
Probably Approximately Global Robustness Certification

Peter Blohm¹ Patrick Indri¹ Thomas Gärtner¹ Sagar Malhotra¹

Abstract

We propose and investigate probabilistic guarantees for the adversarial robustness of classification algorithms. While traditional formal verification approaches for robustness are intractable and sampling-based approaches do not provide formal guarantees, our approach is able to efficiently certify a probabilistic relaxation of robustness. The key idea is to sample an ϵ -net and invoke a local robustness oracle on the sample. Remarkably, the size of the sample needed to achieve probably approximately global robustness guarantees is independent of the input dimensionality, the number of classes, and the learning algorithm itself. Our approach can, therefore, be applied even to large neural networks that are beyond the scope of traditional formal verification. Experiments empirically confirm that it characterizes robustness better than state-of-the-art sampling-based approaches and scales better than formal methods.

1. Introduction

We present a novel sampling-based procedure to certify the *global robustness* of a classifier with high probability. Existing robustness verification methods are decision procedures that provide yes/no answers for a given robustness criterion, for a specific point in the input space. In general, however, robustness is a quantitative property that may depend on additional factors, such as the classifier’s confidence in a given prediction. As a key contribution, we propose and investigate a novel notion of global robustness that quantifies the robustness of any point, given its *prediction confidence*. Our guarantee is obtained by checking local robustness on a *sufficiently large* sample of points. The size of the sample is quantified using a learning-theoretic construct, namely ϵ -nets (Haussler & Welzl, 1986). Our approach is agnostic to the specific method used to perform local robustness checks and, therefore, to the precise notion of robustness investigated. In this paper, we will consider both formal and

adversarial methods to check local robustness.

Neural Network (NN) robustness is a major desideratum as, for non-robust networks, predictions can be drastically changed with only small perturbations to the input (Szegedy et al., 2014). This behavior can be detrimental in safety-critical applications like autonomous driving (Rao & Fruinikj, 2018) or image recognition tasks (Athalye et al., 2018). Hence, methods for evaluating whether NNs are robust to such perturbations are critical to enable their safe and reliable deployment. Significant research efforts have been invested to assess NN robustness, following two main lines of research:

- *Formal verification* methods have been used to provably certify local robustness of NNs (Katz et al., 2019; Wu et al., 2024; Xu et al., 2021). Recent results have extended these techniques to global robustness certification (Athavale et al., 2024) albeit only for NNs with up to a few hundred parameters.
- *Adversarial* methods rely on local optimization to assess whether NNs can withstand adversarial attacks (Goodfellow et al., 2015; Madry et al., 2018; Carlini & Wagner, 2017a). These methods easily scale to large networks, but generally do not provide formal guarantees on the global behavior of the network.

Our approach provides a quantitative characterization of robustness for the whole input space parameterized by the prediction confidence of each point. That is, for each point in the input space we are able to give a high-probability lower bound for its robustness. This only requires independently and identically distributed (iid) samples from the data distribution and access to a local robustness oracle. Such an oracle can be efficiently modeled using existing (formal or adversarial) methods for assessing local robustness. The sample size required by our approach is independent of the input dimensionality, the number of classes, and the learning algorithm itself. Our approach can provide distinct robustness guarantees *for each* confidence value after a *single* sampling procedure. This characterization can be used to infer high-probability lower bounds on the robustness of new data points, that were not part of the sampling procedure.

This paper is organized as follows. In Section 2, we present the relevant related work. In Section 3, we introduce the

¹TU Wien, Austria. Correspondence to: Peter Blohm <peter.blohm@tuwien.ac.at>.

preliminaries. Our main contributions are described in [Section 4](#) and [Section 5](#). We present and discuss our experimental evaluation in [Section 6](#). Finally, [Section 7](#) contains concluding remarks.

2. Related Work

After the seminal work by [Szegedy et al. \(2014\)](#) showed that NNs are sensitive to adversarial examples, a number of techniques to find such examples were introduced. Many such approaches rely on gradient computation such as the Fast Gradient Signed Method (FGSM, [Goodfellow et al., 2015](#)) and an iterative adaptation of it called Projected Gradient Descent (PGD, [Madry et al., 2018](#)). Additionally, the C&W attack ([Carlini & Wagner, 2017a](#)) explicitly takes the distance to the original data point into account, to find a particularly close adversarial example. More recently, empirical approaches to assess ([Webb et al., 2019; Baluta et al., 2021; Kim et al., 2023](#)) as well as improve ([Li et al., 2023; Kim et al., 2023](#)) the robustness of NNs have been introduced, often building from the concept of adversarial training introduced by [Goodfellow et al. \(2015\)](#). While these approaches are applicable to large NNs and can be used to empirically assess robustness, they do not provide theoretical robustness guarantees for new points.

In an effort to obtain provable guarantees, a line of research has developed formal methods for the verification of NNs ([Katz et al., 2019; Chen et al., 2021](#)). While a formally verified NN is provably robust to input perturbations ([Casadio et al., 2022; Meng et al., 2022b](#)), formal verification is limited to small NNs. In fact, it is generally hard to provide guarantees about the behavior of large networks ([Katz et al., 2017](#)) and it is, in particular, hard to detect *all* adversarial examples ([Carlini & Wagner, 2017b](#)). Other approaches have sought to provide *approximate* ([Wu et al., 2020](#)) or *statistical* ([Webb et al., 2019; Cohen et al., 2019; Lecuyer et al., 2019](#)) robustness guarantees, as well as to train NNs which are certifiably robust ([Li et al., 2023; Sinha et al., 2018; Wang et al., 2018](#)).

[Athavale et al. \(2024\)](#) have recently extended existing formal verification methods to certify global robustness. This, however, drastically limits the scope of their method to NNs with only hundreds of parameters. In this paper, we introduce a probabilistic relaxation of their notion of global robustness. We utilize ϵ -nets ([Haussler & Welzl, 1986](#)) to provide high-probability guarantees for this notion of robustness.

3. Preliminaries

In this section, we will introduce basic notation and the necessary background for our theory. We will further introduce and motivate the notion of a *robustness oracle* and *prediction confidence*. These abstract concepts will allow our

results to be easily adapted to different methods of assessing robustness, as shown in our experiments in [Section 6](#).

3.1. Basic Notation

Consider a classification task with n classes on a metric input space \mathcal{X} . We define a classifier as $f : \mathcal{X} \rightarrow \mathbb{R}^n$, where for any instance $\mathbf{x} \in \mathcal{X}$ the classifier returns a vector $f(\mathbf{x}) \in \mathbb{R}^n$. In this vector, the i -th component $f(\mathbf{x})_i$ represents the classifier's output for class i , like the output layer in a NN. We say that the predicted class for \mathbf{x} is $\text{class}(\mathbf{x}) = \arg \max_{i \in [1, n]} f(\mathbf{x})_i$. Furthermore, we say $\text{conf}_f(\mathbf{x}) \in \mathbb{R}$ is the *prediction confidence* of f for $\text{class}(\mathbf{x})$. In the rest of this paper, we will assume that for a class $\text{class}(\mathbf{x}) = c$ the confidence is given by the softmax function, i.e.,

$$\text{conf}_f(\mathbf{x}) = \frac{\exp(f(\mathbf{x})_c)}{\sum_{j \in [1, n]} \exp(f(\mathbf{x})_j)} \quad (1)$$

without restricting the generality of our statements. We assume that the data for the classification task follows a distribution \mathcal{D} over \mathcal{X} . We write $X \sim \mathcal{D}$ to indicate that X is a random variable drawn from \mathcal{D} , and $\mathbf{x} \sim \mathcal{D}$ to denote an observed realization of X . If clear from context, probabilities $\Pr_{\mathcal{D}}$ that are computed with respect to a distribution \mathcal{D} will be written as \Pr .

3.2. Robustness

We say a classifier f is *robust* around a point \mathbf{x} , if $\text{class}(f(\mathbf{x}))$ remains constant within some *neighborhood* $\mathcal{N}(\mathbf{x}) \subset \mathcal{X}$ of \mathbf{x} . In many applications, $\mathcal{N}(\mathbf{x})$ is chosen to be an L_∞ -ball ([Goodfellow et al., 2015; Madry et al., 2018](#)) or another norm-bounded ball ([Baluta et al., 2021; Carlini & Wagner, 2017a; Leino et al., 2021; Athavale et al., 2024](#)).

How robust f is around \mathbf{x} is assessed by a *robustness oracle* $\text{rob}_f(\mathbf{x})$. The robustness oracle $\text{rob}_f(\mathbf{x})$ returns the distance ρ to the closest counterexample $\mathbf{x}' \in \mathcal{N}(\mathbf{x})$ s.t. $\text{class}(f(\mathbf{x}')) \neq \text{class}(f(\mathbf{x}))$. Examples of robustness oracles rob_f include verification tools ([Katz et al., 2017; Meng et al., 2022a; Xu et al., 2021; Athavale et al., 2024](#)), which offer exact results at high computational costs. For such exhaustive search methods, we can state $\text{rob}_f(\mathbf{x}) = \min_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} \{ \|\mathbf{x} - \mathbf{x}'\| : \text{class}(f(\mathbf{x})) \neq \text{class}(f(\mathbf{x}')) \}$. In contrast, heuristic or adversarial methods ([Goodfellow et al., 2015; Madry et al., 2018; Carlini & Wagner, 2017a](#)) can efficiently find very close points with different class membership, which can be used as an upper bound for ρ . We will further discuss practically useful robustness oracles based on well-established methods in [Section 6](#).

The following definition captures this oracle-based notion of robustness around a given point.

Definition 3.1 (Local ρ -robustness). A classifier $f : \mathcal{X} \rightarrow \mathbb{R}^n$ is locally ρ -robust around \mathbf{x} according to the robustness

oracle \mathbf{rob}_f if

$$\mathbf{rob}_f(\mathbf{x}) \geq \rho. \quad (2)$$

Based on this definition, we need to define a notion of *global* robustness for a given classifier f , so we can guarantee the classifier will behave robustly regardless of the input. A natural definition of *global* ρ -robustness would require local ρ -robustness for every possible input. However, enforcing this for all inputs in dense spaces like \mathbb{R}^m would constrain the classifier to make constant predictions. To address this, a more useful notion of global robustness focuses on specific regions of interest, such as areas where the classifier's prediction confidence $\mathbf{conf}_f(\mathbf{x})$ is high (Leino et al., 2021; Athavale et al., 2024). More confident predictions require larger changes in the output of f to alter the predicted class, so we consequently expect points classified with high confidence to be more robust. The following definition formalizes global robustness, based on a confidence threshold κ .

Definition 3.2 (Global ρ - κ -robustness). A classifier $f : \mathcal{X} \rightarrow \mathbb{R}^n$ is *globally* ρ - κ -robust if it is locally ρ -robust for all $\mathbf{x} \in \mathcal{X}$ where the prediction confidence $\mathbf{conf}_f(\mathbf{x})$ is larger than κ .

$$\forall \mathbf{x} \in \mathcal{X} : \mathbf{conf}_f(\mathbf{x}) \geq \kappa \implies \mathbf{rob}_f(\mathbf{x}) \geq \rho. \quad (3)$$

Definition 3.2 is equivalent to the following statement, which states the absence of counterexamples:

$$\nexists \mathbf{x} \in \mathcal{X} : \mathbf{rob}_f(\mathbf{x}) < \rho \wedge \mathbf{conf}_f(\mathbf{x}) \geq \kappa. \quad (4)$$

Equation (4) highlights that a classifier is certified to be robust if no counterexample exists, that is, if no point with both high confidence and insufficient robustness is found by the oracle. However, certifying robustness in this manner is intractable in general (Katz et al., 2017). In this paper, we focus instead on bounding the probability of encountering counterexamples.

3.3. Epsilon Nets

Our robustness guarantees build on concepts from computational geometry and learning theory, and our definitions are based on Mitzenmacher & Upfal (2017).

Definition 3.3 (Range space). Let \mathcal{Y} be a (possibly infinite) set and \mathcal{R} a family of subsets of \mathcal{Y} called *ranges*. A range space is a tuple $(\mathcal{Y}, \mathcal{R})$.

Definition 3.4 (VC Dimension, Vapnik & Chervonenkis (2015)). Let $(\mathcal{Y}, \mathcal{R})$ be a range space. The Vapnik-Chervonenkis (VC) dimension of $(\mathcal{Y}, \mathcal{R})$ is the size of the largest finite set $S \subseteq \mathcal{Y}$ such that for every subset $T \subseteq S$, there exists a range $R \in \mathcal{R}$ satisfying $R \cap S = T$. If no such maximum exists, the VC dimension is infinite.

Definition 3.5 (ϵ -net, Haussler & Welzl (1986)). Let $(\mathcal{Y}, \mathcal{R})$ be a range space and \mathcal{D} be a probability distribution on \mathcal{Y} . A

(finite) set $N \subseteq \mathcal{Y}$ is an ϵ -net for \mathcal{Y} with respect to \mathcal{D} if, for every set $R \in \mathcal{R}$ such that $\Pr(\mathcal{D}(R)) \geq \epsilon$, the set R contains at least one point from N , i.e.,

$$\forall R \in \mathcal{R} : \Pr(\mathcal{D}(R)) \geq \epsilon \implies R \cap N \neq \emptyset. \quad (5)$$

ϵ -nets provide a general notion of coverage. Our approach for certifying PAG-robustness relies on the following theorem for constructing ϵ -nets from iid samples.

Theorem 3.6 (ϵ -nets from iid samples (Mitzenmacher & Upfal, 2017)). Let $(\mathcal{Y}, \mathcal{R})$ be a range space with VC dimension d and let $\mathcal{D}_{\mathcal{Y}}$ be a probability distribution on \mathcal{Y} . For any $0 < \delta, \epsilon \leq \frac{1}{2}$, an iid sample from $\mathcal{D}_{\mathcal{Y}}$ of size s is an ϵ -net for \mathcal{Y} with probability at least $1 - \delta$ if

$$s = \mathcal{O}\left(\frac{d}{\epsilon} \ln \frac{d}{\epsilon} + \frac{1}{\epsilon} \ln \frac{1}{\delta}\right) \quad (6)$$

While the big- \mathcal{O} notation of this result is convenient for theoretical asymptotic analyses, it neglects constant factors that may be significant in practice. In their proof, however, Mitzenmacher & Upfal (2017) show that **Theorem 3.6** always holds if s satisfies the following inequality:

$$s(\epsilon, \delta, d) = \min_{s \in \mathbb{N}} \left\{ s : s \geq \frac{2}{\epsilon} \left(\ln \left(\frac{2}{\delta} \right) + d \ln(2s) \right) \right\}. \quad (7)$$

To obtain sample complexities, we find $s(\epsilon, \delta, d)$ using numerical methods.

4. PAG Robustness

In this section, we present our main theoretical results. We show how to get high-probability certificates for the following relaxation of global ρ - κ -robustness (**Definition 3.2**).

Definition 4.1. Given a data-distribution \mathcal{D} , a classifier $f : \mathcal{X} \rightarrow \mathbb{R}^n$ is *approximately globally robust* according to a robustness oracle \mathbf{rob}_f if, for a given confidence κ , robustness ρ , and $X \sim \mathcal{D}$, we have that

$$\Pr(\mathbf{rob}_f(X) < \rho \mid \mathbf{conf}_f(X) \geq \kappa) < \epsilon. \quad (8)$$

where $0 < \epsilon < 1$ is a chosen parameter.

Our relaxation states that all κ -confident predictions of f will also be at least ρ -robust, with high probability. This probabilistic statement is inspired by the work of Athavale et al. (2024), where the confidence threshold is introduced as a mechanism for f to abstain from prediction. A key difference to their work is that we are able to decide the probabilistic statement for all values of (ρ, κ) simultaneously. For this, we devise on a sampling-based methodology using ϵ -nets to obtain the bound in **Equation (8)**.

We rephrase the left-hand side of **Equation (8)** as

$$\frac{\Pr(\mathbf{rob}_f(X) < \rho \wedge \mathbf{conf}_f(X) \geq \kappa)}{\Pr(\mathbf{conf}_f(X) \geq \kappa)} \quad (9)$$

to then separately provide a lower bound for the numerator (using [Proposition 4.2](#)) and an upper bound for the denominator (using [Lemma 4.3](#)). Both of these bounds can then be realized with the same iid sample. In the rest of this section, we introduce a set of novel definitions required for our results and then proceed with the bounds themselves.

4.1. Quality Space

An important aspect of our method is that our guarantees are independent of the properties of the input space and of the classifier f . This is because our guarantees only consider the two properties of confidence and robustness of a given point in the input space. In this section, we formalize this idea. We define a function q that maps a given point $\mathbf{x} \in \mathcal{X}$ to its robustness-confidence tuple:

$$q(\mathbf{x}) \mapsto (\text{rob}_f(\mathbf{x}), \text{conf}_f(\mathbf{x})). \quad (10)$$

As $q(\mathbf{x})$ captures precisely the characteristics, or *qualities*, of each point \mathbf{x} that are relevant for our guarantees, it is useful to introduce the concept of a *quality space* \mathcal{Q} . The quality space is a 2-dimensional real space defined by the outputs of the map q . For a given pair (ρ, κ) , we say \mathbf{x} is a *counterexample* to global ρ - κ -robustness if $q(\mathbf{x}) \in R(\rho, \kappa)$, where

$$R(\rho, \kappa) := \{(\rho', \kappa') \in \mathbb{R}^2 : \rho' < \rho \wedge \kappa' \geq \kappa\}. \quad (11)$$

The quality space allows us to easily define and detect whether f is robust or not. Each $R(\rho, \kappa)$ corresponds to the intersection of two-axis aligned half-spaces in \mathcal{Q} , as illustrated in [Figure 1](#).

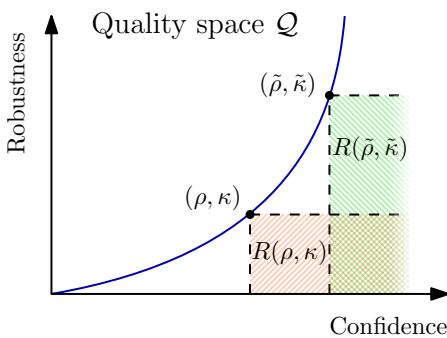


Figure 1. Visualization of the quality space \mathcal{Q} and of the region of counterexamples defined in [Equation \(11\)](#) for two possible pairs of robustness-confidence values (ρ, κ) and $(\tilde{\rho}, \tilde{\kappa})$.

We define the family of these counterexample ranges over all possible (ρ, κ) as

$$\mathcal{R} = \{R(\rho, \kappa) : (\rho, \kappa) \in \mathbb{R}^2\}. \quad (12)$$

Under a data distribution \mathcal{D} , we have

$$\begin{aligned} \Pr(R(\rho, \kappa)) &= \Pr(q(X) \in R(\rho, \kappa)) \\ &= \Pr(\text{conf}_f(X) < \rho \wedge \text{rob}_f(X) \geq \kappa). \end{aligned} \quad (13)$$

Given a set $N \subseteq \mathcal{X}$, we use $q(N)$ to denote the set $q(N) = \{q(\mathbf{x}) \in \mathcal{Q} : \mathbf{x} \in N\}$. With these concepts, we can obtain sample complexities independent from the dimensionality of \mathcal{X} . We obtain an iid sample N in \mathcal{X} , map it to \mathcal{Q} , and interpret $q(N)$ as ϵ -net over the range space $(\mathcal{Q}, \mathcal{R})$. This range space is equivalent to the intersection of two axis-aligned half-spaces in \mathbb{R}^2 , with a VC-dimension $d = 2$.

4.2. Bounding the Joint Probability

In this section, we define a method that can check under which conditions and for which (ρ, κ) the inequality

$$\Pr(\text{rob}_f(X) < \rho \wedge \text{conf}_f(X) \geq \kappa) < \epsilon \quad (14)$$

holds true. For our guarantee, we consider an iid sample N from the data distribution \mathcal{D} . If our iid sample is an ϵ -net, it intersects all ϵ -likely ranges. Therefore, if no counterexample to ρ - κ -robustness is found in an ϵ -net over $(\mathcal{Q}, \mathcal{R})$, then $R(\rho, \kappa)$ is known to be less than ϵ -likely. The following proposition formalizes this idea.

Proposition 4.2. *Let \mathcal{D} be a data distribution and $f : \mathcal{X} \rightarrow \mathbb{R}^n$ be a classifier. Given an ϵ -net $q(N)$ over the range space $(\mathcal{Q}, \mathcal{R})$, then for a given confidence κ and robustness radius ρ , we have that*

$$q(N) \cap R(\rho, \kappa) = \emptyset \implies \Pr(R(\rho, \kappa)) < \epsilon \quad (15)$$

where $R(\rho, \kappa)$ denotes the set of counterexamples to ρ - κ -robustness.

Proof. As $q(N)$ is an ϵ -net, it holds that

$$\forall R \in \mathcal{R} : \Pr(R) \geq \epsilon \implies q(N) \cap R \neq \emptyset. \quad (16)$$

By contraposition, this statement is equivalent to

$$\forall R \in \mathcal{R} : \Pr(R) < \epsilon \iff q(N) \cap R = \emptyset. \quad (17)$$

Then, $q(N) \cap R(\rho, \kappa) = \emptyset \implies \Pr(R(\rho, \kappa)) < \epsilon$. \square

This result allows us to bound the numerator of [Equation \(8\)](#). Note that [Proposition 4.2](#) allows us to give guarantees on the behavior of f . However, a bound on the joint probability (alone) may lead to vacuous guarantees. In fact, the probability in [Equation \(14\)](#) is trivially zero if we consider a confidence value κ large enough such that $\Pr(\text{conf}_f(X) \geq \kappa) = 0$. For this reason, we require additional information about $\Pr(\text{conf}_f(X) \geq \kappa)$.

4.3. Bounding the prediction confidence

In this section, we provide a method to lower bound the probability of obtaining a given prediction confidence as

$$\Pr(\mathbf{conf}_f(X) \geq \kappa) \geq p_{\min}. \quad (18)$$

[Equation \(18\)](#) is implied by the following statement about the cumulative distribution function of confidence:

$$\Pr(\mathbf{conf}_f(X) \leq \kappa) \leq 1 - p_{\min} \quad (19)$$

The following auxiliary lemma provides a general method to obtain a bound for a given quantile of a random variable from an iid sample.

Lemma 4.3. *Let K be a real-valued random variable, and N be an iid sample of K with $|N| = s$. Denote with $N_{(i)} \in \mathbb{R}$ the i^{th} -largest element in the sample. Then for parameters $1 > p \geq \frac{1}{2}$ and $0 < \delta < \frac{1}{2}$, with probability of at least $1 - \delta$, we have that*

$$\Pr(K \leq N_{(i)}) \leq p \quad (20)$$

for any integer i such that

$$i < sp - \sqrt{2sp \ln\left(\frac{1}{\delta}\right)}. \quad (21)$$

Proof sketch. The number of elements in an iid sample that are smaller or equal to the p -quantile K_p of K is a random variable $I \sim \text{Binom}(s, p)$. We are interested in finding the largest integer i such that $N_{(i)} \leq K_p$ holds with probability at least $1 - \delta$ and, therefore, the largest integer i such that $\Pr(I \geq i) \geq 1 - \delta$. We use a Chernoff bound for the deviation of I from its expected value $\mathbb{E}[I] = sp$ to obtain an integer i that is a high-probability lower bound for I . \square

We use [Lemma 4.3](#) with $K = \mathbf{conf}_f(X)$ and $p = 1 - p_{\min}$ to obtain κ values for which

$$\Pr(\mathbf{conf}_f(X) > \kappa) \geq p_{\min}. \quad (22)$$

As $\Pr(\mathbf{conf}_f(X) \geq \kappa) \geq \Pr(\mathbf{conf}_f(X) > \kappa)$, we arrive at [Equation \(18\)](#). For compactness, we use

$$i(s, p, \delta) := \max_{i \in \mathbb{N}} \left\{ i : i < sp - \sqrt{2sp \ln\left(\frac{1}{\delta}\right)} \right\} \quad (23)$$

to refer to the largest i which respects the bound in [Equation \(21\)](#). In our sample, we refer to the confidence value corresponding to this i as κ_{\max} . For confidence values $\kappa \leq \kappa_{\max}$, [Lemma 4.3](#) allows us to state with high probability that $\Pr(\mathbf{conf}_f(X) > \kappa) \geq p_{\min}$. We cannot, however, make a statement about confidence values that were not observed frequently enough in the sample, i.e., for $\kappa > \kappa_{\max}$. This upper bound on confidence values being certified is natural, as our statement relies only on the observed sample.

4.4. PAG Robustness

We have now introduced all necessary tools and conditions to certify a classifier f to be approximately globally robust. So far, we assumed an ϵ -net in the quality space \mathcal{Q} as given. We now discuss how to obtain such an ϵ -net using [Theorem 3.6](#) to derive our main contribution: an iid sampling-based procedure that allows us to certify the approximate global robustness of a classifier f with high probability.

Theorem 4.4 (PAG robustness). *Let \mathcal{D} be a data distribution, $f : \mathcal{X} \rightarrow \mathbb{R}^n$ be a classifier and \mathbf{rob}_f be a local robustness oracle. With $q(\mathbf{x})$ defined as in [Equation \(10\)](#), with parameters $0 < \epsilon, p_{\min}, \delta < \frac{1}{2}$. Consider a sample $N \sim \mathcal{D}$ with $|N| \geq s(\epsilon, \delta/2, 2)$ as per [Equation \(7\)](#) and an integer $i = i(|N|, 1 - p_{\min}, \delta/2)$ as per [Equation \(23\)](#). Then, with a probability of at least $1 - \delta$ the following implication holds for all ρ and for all $\kappa \leq N_{(i)}$:*

$$\begin{aligned} (q(N) \cap R(\rho, \kappa) = \emptyset) &\implies \\ \Pr(\mathbf{rob}_f(X) < \rho \mid \mathbf{conf}_f(X) \geq \kappa) &< \frac{\epsilon}{p_{\min}}. \end{aligned} \quad (24)$$

That is, under the absence of counterexamples in N , the classifier f can be certified to be Probably Approximately Globally (PAG) robust.

We present a proof of [Theorem 4.4](#) that builds on the following lemma, which reformulates [Proposition 4.2](#) as conditional probability.

Lemma 4.5. *Let $X \sim \mathcal{D}$ be a random data point in \mathcal{X} , $f : \mathcal{X} \rightarrow \mathbb{R}^n$ be a classifier, and parameters $0 < \epsilon, p_{\min} \leq \frac{1}{2}$ be a parameter. Given an ϵ -net $q(N)$ over the range space $(\mathcal{Q}, \mathcal{R})$, for all ρ, κ such that $\Pr(\mathbf{conf}_f(X) \geq \kappa) \geq p_{\min}$, it holds that*

$$\begin{aligned} (q(N) \cap R(\rho, \kappa) = \emptyset) &\implies \\ \Pr(\mathbf{rob}_f(X) < \rho \mid \mathbf{conf}_f(X) \geq \kappa) &< \frac{\epsilon}{p_{\min}}. \end{aligned} \quad (25)$$

Proof. Let us define, for brevity, the two random events $E_r = \{\mathbf{rob}_f(X) < \rho\}$ and $E_c = \{\mathbf{conf}_f(X) \geq \kappa\}$. By [Proposition 4.2](#) we know

$$q(N) \cap R(\rho, \kappa) = \emptyset \implies \Pr(E_r \wedge E_c) < \epsilon. \quad (26)$$

Furthermore, by the definition of conditional probability

$$\Pr(E_r \mid E_c) = \frac{\Pr(E_r \wedge E_c)}{\Pr(E_c)}. \quad (27)$$

When $\Pr(E_c) = \Pr(\mathbf{conf}_f(X) \geq \kappa) \geq p_{\min}$, it holds that

$$\Pr(E_r \mid E_c) \leq \frac{\Pr(E_r \wedge E_c)}{p_{\min}} < \frac{\epsilon}{p_{\min}}, \quad (28)$$

which completes the proof. \square

Proof of Theorem 4.4. We assume that $|N| \geq s(\epsilon, \delta/2, 2)$. Theorem 3.6 then implies that $q(N)$ is an ϵ -net over $(\mathcal{Q}, \mathcal{R})$ with a probability of at least $1 - \delta/2$. We further assert $i(|N|, 1 - p_{\min}, \delta/2) \leq i(|N|, 1 - p_{\min}, \delta/2)$. Lemma 4.3 then implies that $\Pr(\mathbf{conf}_f(X) \geq N_{(i)}) \geq p_{\min}$ with a probability of at least $1 - \delta/2$ as well. We use the union bound to state that both conditions hold true with a probability of at least $1 - \delta$. Given that both $N_{(i)}$ is a valid quantile bound, i.e., $\Pr(\mathbf{conf}_f(X) \geq N_{(i)}) \geq p_{\min}$, and $\kappa \leq N_{(i)}$, then $\Pr(\mathbf{conf}_f(X) \geq \kappa) \geq p_{\min}$. The theorem then follows from Lemma 4.5. \square

In some settings, it might not be possible to directly sample from the data distribution, but rather from a distribution \mathcal{D}' that is used to approximate the true distribution \mathcal{D} . Even in these settings we can transfer our guarantee, as long as we can quantify the difference between the two distributions.

Theorem 4.6 (PAG under distribution shift). *Let \mathcal{D} be a data distribution, $f : \mathcal{X} \rightarrow \mathbb{R}^n$ be a classifier and \mathbf{rob}_f be some robustness oracle. Let \mathcal{D}' be a distribution such that $\text{TV}(\mathcal{D}, \mathcal{D}') = \Lambda$. Consider an iid sample $N \sim \mathcal{D}'^s$ and parameters ϵ, δ and p_{\min} . Consider an s that satisfies the conditions in Theorem 4.4 for \mathcal{D}' . Then, with probability at least $1 - \delta$*

$$\begin{aligned} (q(N) \cap R(\rho, \kappa) = \emptyset) \implies \\ \Pr_{\mathcal{D}}(\mathbf{rob}_f(X) < \rho \mid \mathbf{conf}_f(X) \geq \kappa) < \frac{\epsilon + \Lambda}{p_{\min} - \Lambda}. \end{aligned} \quad (29)$$

That is, in the absence of counterexamples f can be certified to be PAG robust with respect to the data distribution \mathcal{D} .

Proof sketch. The data processing inequality (Raginsky, 2014) can be used to show (Lemma A.2) that Λ is an upper bound for the total variation distance of the distribution of instances sampled from \mathcal{D} and \mathcal{D}' when they are mapped into the quality space \mathcal{Q} . The numerator in the right-hand side can be derived by showing (Lemma A.3) that an ϵ -net under \mathcal{D}' is an $(\epsilon + \Lambda)$ -net under \mathcal{D} . The denominator can be derived by observing (Proposition A.5) that the quantiles of \mathcal{D} and \mathcal{D}' can differ by at most Λ . The theorem follows (Appendix A) from these results. \square

In the following section, we describe how to use the theoretical results obtained so far and practically derive robustness lower-bounds for a given classifier.

5. Robustness Lower-Bounds

In the previous section, we described how to provide global robustness guarantees given an iid sample N is found to be robust. Specifically, for a given choice of parameters ϵ and δ , we bound the required size of N . This single sample N can

be used to evaluate the robustness with all tuples (ρ, κ) . A lower-bound on the robustness of a point \mathbf{x} with prediction confidence $\kappa = \mathbf{conf}_f(\mathbf{x})$ can be obtained using the smallest observed robustness for a confidence of at least κ in the sample N . We define a mapping function $M(\kappa) \mapsto \rho$ that maps a given confidence κ to a corresponding robustness lower bound ρ :

$$M(\kappa) := \begin{cases} \min_{\mathbf{x} \in N} \mathbf{rob}_f(\mathbf{x}) & \text{if } \kappa \leq \kappa_{\max} \\ \text{s.t. } \mathbf{conf}_f(\mathbf{x}) \geq \kappa & \\ \text{UNDEFINED} & \text{else} \end{cases} \quad (30)$$

Equation (30) assures that no counterexample of ρ - κ -robustness is in N for $\rho = M(\kappa)$, and that the returned ρ satisfies all conditions of Theorem 4.4. The mapping function $M(\kappa)$ does not return a robustness lower-bound for confidence values larger than κ_{\max} . In this case, for the considered sample size the uncertainty for quantile estimation is too large to guarantee $\Pr(\mathbf{conf}_f(X) \geq \kappa_{\max}) > 0$, as described in Lemma 4.3. Note that, for $\kappa > \kappa_{\max}$, we can nevertheless guarantee that $\Pr(\mathbf{rob}_f(X) < \rho \wedge \mathbf{conf}_f(X) \geq \kappa) < \epsilon$. The map M can be constructed from a sample N in $\mathcal{O}(|N| \log(|N|))$ time, as detailed in Algorithm 1.

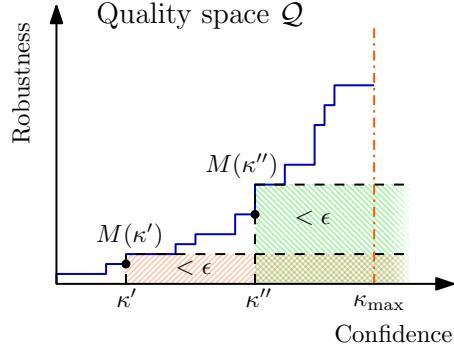


Figure 2. Visualization of the robustness lower-bound map $M(\kappa)$. Each rectangular region under the curve has a probability mass smaller than ϵ . The yellow line represents the maximum confidence value above which $M(\kappa)$ is undefined.

For a given mapping M , we now discuss how likely it is that our guarantees hold across all confidence values. This is not directly addressed by our analysis so far and requires to bound

$$\Pr(\mathbf{rob}_f(X) < M(\mathbf{conf}_f(X))). \quad (31)$$

We can obtain a bound on this probability as a direct consequence of Proposition 4.2. In this sense, the ρ - κ -mapping M is not only useful to obtain robustness lower-bounds for individual points, but can furthermore be used to bound the probability that for a point \mathbf{x} sampled from \mathcal{D} , it holds that $\mathbf{rob}_f(\mathbf{x}) < M(\mathbf{conf}_f(\mathbf{x}))$.

Proposition 5.1. Consider a classifier $f : \mathcal{X} \rightarrow \mathbb{R}^n$ and a $\rho\text{-}\kappa$ -mapping M constructed from an ϵ -net $q(N)$ as in Equation (30). Let $|M|$ be the size of the codomain of M . Then

$$\Pr \left(\mathbf{rob}_f(X) < M(\mathbf{conf}_f(X)) \right) < |M|\epsilon. \quad (32)$$

Proof. We use the properties of ϵ -nets together with a union bound based on the size of the codomain of the mapping M . As M consists, by construction, of $|M|$ discrete steps, $|M|$ ranges fully cover the area in the quality space below M . In the worst case, we have to assume their probability mass is additive and proceed with a union bound, proving our statement. \square

In practical settings, $|M|$ is typically small; we empirically demonstrate this in Section 6. We illustrate the practical effectiveness of our method in experiments with NNs in the following section.

6. Experiments

In this section, we discuss practical aspects that should be taken into account when applying our method. We elaborate on how both adversarial and formal methods can be used as an oracle to quantify local robustness. Moreover, we discuss how the data distribution \mathcal{D} can be approximated sufficiently well for our purposes with Gaussian noise. Our empirical results show that our theory can be readily translated into practical settings. In particular, we observe that: (i) more confident predictions do tend to be more robust; and (ii) our guarantees are meaningful, i.e., prediction confidence strongly influences robustness, and our guarantees capture this dependence well.

Setup We train a feed-forward NN with ReLU activation function on MNIST (Deng, 2012) and a pre-trained ResNet-20 (He et al., 2016) on CIFAR-10 (Krizhevsky, 2009), repeating the experiments for 3 different seeds. We use the *MAIR framework for robust training* (Kim et al., 2023) to train and fine-tune each network. The networks are trained once with a standard, non-robust procedure and once with the robust training procedure TRADES (Madry et al., 2017). The dataset is divided into three splits: training, sampling, and testing. The training and sampling splits are obtained from the MNIST/CIFAR-10 training datasets. The sampling split, specifically, is used to sample the ϵ -nets needed for our guarantees. In particular, the ϵ -net is obtained by applying additive Gaussian noise to the points in the sampling split. These points are unseen during the training procedure. The testing splits correspond to the MNIST/CIFAR-10 testing datasets. All experimental results have been obtained on a single desktop machine. For the exact setup, refer to Appendix C.

Robustness Oracles We utilize two robustness oracles, one based on adversarial techniques and one based on formal methods. Our adversarial oracle uses PGD (Madry et al., 2018), with a high number of small gradient steps, to report the closest counterexample found. For the experiments with PGD, we set $\epsilon = 10^{-4}$, $p_{\min} = 0.01$, $\delta = 0.01$ and thus sample $s(\epsilon, \delta/2, 2) = 685044$ images. Our formal robustness oracle uses the NN verification tool Marabou (Katz et al., 2019) to perform a binary search over the robustness radii for each point in our sampling dataset. We restrict formal verification to MNIST with a smaller sample size, due to computational restrictions (Appendix C). In both cases, the oracle certifies local robustness in an L_∞ hypercube centered on a given data point. With this setup, our only assumption about adversaries is that they do not produce (easily detectable) outliers.

Evaluation After the sampling procedure is completed, we construct the map M as in Equation (30) from the sampling split of our data. The map M provides the robustness lower-bound obtained with the sampling procedure. We are then interested in empirically checking whether the robustness lower-bound we obtained holds on the unseen testing split, that is, whether the testing data respects our guarantees. To this end, we define the following estimators. For a given confidence value κ , we estimate the conditional probability $\Pr(\mathbf{rob}_f(\mathbf{x}') < M(\kappa) \mid \mathbf{conf}_f(\mathbf{x}') \geq \kappa)$ for $\mathbf{x}' \in D_{\text{test}}$, that is, the probability that the testing data *does not* respect the robustness lower-bound provided by M .

$$p_\kappa = \frac{|\{\mathbf{x}' : \mathbf{rob}_f(\mathbf{x}') < M(\kappa) \wedge \mathbf{conf}_f(\mathbf{x}') \geq \kappa\}|}{|\{\mathbf{x}' : \mathbf{conf}_f(\mathbf{x}') \geq \kappa\}|}, \quad (33)$$

where $\mathbf{x}' \in D_{\text{test}}$. To estimate the worst-case violation of our guarantees, we consider the $\hat{p}_\kappa = \max_{\kappa \leq \kappa_{\max}} p_\kappa$ and check if $\hat{p}_\kappa \leq \epsilon/p_{\min}$. Because of our guarantee, we expect $\hat{p}_\kappa < 0.01$. We furthermore report

$$n_c = |\{\mathbf{x}' \in D_{\text{test}} : \mathbf{rob}_f(\mathbf{x}') < M(\mathbf{conf}_f(\mathbf{x}'))\}|. \quad (34)$$

This helps evaluate Proposition 5.1 empirically. As discussed in Proposition 5.1, $|M|$ denotes the number of steps in M , i.e., the size of its codomain. If Proposition 5.1 holds for our sample, we then expect $n_c \leq |D_{\text{test}}||M|\epsilon$. For both MNIST and CIFAR-10, $|D_{\text{test}}| = 10^4$ and thus $|D_{\text{test}}|\epsilon = 1$. We then expect $n_c/|M| < 1$.

Results Table 1 reports a summary of our experimental evaluation. We perform a total of 58 experiments, checking our guarantees against the estimators described above. In the large majority of cases, our estimators suggest that our guarantees hold across *all* confidence values. In only 3 experiments, the worst-case estimator \hat{p}_κ is above 0.01. To address this result, we first remark that, in accordance to Theorem 4.4, our guarantees may *not* hold with probability

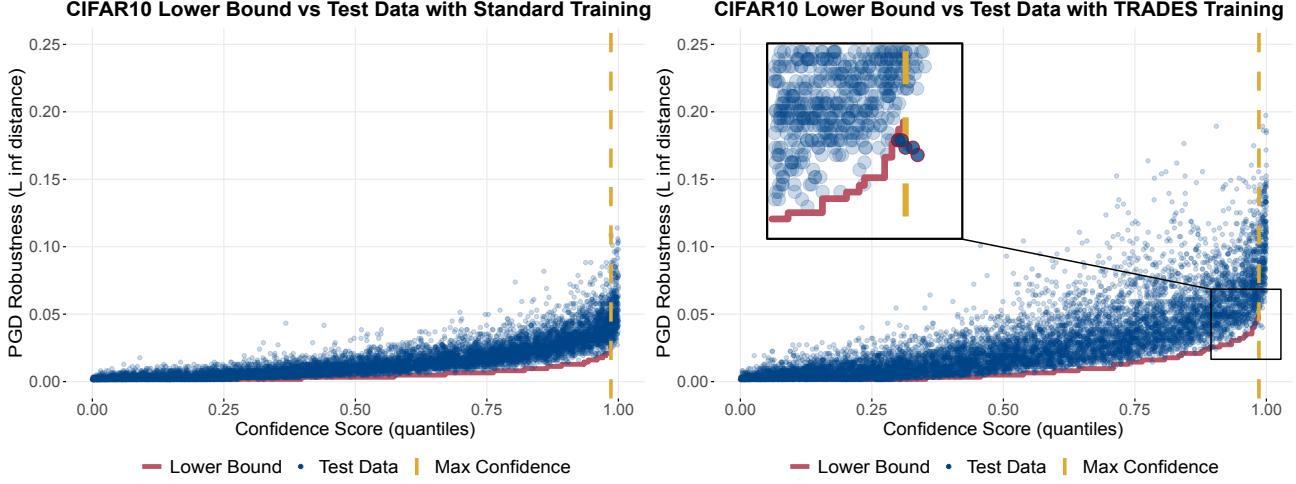


Figure 3. Scatter plot of the CIFAR-10 test dataset D_{test} , with $|D_{\text{test}}| = 10000$, in the quality space \mathcal{Q} for ResNet-20. The left network is trained with standard methods, the right network is trained robustly with TRADES. The red line depicts the lower bound obtained from validation sample N , with the parameters $\epsilon = 10^{-4}$, $\delta = p_{\min} = 0.01$, with $|N| = s(\epsilon, \delta/2, 2) = 685044$. On the right-hand side, the 5 counterexamples, which result in $\hat{p}_\kappa > 0.01$, are highlighted. Note that, despite this apparent violation, M tightly fits the test data.

up to 0.01. We could thus expect a small fraction of our experiments to provide guarantees which are not always satisfied. Additionally, a deeper analysis of the 3 experiments in question shows that these exceptions occur in tiny subsets of the data. With a small number of points, p_κ has a high variance and is thus more prone to falsely report violations. In all other cases, our guarantees are satisfied and describe the behavior of the networks well. With these considerations, we conclude that our guarantees are reliable and closely describe the networks under analysis.

Table 1. Summary results for all experiments. We report *worst* results aggregated over 3 random seeds and over the different hyperparameter values used for the TRADES adversarial training. The subscripts indicate standard training (ST) or adversarial training (AT), and if the used oracle is PGD (P) or Marabou (M). For each experiment, we report the values of \hat{p}_κ and $n_c/|M|$, where bold numbers denote that the estimators are consistent with our guarantees for all the $\kappa \leq \kappa_{\max}$, for all the runs considered. For compactness, we only report order of magnitude for $n_c/|M|$. Moreover, we report the number of individual “good runs” that are consistent with our guarantees when considering the worst-case \hat{p}_κ . More extensive results are available in Appendix D.

Experiment	$\hat{p}_\kappa \cdot 10^{-2}$	$n_c/ M $	good runs
CIFAR _{ST,P}	0.5	10^{-1}	3/3
CIFAR _{AT,P}	1.5	10^{-1}	14/15
MNIST _{ST,P}	0.6	10^{-2}	3/3
MNIST _{AT,P}	1.93	10^{-2}	33/35
MNIST _{ST,M}	0	0	1/1
MNIST _{AT,M}	10^{-2}	10^{-4}	1/1

In addition to the tabular results, we plot the map M and the test-set of CIFAR-10 with ResNet-20 in Figure 3. To investigate our method on the more challenging examples, we choose to plot the results for one of the 3 experiments for which we observe $p_\kappa > 0.01$ for some κ . Despite this, the lower bound provided by M generally holds on test data and well describes the specific behavior of the networks, despite very few (7) exceptions. The bounds also allow to clearly differentiate between the behavior of the robust and the non-robust network, as observable in Figure 4 too.

7. Conclusion

In this work, we use ϵ -nets to devise a sampling-based approach to provide probabilistic guarantees on the global robustness of a given classifier. Our approach is agnostic to the specific robustness oracle and can thus be adapted to a variety of existing approaches. Our guarantees are conditioned on the confidence of a prediction to allow for a flexible characterization of robustness. In our experimental evaluation, we used PGD and Marabou to evaluate local robustness and showed how our method: (i) characterizes the behavior of networks trained with both standard and adversarial methods, and (ii) obtains useful global robustness guarantees that transfer effectively to unseen data.

In future work, we will use other informative properties in addition to confidence, such as the predicted class, to further refine our guarantees. As our approach is agnostic to the properties of the tested object, it can be used in settings beyond NN verification.

References

- Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In *International conference on machine learning*, pp. 284–293. PMLR, 2018.
- Athavale, A., Bartocci, E., Christakis, M., Maffei, M., Nickovic, D., and Weissenbacher, G. Verifying global two-safety properties in neural networks with confidence. In *International Conference on Computer Aided Verification*, pp. 329–351. Springer, 2024.
- Baluta, T., Chua, Z. L., Meel, K. S., and Saxena, P. Scalable quantitative verification for deep neural networks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 312–323, 2021. doi: 10.1109/ICSE43902.2021.00039.
- Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017a. doi: 10.1109/SP.2017.49.
- Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec ’17*, pp. 3–14, New York, NY, USA, 2017b. Association for Computing Machinery. ISBN 9781450352024. doi: 10.1145/3128572.3140444. URL <https://doi.org/10.1145/3128572.3140444>.
- Casadio, M., Komendantskaya, E., Daggitt, M. L., Kokke, W., Katz, G., Amir, G., and Refaeli, I. Neural network robustness as a verification property: A principled case study. In Shoham, S. and Vizel, Y. (eds.), *Computer Aided Verification*, pp. 219–231, Cham, 2022. Springer International Publishing. ISBN 978-3-031-13185-1.
- Chen, Y., Wang, S., Qin, Y., Liao, X., Jana, S., and Wagner, D. Learning security classifiers with verified global robustness properties. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, CCS ’21*, pp. 477–494, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544. doi: 10.1145/3460120.3484776. URL <https://doi.org/10.1145/3460120.3484776>.
- Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.
- Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing 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, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6572>.
- Haussler, D. and Welzl, E. Epsilon-nets and simplex range queries. In *Proceedings of the Second Annual Symposium on Computational Geometry*, SCG ’86, pp. 61–71, New York, NY, USA, 1986. Association for Computing Machinery. ISBN 0897911946. doi: 10.1145/10515.10522. URL <https://doi.org/10.1145/10515.10522>.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pp. 97–117. Springer, 2017.
- Katz, G., Huang, D. A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljić, A., Dill, D. L., Kochenderfer, M. J., and Barrett, C. The marabou framework for verification and analysis of deep neural networks. In Dillig, I. and Tasiran, S. (eds.), *Computer Aided Verification*, pp. 443–452, Cham, 2019. Springer International Publishing. ISBN 978-3-030-25540-4.
- Kim, H., Park, J., Choi, Y., and Lee, J. Fantastic robustness measures: The secrets of robust generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=AGVBqJuL0T>.
- Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.
- Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672, 2019. doi: 10.1109/SP.2019.00044.
- 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>.
- Li, L., Xie, T., and Li, B. SoK: Certified Robustness for Deep Neural Networks. In *2023 IEEE Symposium on*

- Security and Privacy (SP)*, pp. 1289–1310, 2023. URL <https://doi.org/10.1109/SP46215.2023.10179303>.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9), 2017.
- 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>.
- Meng, M. H., Bai, G., Teo, S. G., Hou, Z., Xiao, Y., Lin, Y., and Dong, J. S. Adversarial robustness of deep neural networks: A survey from a formal verification perspective. *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2022a. doi: 10.1109/TDSC.2022.3179131.
- Meng, M. H., Bai, G., Teo, S. G., Hou, Z., Xiao, Y., Lin, Y., and Dong, J. S. Adversarial robustness of deep neural networks: A survey from a formal verification perspective. *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2022b. doi: 10.1109/TDSC.2022.3179131.
- Mitzenmacher, M. and Upfal, E. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, USA, 2nd edition, 2017. ISBN 110715488X. URL <https://dl.acm.org/doi/abs/10.5555/3134214>.
- Raginsky, M. Strong data processing inequalities and ϕ -sobolev inequalities for discrete channels. *IEEE Transactions on Information Theory*, 62:3355–3389, 2014. URL <https://api.semanticscholar.org/CorpusID:14185667>.
- Rao, Q. and Frtunikj, J. Deep learning for self-driving cars: chances and challenges. In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems, SEFAIS ’18*, pp. 35–38, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357395. doi: 10.1145/3194085.3194087. URL <https://doi.org/10.1145/3194085.3194087>.
- Sinha, A., Namkoong, H., and Duchi, J. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=Hk6kPgZA->.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.
- Vapnik, V. N. and Chervonenkis, A. Y. *On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities*, pp. 11–30. Springer International Publishing, Cham, 2015. ISBN 978-3-319-21852-6. doi: 10.1007/978-3-319-21852-6_3. URL https://doi.org/10.1007/978-3-319-21852-6_3.
- Wang, S., Chen, Y., Abdou, A., and Jana, S. Mixtrain: Scalable training of verifiably robust neural networks. *arXiv preprint arXiv:1811.02625*, 2018.
- Webb, S., Rainforth, T., Teh, Y., and Mudigonda, P. A statistical approach to assessing neural network robustness. *International Conference on Learning Representations*, 2019.
- Wu, H., Isac, O., Zeljić, A., Tagomori, T., Daggitt, M., Kokke, W., Refaeli, I., Amir, G., Julian, K., Bassan, S., et al. Marabou 2.0: a versatile formal analyzer of neural networks. In *International Conference on Computer Aided Verification*, pp. 249–264. Springer, 2024.
- Wu, M., Wicker, M., Ruan, W., Huang, X., and Kwiatkowska, M. A game-based approximate verification of deep neural networks with provable guarantees. *Theoretical Computer Science*, 807:298–329, 2020. ISSN 0304-3975. doi: <https://doi.org/10.1016/j.tcs.2019.05.046>. In memory of Maurice Nivat, a founding father of Theoretical Computer Science - Part II.
- Xu, K., Zhang, H., Wang, S., Wang, Y., Jana, S., Lin, X., and Hsieh, C.-J. Fast and Complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=nVZtXBI6LNn>.

A. Proofs

Lemma 4.3. Let K be a real-valued random variable, and N be an iid sample of K with $|N| = s$. Denote with $N_{(i)} \in \mathbb{R}$ the i^{th} -largest element in the sample. Then for parameters $1 > p \geq \frac{1}{2}$ and $0 < \delta < \frac{1}{2}$, with probability of at least $1 - \delta$, we have that

$$\Pr(K \leq N_{(i)}) \leq p \quad (20)$$

for any integer i such that

$$i < sp - \sqrt{2sp \ln\left(\frac{1}{\delta}\right)}. \quad (21)$$

Proof. Let $K_p \in \mathbb{R}$ be the p -quantile of K , i.e.,

$$\Pr(K \leq K_p) = p \quad (35)$$

The number of elements in an iid sample *smaller* than K_p is a binomially distributed random variable $I \sim \text{Binom}(s, p)$. We aim to find an integer i that results in a high-probability lower bound for I , that is, we look for the largest i such that

$$\Pr(I > i) \geq 1 - \delta \quad (36)$$

or, equivalently,

$$\Pr(I \leq i) = \sum_{k=1}^i \binom{s}{k} p^k (1-p)^{s-k} < \delta. \quad (37)$$

There is no straightforward closed form expression to find this integer i . We proceed using Chernoff bounds for the deviation of I from $\mathbb{E}[I] = sp$. Recall that Chernoff bounds for some relative deviation $\eta \in [0, 1]$ is defined as (Mitzenmacher & Upfal, 2017, Chapter 4)

$$\Pr(I \leq (1 - \eta)\mathbb{E}[I]) \leq \exp\left(\frac{-\eta^2 \mathbb{E}[I]}{2}\right). \quad (38)$$

In our discrete setting, the relative deviation η of I from $\mathbb{E}[I]$ is defined as $\eta := \frac{sp - i}{sp}$. The Chernoff bound can then be written as

$$\Pr(I \leq i) \leq \exp\left(\frac{-(sp - i)^2}{2sp}\right) < \delta. \quad (39)$$

We now perform routine calculations to obtain an upper bound for i .

$$-\frac{(sp - i)^2}{2sp} < \ln(\delta) \quad (40)$$

$$\frac{(sp - i)^2}{2sp} > \ln\left(\frac{1}{\delta}\right) \quad (41)$$

$$sp - i > \pm \sqrt{2sp \ln\left(\frac{1}{\delta}\right)} \quad (42)$$

$$i < sp - \sqrt{2sp \ln\left(\frac{1}{\delta}\right)} \quad (43)$$

Given Equation (43) holds for a particular i , at least i elements in a sample of size s are smaller than K_p with probability at least $1 - \delta$, i.e., $\Pr(N_{(i)} \leq K_p) \geq 1 - \delta$. Finally, the event $N_{(i)} \leq K_p$ implies $\Pr(K \leq N_{(i)}) \leq p$, which then holds true with a probability of at least $1 - \delta$ as well. \square

Definition A.1 (Total variation distance). Consider two distributions $\mathcal{D}, \mathcal{D}'$ over \mathcal{X} and let $R \subseteq \mathcal{X}$ be a measurable set. The total variation (TV) distance between \mathcal{D} and \mathcal{D}' is defined as

$$\text{TV}(\mathcal{D}, \mathcal{D}') = \sup_{R \subseteq \mathcal{X}} |\Pr_{\mathcal{D}}(R) - \Pr_{\mathcal{D}'}(R)|. \quad (44)$$

Lemma A.2 (Total variation distance in the quality space \mathcal{Q}). *Consider two distributions \mathcal{D} and \mathcal{D}' over \mathcal{X} and denote with $\text{TV}(\mathcal{D}, \mathcal{D}')$ their total variation distance. For the function $q : \mathcal{X} \rightarrow \mathbb{R}^2$ that maps instances into the quality space \mathcal{Q} . We denote the distribution of the instances after they are mapped onto \mathcal{Q} as $\mathcal{D}_q, \mathcal{D}'_q$. It then holds that*

$$\text{TV}(\mathcal{D}_q, \mathcal{D}'_q) \leq \text{TV}(\mathcal{D}, \mathcal{D}'). \quad (45)$$

Proof. For ease of presentation, we discuss a proof of the statement for more general conditions, which correspond to a version of the data processing inequality (Raginsky, 2014). Let \mathcal{F} be a σ -algebra on the input space \mathcal{X} and consider two probability measures P and Q on $(\mathcal{X}, \mathcal{F})$. Let $t : \mathcal{X} \rightarrow \mathcal{Y}$ be a measurable map and \mathcal{G} be a σ -algebra on the space \mathcal{Y} . With slight abuse of notation we denote with $t(P)(B) := P(t^{-1}(B)), \forall B \in \mathcal{G}$ the pushforward measure induced by t on $(\mathcal{Y}, \mathcal{G})$. We do so analogously for Q too. The total variation distance for probability measures is defined as

$$\text{TV}(t(P), t(Q)) = \sup_{B \in \mathcal{G}} |t(P)(B) - t(Q)(B)| = \sup_{B \in \mathcal{G}} |P(t^{-1}(B)) - Q(t^{-1}(B))| \quad (46)$$

Note that for every measurable set $B \in \mathcal{G}$ in the measure space $(\mathcal{Y}, \mathcal{G})$, the preimage $t^{-1}(B) \in \mathcal{F}$ is a measurable set in $(\mathcal{X}, \mathcal{F})$. It therefore holds that

$$\sup_{B \in \mathcal{G}} |P(t^{-1}(B)) - Q(t^{-1}(B))| \leq \sup_{A \in \mathcal{F}} |P(A) - Q(A)| = \text{TV}(P, Q), \quad (47)$$

from which it follows that

$$\text{TV}(t(P), t(Q)) \leq \text{TV}(P, Q). \quad (48)$$

The statement of lemma holds when t is the function q that maps to the quality space \mathcal{Q} , and \mathcal{Y} is \mathcal{Q} . \square

Lemma A.3. *Given two distributions \mathcal{D} and \mathcal{D}' on \mathcal{X} such that $\text{TV}(\mathcal{D}, \mathcal{D}') \leq \Lambda$, if N is an ϵ -net for \mathcal{X} with respect to \mathcal{D}' , then it is an $(\epsilon + \Lambda)$ -net for \mathcal{X} with respect to \mathcal{D} .*

Proof. Using Definition A.1, $\text{TV}(\mathcal{D}, \mathcal{D}') = \sup_{R \subseteq \mathcal{X}} |\Pr_{\mathcal{D}}(R) - \Pr_{\mathcal{D}'}(R)| \leq \Lambda$. Specifically, $\Pr_{\mathcal{D}}(R) \geq \epsilon + \Lambda$ implies that $\Pr_{\mathcal{D}'}(R) \geq \epsilon$. As N is an ϵ -net for \mathcal{X} with respect to \mathcal{D}' , it intersects all $R \subseteq \mathcal{X}$ such that $\Pr_{\mathcal{D}'}(R) \geq \epsilon$, and it thus intersects all $R \subseteq \mathcal{X}$ such that $\Pr_{\mathcal{D}}(R) \geq \epsilon + \Lambda$. Therefore, N is an $(\epsilon + \Lambda)$ -net for \mathcal{X} with respect to \mathcal{D} . \square

Lemma A.4. *Let K be a random variable sampled from \mathcal{D}' . Let \mathcal{D} be another distribution such that $\text{TV}(\mathcal{D}, \mathcal{D}') \leq \Lambda$, and $\Pr_{\mathcal{D}'}(K \leq \kappa) \leq p$ then we have that*

$$\Pr_{\mathcal{D}}(K \leq \kappa) \leq p + \Lambda \quad (49)$$

Proof. The proof follows from the fact that if $\text{TV}(\mathcal{D}, \mathcal{D}') \leq \Lambda$, then probability of no event under \mathcal{D} and \mathcal{D}' can differ by more than Λ . \square

Proposition A.5. *Consider a random variable $K \sim \mathcal{D}'$ and let $N, N_{(i)}$, p , and δ be as in the setting for Lemma 4.3 so that $\Pr_{\mathcal{D}'}(K \leq N_{(i)}) \leq p$ holds with probability at least $1 - \delta$ for any integer i such that $i < np - \sqrt{2np \ln(1/\delta)}$. Consider a distribution \mathcal{D} such that $\text{TV}(\mathcal{D}, \mathcal{D}') \leq \Lambda$. Then, under the distribution \mathcal{D} and for the same $N, N_{(i)}$, p , and δ , with probability of at least $1 - \delta$:*

$$\Pr_{\mathcal{D}}(K \leq N_{(i)}) \leq p + \Lambda. \quad (50)$$

Proof. Using Definition A.1, for any measurable set $R \subseteq \mathbb{R}^n$ it holds that $|\Pr_{\mathcal{D}}(R) - \Pr_{\mathcal{D}'}(R)| \leq \Lambda$. If we consider the event $\{K \leq N_{(i)}\} \subseteq \mathbb{R}^n$, with probability of at least $1 - \delta$ it holds that:

$$\Pr_{\mathcal{D}}(K \leq N_{(i)}) = \Pr_{\mathcal{D}'}(K \leq N_{(i)}) + \Lambda \stackrel{(*)}{\leq} p + \Lambda, \quad (51)$$

where in $(*)$ we used Lemma 4.3. \square

Theorem 4.6 (PAG under distribution shift). *Let \mathcal{D} be a data distribution, $f : \mathcal{X} \rightarrow \mathbb{R}^n$ be a classifier and rob_f be some robustness oracle. Let \mathcal{D}' be a distribution such that $\text{TV}(\mathcal{D}, \mathcal{D}') = \Lambda$. Consider an iid sample $N \sim \mathcal{D}'^s$ and parameters ϵ , δ and p_{\min} . Consider an s that satisfies the conditions in Theorem 4.4 for \mathcal{D}' . Then, with probability at least $1 - \delta$*

$$\begin{aligned} & \left(q(N) \cap R(\rho, \kappa) = \emptyset \right) \implies \\ & \Pr_{\mathcal{D}} \left(\text{rob}_f(X) < \rho \mid \text{conf}_f(X) \geq \kappa \right) < \frac{\epsilon + \Lambda}{p_{\min} - \Lambda}. \end{aligned} \quad (29)$$

That is, in the absence of counterexamples f can be certified to be PAG robust with respect to the data distribution \mathcal{D} .

Proof. We show that the two assumptions in Theorem 4.4 lead to the statement of the theorem if $\text{TV}(\mathcal{D}, \mathcal{D}') = \Lambda$.

1. To satisfy assumption 1. in Theorem 4.4, $s \geq \frac{2}{\epsilon} \left(\log \left(\frac{4}{\delta} \right) + d \log(2s) \right)$, i.e., N is an ϵ -net for \mathcal{X} with respect to \mathcal{D}' with probability at least $1 - \delta/2$. Because of Lemma A.3, N is also an $(\epsilon + \Lambda)$ -net for \mathcal{X} with respect to \mathcal{D} with probability at least $1 - \delta/2$.
2. To satisfy assumption 2. in Theorem 4.4, $i < s(1 - p_{\min}) - \sqrt{2s(1 - p_{\min}) \ln \left(\frac{2}{\delta} \right)}$ for $i = |\{x \in N : \text{conf}_f(x) < \kappa\}|$, i.e., $\Pr_{\mathcal{D}'}(\text{conf}_f(X) \leq \kappa) \leq 1 - p_{\min}$ with probability at least $1 - \delta/2$. Using Proposition A.5 with $p = 1 - p_{\min}$, it follows that $\Pr_{\mathcal{D}}(\text{conf}_f(X) \leq \kappa) \leq 1 - p_{\min} + \Lambda$ with probability at least $1 - \delta/2$.

By union bound, the probability that either condition does not hold on a given sample N is at most $\delta/2 + \delta/2 = \delta$. Thus, with probability of at least $1 - \delta$, N will be an $(\epsilon + \Lambda)$ -net for \mathcal{X} under \mathcal{D} and the quantile bound holds under \mathcal{D} . The data processing inequality (Raginsky, 2014) can then be used to show (Lemma A.2) that Λ is an upper bound for the total variation distance of the distribution of instances sampled from \mathcal{D} and \mathcal{D}' when they are mapped into the quality space \mathcal{Q} . With this, the theorem follows from the definition of ϵ -nets and Lemma 4.5. \square

B. Construction of Mapping

In this section, we expand on the definition of the mapping function M in Equation (30). The construction of the map can be performed in a single pass over the sorted sample N , and is illustrated in Algorithm 1. This process is $\mathcal{O}(|N| \log(|N|))$, bound by the process of sorting. At inference time, $M(\kappa)$ returns the ρ corresponding to the largest confidence $\kappa' \leq \kappa$.

The size of the mapping $|M|$, and the corresponding strictness of Proposition 5.1, can be controlled via quantization. Rounding down robustness radii reduces the size of the codomain of M , keeps the guarantees valid, and reduces the spacial requirements of the mapping.

Algorithm 1 Obtain κ - ρ -mapping

Input: ϵ -net N , confidence upper-bound κ_{\max}
Output: κ - ρ -mapping $M(\kappa)$
Let $M = \emptyset$
Let $\kappa' = -\infty$
/ iterate through sample in order of increasing robustness ρ */*
for $(\rho, \kappa) \in N$ in lexicographic order **do**
 | **if** $\kappa' < \kappa \leq \kappa_{\max}$ **then**
 | | $M = M \cup \{\kappa \mapsto \rho\}$ // add new step from κ to ρ to the mapping
 | | $\kappa' = \kappa$
 | **end**
end
return M

C. Details on Experimental Setup

In this section, we detail how the experiments were conducted. The full code to train, test and analyze the experiments is available at our [repository](#).

C.1. Software and Hardware Used

We use PyTorch to conduct our experiments and perform training with the MAIR (Kim et al., 2023) library. All the experiments were run on a single desktop machine equipped with an Intel i9-11900KF @ 3.50GHz CPU and a NVIDIA GeForce RTX 3080 GPU.

C.2. Robustness Oracles

PGD We adapt the internal PGD function in MAIR to return the distance to adversarial examples. To get a fine-grained result, we perform many gradient steps with a small step size. For MNIST, we use a step size of 0.5/256 and up to 200 steps to find an adversarial example. We project our examples to valid pixel values between 0 and 1. For CIFAR-10, we use a similar setup with a smaller step size of 0.1/256 and up to 500 steps. We constraint adversarial images to have L_∞ distance of at most 0.5 to the original data point.

Marabou We use the Python interface of the Marabou 2.0 NN verification tool, to verify the following query for our network for a given data point $\mathbf{x} \in \mathcal{X}$ and a given robustness radius ρ :

$$\exists \mathbf{x}' \in \mathcal{X} : \|\mathbf{x} - \mathbf{x}'\|_\infty \leq \rho \wedge \text{class}(\mathbf{x}) \neq \text{class}(\mathbf{x}'). \quad (52)$$

If the formula is UNSAT, the network is locally ρ -robust. We perform a binary search to get a bounded estimate of the local robustness radius, and obtain a lower bound of the L_∞ local robustness radius with 4 bits of precision, up to a value of 0.5.

C.3. NN Architectures

Feed Forward Network We considered three-layer, fully-connected ReLu networks with (768, 50, 10) neurons, trained on MNIST. Each network was trained for 20 epochs. The networks were trained both using standard training and using the TRADES method with varying values for the robustness-accuracy hyperparameter β . Refer to our [github repository](#) and to [Appendix D](#) for the exact (hyper)parameter values.

ResNet-20 We downloaded a pretrained ResNet-20 network from [this Github repository](#). Each network was then trained for 200 additional epochs. The networks were trained both using standard training and using the TRADES method with varying values for the robustness-accuracy hyperparameter β . Refer to [Appendix D](#) for the exact hyperparameter values.

C.4. Datasets and Splits

We use MNIST and CIFAR-10 for our experiments. We take a random split of 16% of the training dataset as sampling split. This split is not used during training. To obtain the sampling datasets, we consider the following settings:

- For the PGD oracle, we choose the verification parameters $\epsilon = 10^{-4}$ and $\delta = p_{\min} = 0.01$. This results in a sample size of $s(\epsilon, \delta/2, 2) = 685044$.
- For the Marabou oracle, due to the costly local verification procedure, we relax our parameters to $\epsilon = 2.5 \cdot 10^{-3}$, $\delta = 0.01$ and $p_{\min} = 0.05$. This results in a sample size of $s(\epsilon, \delta/2, 2) = 21892$.

For each of our experiments, the sampling dataset is then obtained by uniformly choosing data from the sampling split and adding Gaussian noise with a mean of 0 and a standard deviation of 8/256 for both. The sample we obtain is then used to derive our guarantees.

D. Additional Results

For each of our experimental runs, we report the training procedure used (Procedure), the seed, the TRADES robustness-accuracy parameter β , the mapping size $|M|$, the number of predictions for which κ is smaller than κ_{\max} denoted by $|\mathbf{x}_{\kappa \leq \kappa_{\max}}|$, the estimators n_c and \hat{p}_κ from [Section 6](#), the runtime, and the accuracy.

D.1. Additional Results on MNIST

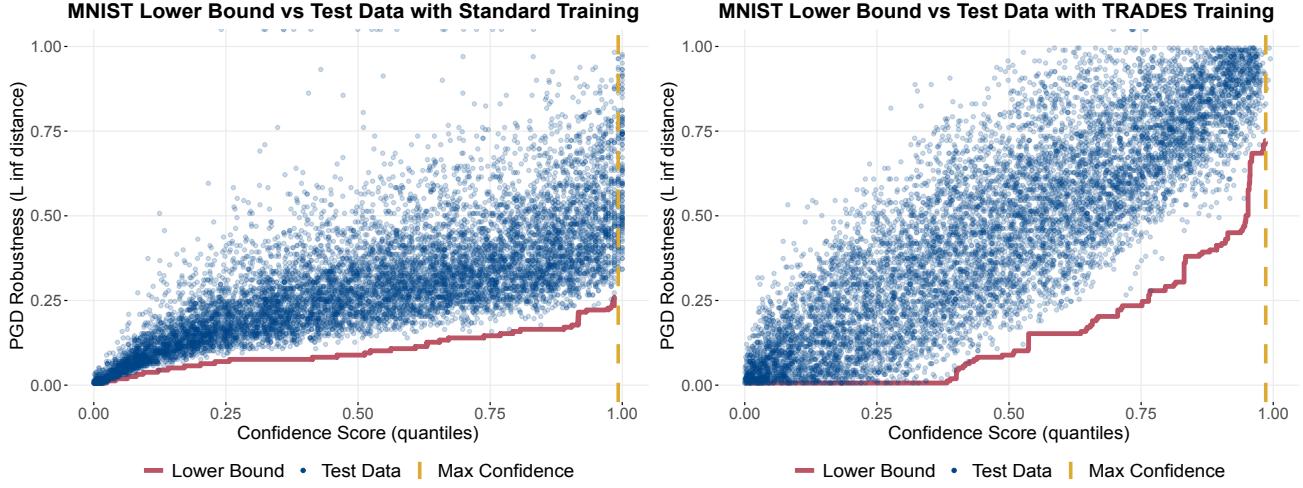


Figure 4. Scatter plot of the MNIST test dataset D_{test} , with $|D_{\text{test}}| = 10000$, in the quality space \mathcal{Q} for our feed forward network. The left network is trained with standard methods, the right network is trained robustly with TRADES ($\beta = 5$), both with random seed 20. Results for parameters $\epsilon = 10^{-4}$, $\delta = p_{\min} = 0.01$, with $|N| = s(\epsilon, \delta/2, 2) = 685044$.

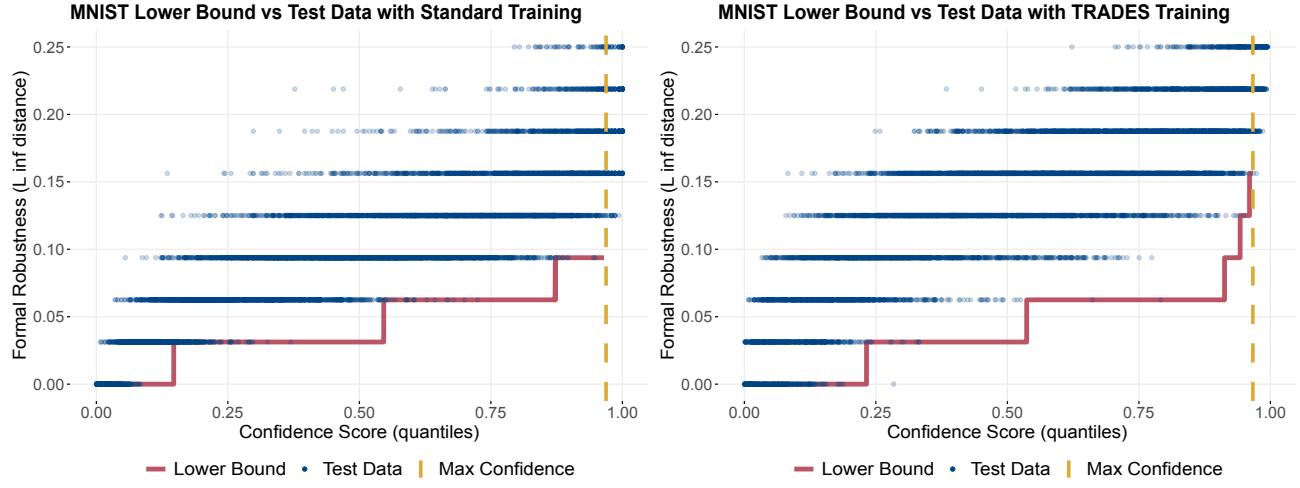


Figure 5. Scatter plot of the MNIST test dataset D_{test} , with $|D_{\text{test}}| = 10000$, in the quality space \mathcal{Q} for our feed forward network. The left network is trained with standard methods, the right network is trained robustly with TRADES ($\beta = 2$), both with random seed 20. Results for parameters $\epsilon = 2.5 \cdot 10^{-3}$, $\delta = 0.01$, $p_{\min} = 0.05$, with $|N| = s(\epsilon, \delta/2, 2) = 21892$.

Table 2. Results for the MNIST dataset using PGD as a local robustness oracle.

Procedure	seed	β	$ M $	$ \mathbf{x}_{\kappa \leq \kappa_{\max}} $	n_c	\hat{p}_κ	runtime (s)	accuracy
AT	10	0.001	31	9743	0	0	85	0.96
		0.01	46	9687	2	0.0017	92	0.98
		0.02	61	9668	1	0.0014	92	0.98
		0.05	50	9794	0	0	79	0.98
		0.1	66	9772	2	0.0002	87	0.98
		0.2	70	9764	5	0.0010	98	0.98
		0.5	58	9874	3	0.0022	67	0.97
		1	49	9904	5	0.0192	67	0.96
		10	40	9893	5	0.0004	101	0.82
		2	57	9879	3	0.0004	64	0.93
		20	56	9924	1	0.0020	99	0.58
		5	47	9894	4	0.0009	64	0.90
AT	20	0.001	39	9816	1	0.0020	79	0.95
		0.01	49	9686	1	0.0010	90	0.98
		0.02	48	9651	0	0	91	0.97
		0.05	53	9764	0	0	85	0.98
		0.1	61	9776	3	0.0002	59	0.98
		0.2	73	9811	5	0.0011	58	0.97
		0.5	69	9849	0	0	92	0.97
		1	50	9894	2	0.0003	100	0.95
		10	26	9875	4	0.0016	56	0.78
		2	55	9873	6	0.0118	70	0.92
		5	58	9892	0	0	94	0.88
AT	30	0.001	42	9739	0	0	90	0.96
		0.01	44	9840	0	0	55	0.98
		0.02	53	9746	2	0.0030	75	0.98
		0.05	61	9805	0	0	57	0.98
		0.1	67	9810	2	0.0008	58	0.98
		0.2	72	9762	3	0.0031	79	0.97
		0.5	69	9851	3	0.0034	94	0.97
		1	64	9884	2	0.0025	63	0.95
		10	36	9884	5	0.0082	105	0.79
		2	38	9877	0	0	75	0.93
		20	7	9879	0	0	95	0.65
		5	36	9877	0	0	64	0.90
Standard	10	0.1	34	9836	3	0.0053	71	0.97
Standard	20	0.1	32	9794	0	0	90	0.96
Standard	30	0.1	42	9858	0	0	77	0.94

Table 3. Results for the MNIST dataset using Marabou as a local robustness oracle.

Procedure	seed	β	$ M $	$ \mathbf{x}_{\kappa \leq \kappa_{\max}} $	n_c	\hat{p}_κ	runtime (h)	accuracy
AT	20	2	5	9659	1	0.0001	30	0.92
Standard	20	2	3	9585	0	0	30	0.96

D.2. Additional Results on CIFAR-10

Table 4. Results for the CIFAR-10 dataset using PGD as a local robustness oracle.

Procedure	seed	β	$ M $	$ \mathbf{x}_{\kappa \leq \kappa_{\max}} $	n_c	\hat{p}_κ	runtime (s)	accuracy
AT	10	0.1	22	9691	2	0.0004	561	0.76
		0.2	24	9735	7	0.0148	639	0.67
		0.5	33	9754	0	0	833	0.69
		1	31	9738	5	0.0022	892	0.81
		2	29	9770	0	0	1087	0.79
AT	20	0.1	19	9728	2	0.0010	596	0.82
		0.2	20	9727	2	0.0003	608	0.64
		0.5	31	9717	0	0	947	0.72
		1	38	9800	0	0	1025	0.70
		2	42	9746	2	0.0005	1165	0.82
AT	30	0.1	29	9771	8	0.0025	608	0.67
		0.2	21	9717	1	0.0004	666	0.63
		0.5	22	9672	3	0.0009	591	0.83
		1	29	9738	2	0.0025	945	0.81
		2	36	9785	2	0.0052	1184	0.79
Standard	10	0.1	12	9708	0	0	345	0.70
Standard	20	0.1	12	9564	0	0	331	0.75
Standard	30	0.1	15	9694	2	0.0020	423	0.48