample limitation.
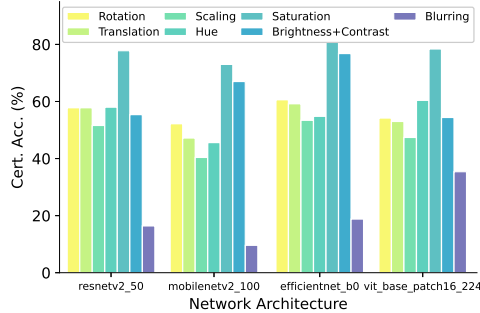
Fig. 2. ImageNet - Comparison of probabilistic certified accuracy with confidence level $1 - 10^{-4}$, computed over the 500 randomly selected ImageNet images, amongst the models described in Table 6.

Table 3. CIFAR-10 - Confusion matrix comparing SRC and PRoA [plain model under brightness+contrast perturbation, $\delta = 10^{-10}$].

		PRoA		
		Certified	Uncertified	Termination
	Certified	329	32	12
SRC	Uncertified	36	9	0
	Infeasible	69	11	2

5.4 Comparing probabilistic robustness across models on ImageNet

We also use our method to analyse four large state-of-the-art neural networks against perturbation functions as mentioned earlier with 500 images randomly picked from the ImageNet test set. Fig. 2 demonstrates the robustness comparison of different models when subjected to diverse functional perturbations. All validation results of different models are shown as percentages in Fig. 2. For the 'rotation' scenario, the certified accuracy of resnetv2_50 produced by PRoA is 57.8%, which means we have 99.99% confidence in the claim that on average, in resnetv2_50, 57.8% of images will produce an adversarial example with a chance of more than 5% in the 'rotation' scenario, e.g. camera rotation.

We also compare our algorithm to the Agresti–Coull confidence interval and SRC with a moderate confidence level, *i.e.* $\delta = 10^{-10}$, as shown in Table 4. On the one hand, our method provides the highest certified accuracy for practically all scenarios and models; on the other hand, the average runtime of our method is comparatively longer than baselines, due to the error bounds of the estimate, which are not tight enough to make decisions and necessitate more samples. Interestingly, our algorithm takes the shortest time to certify images under a sophisticated functional perturbation, the Gaussian blurring, whereas the computation time of A-C CI and SRC increases. This is because, instead

of a predetermined and decided a priori number of samples, our method terminates at any runtime J depending on the ongoing process once it is capable of delivering a result, avoiding superfluous samples.

Finally, the average number of samples and the average runtime for a single image are reported in Table 5. As one can notice, our method can be easily scaled to various SOTA network architectures, and the computation time and required samples increase reasonably with network size and complexity of perturbation function.

Table 4. ImageNet - Comparison Agresti–Coull, SRC and PRoA [$\delta = 10^{-10}$].

Model	Perturbation	Certified (%)			Avg. runtime (sec.)		
		Agresti–Coull	SRC	PRoA	Agresti–Coull	SRC	PRoA
Mobilenetv2_100	Rotation	38	40	**43**	**5.08**	5.10	8.35
	Translation	41	34	**47**	**5.14**	5.20	8.96
	Scaling	38	30	**44**	5.32	**5.07**	8.64
	Hue	40	**48**	**48**	5.64	**5.16**	5.19
	Saturation	65	71	**72**	5.60	**5.16**	7.26
	Brightness+Contrast	47	54	**62**	5.58	**5.17**	7.03
	Gaussian Blurring	3	6	**8**	6.38	5.72	**3.89**
efficientnet_b0	Rotation	46	47	**49**	**5.08**	6.25	6.28
	Translation	49	44	**57**	**5.14**	6.24	7.77
	Scaling	46	44	**51**	**5.32**	6.25	9.83
	Hue	48	55	**57**	**5.64**	6.53	8.69
	Saturation	73	79	**81**	**5.60**	6.53	9.83
	Brightness+Contrast	55	56	**65**	**5.59**	6.53	12.37
	Gaussian Blurring	10	14	**17**	6.38	7.11	**5.61**
Resnetv2_50	Rotation	51	46	**54**	12.77	**9.68**	15.76
	Translation	58	44	**57**	12.80	**9.56**	18.88
	Scaling	51	38	**54**	12.80	**9.54**	17.89
	Hue	61	61	**63**	13.10	**9.81**	13.04
	Saturation	77	83	**86**	13.15	**9.82**	16.87
	Brightness+Contrast	39	32	**40**	14.02	**9.81**	20.51
	Gaussian Blurring	15	14	**17**	14.16	10.43	**6.66**
vit_base_patch16_224	Rotation	39	34	**41**	34.68	**33.04**	49.62
	Translation	47	32	**49**	34.43	**33.06**	59.18
	Scaling	40	33	**43**	34.32	**33.00**	63.21
	Hue	**63**	53	54	34.61	**33.31**	45.70
	Saturation	70	71	**73**	37.54	**33.31**	41.37
	Brightness+Contrast	32	24	**34**	36.83	**34.19**	70.69
	Gaussian Blurring	**32**	28	30	35.99	34.88	**33.96**

6   Conclusion

This paper aims to certify the probabilistic robustness of a target neural network to a functional threat model with an adaptive process inspired by the Adaptive Concentration Inequalities. With PRoA, we can certify that a trained neural network is robust if the estimated probability of the failure is within a tolerance level. PRoA is dependent on the ongoing hypothesis test, avoiding a-prior sample size. The tool is scalable, efficient and generic to black-box classifiers, and it also comes with provable guarantees. In this paper, the hypothesis testing and adaptive sampling procedure are sequential and bring difficulty for parallelization, so one interesting future direction lies in how to further boost PRoA's efficiency,

Table 5. ImageNet - PRoA $[\delta = 10^{-10}]$.

Model	Perturbation	Avg. runtime (sec.$\pm std$)	Avg. sample num.	Certified (%)
resnetv2_50	Rotation	15.76±17.74	5820	54
	Brightness+Contrast	20.51±33.05	7930	40
	Blurring	6.66±9.64	2420	17
mobilenetv2_100	Rotation	8.35±13.40	7860	43
	Brightness+Contrast	7.03±8.23	6650	62
	Blurring	3.88±10.55	3180	6
efficientnet_b0	Rotation	6.28±7.28	4970	49
	Brightness+Contrast	12.37±13.81	9370	65
	Blurring	5.61±2.50	3790	17
vit_base_patch16_224	Rotation	49.62±80.11	7260	41
	Brightness+Contrast	70.69±106.84	9950	34
	Blurring	35.96±69.26	5020	30

e.g., by enabling parallelization on GPUs. Another interesting future work is to bridge the gap between worst-case certification and chance-case certification.

References

1. Amir, G., Wu, H., Barrett, C., Katz, G.: An smt-based approach for verifying binarized neural networks. In: TACAS. pp. 203—222. Springer, Luxembourg (2021)
2. Boucheron, S., Lugosi, G., Massart, P.: Concentration inequalities: A nonasymptotic theory of independence. 1st edn. Oxford university press, Oxford (2013)
3. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: ICML. pp. 1310—1320. PMLR, California (2019)
4. Hoeffding, W.: Probability inequalities for sums of bounded random variables. In: The collected works of Wassily Hoeffding, pp. 409–426. Springer (1994)
5. Huang, C., Hu, Z., Huang, X., Pei, K.: Statistical certification of acceptable robustness for neural networks. In: ICANN. pp. 79—90. Springer, Bratislava (2021)
6. ISO, I.: Iso/iec guide 51: Safety aspects-guidelines for their inclusion in standards. Geneva, Switzerland (2014)
7. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NeurIPS. pp. 2017—2025. The MIT Press, Quebec (2015)
8. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: a calculus for reasoning about deep neural networks. Formal Methods in System Design, pp. 1—30 (2021)
9. Katz, G., Barrett, C.W., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient SMT solver for verifying deep neural networks. In: CAV. pp. 97—117. Springer, Heidelberg (2017)
10. Kurakin, A., Goodfellow, I., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., Xie, C., et al.: Adversarial attacks and defences competition. In: The NIPS'17 Competition: Building Intelligent Systems, pp. 195—231. Springer, Cham (2018)
11. Laidlaw, C., Feizi, S.: Functional adversarial attacks. In: NeurIPS. pp. 10408—10418. The MIT Press, Vancouver (2019)
12. Laidlaw, C., Singla, S., Feizi, S.: Perceptual adversarial robustness: Defense against unseen threat models. In: ICLR. Austria (2021)

13. Li, J., Liu, J., Yang, P., Chen, L., Huang, X., Zhang, L.: Analyzing deep neural networks with symbolic propagation: Towards higher precision and faster verification. In: SAS. pp. 296—319. Springer, Porto (2019)
14. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR. Vancouver (2018)
15. Mohapatra, J., Weng, T.W., Chen, P.Y., Liu, S., Daniel, L.: Towards verifying robustness of neural networks against a family of semantic perturbations. In: CVPR. pp. 244—252. IEEE, Seattle (2020)
16. Pautov, M., Tursynbek, N., Munkhoeva, M., Muravev, N., Petiushko, A., Oseledets, I.: Cc-cert: A probabilistic approach to certify general robustness of neural networks. arXiv preprint. arXiv:2109.10696 (2021)
17. Rivasplata, O.: Subgaussian random variables: An expository note (2012)
18. Ruan, W., Huang, X., Kwiatkowska, M.: Reachability analysis of deep neural networks with provable guarantees. In: IJCAI. pp. 2651—2659. IJCAI.org, Stockholm (2018)
19. Singh, G., Ganvir, R., Püschel, M., Vechev, M.T.: Beyond the single neuron convex barrier for neural network certification. In: NeurIPS. pp. 15072—15083. The MIT Press, Vancouver (2019)
20. Singh, G., Gehr, T., Püschel, M., Vechev, M.T.: An abstract domain for certifying neural networks. Proc. ACM Program. Lang. **3**(POPL), 1—30 (2019)
21. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: ICLR. Banff (2014)
22. Tran, H.D., Yang, X., Manzanas Lopez, D., Musau, P., Nguyen, L.V., Xiang, W., Bak, S., Johnson, T.T.: Nnv: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In: CAV. pp. 3—17. Springer, Los Angeles (2020)
23. Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.J., Kolter, J.Z.: Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In: NeurIPS. The MIT Press, Virtual (2021)
24. Webb, S., Rainforth, T., Teh, Y.W., Kumar, M.P.: A statistical approach to assessing neural network robustness. In: ICLR. New Orleans (2019)
25. Weng, L., Zhang, H., Chen, H., Song, Z., Hsieh, C.J., Daniel, L., Boning, D., Dhillon, I.: Towards fast computation of certified robustness for relu networks. In: ICML. pp. 5276—5285. PMLR, Stockholm (2018)
26. Xiang, W., Tran, H.D., Johnson, T.T.: Output reachable set estimation and verification for multilayer neural networks. IEEE transactions on neural networks and learning systems 29(11), 5777—5783 (2018)
27. Yang, X., Johnson, T.T., Tran, H.D., Yamaguchi, T., Hoxha, B., Prokhorov, D.V.: Reachability analysis of deep relu neural networks using facet-vertex incidence. In: HSCC. pp. 18:1—18:7. ACM, Nashville (2021)
28. Zhang, D., Ye, M., Gong, C., Zhu, Z., Liu, Q.: Black-box certification with randomized smoothing: A functional optimization based framework. In: NeurIPS. The MIT Press, Virtual (2020)
29. Zhang, H., Weng, T.W., Chen, P.Y., Hsieh, C.J., Daniel, L.: Efficient neural network robustness certification with general activation functions. In: NeurIPS. The MIT Press, Montréal (2018)
30. Zhao, S., Zhou, E., Sabharwal, A., Ermon, S.: Adaptive concentration inequalities for sequential decision problems. In: NeurIPS, pp. 1343–1351. The MIT Press, Barcelona (2016)