Understanding ReLU Network Robustness Through Test Set Certification Performance

Nicola Franco Jeanette Miriam Lorenz
Karsten Roscher
Fraunhofer Institute for Cognitive Systems IKS, Munich, Germany
{nicola.franco, jeanette.miriam.lorenz, karsten.roscher}@iks.fraunhofer.de

Stephan Günnemann
Dept. of Computer Science & Munich Data Science Institute, Technical Univ. of Munich, Germany
guennemann@in.tum.de

Abstract

Neural networks can be vulnerable to small changes in input within their learning distribution, and this vulnerability increases for distributional shifts or input completely outside their training distribution. To ensure networks are used safely, robustness certificates offer formal assurances about the stability of their predictions in a pre-defined range around the input. However, the relationship between correctness and certified robustness remains unclear. In this work, we investigate the unexpected outcomes of verification methods applied to piecewise linear classifiers for clean, perturbed, in- and out-of-distribution samples. In our experiments focused on image classification, we observed that introducing a modest stability margin around the input sample leads to an important reduction in misclassified samples — approximately a 75% decrease — compared to the roughly 11% for samples that are correctly classified. This finding emphasizes the value of formal verification methods as an extra layer of safety, illustrating their effectiveness in enhancing accuracy for data that falls within the distribution. On the other hand, we provide a theoretical demonstration that formal verification methods robustly certify samples sufficiently far from the training distribution. These results are integrated with an experimental analysis and demonstrate their limitations compared to standard out-of-distribution detection methods.

1. Introduction

Building reliable artificial intelligence systems requires systematic methods for assessing their quality to gain confidence in their correctness or to identify possible failures. In general, neural networks are non-robust against geometric perturbations and are easily fooled by precisely calculated *adversarial attacks* [4, 37]. For these reasons, relying solely on model's prediction is not sufficient to ensure safe results. The problem of adversarial attacks has been addressed in the literature with a variety of defense mechanisms, divided into *empirical* and *provable* defenses. Empirical defenses aim to improve the robustness of the model through training with adversarial samples [1, 6, 13, 29]. However, robustness comes at the expense of accuracy [21, 43], and there is absolutely no guarantee that the model will behave correctly in the event of new, unseen attacks. To overcome this problem, formal verification methods [12, 22, 32, 34, 38], are proposed to increase the trustworthiness of a prediction by assuring its stability in the vicinity of the input.

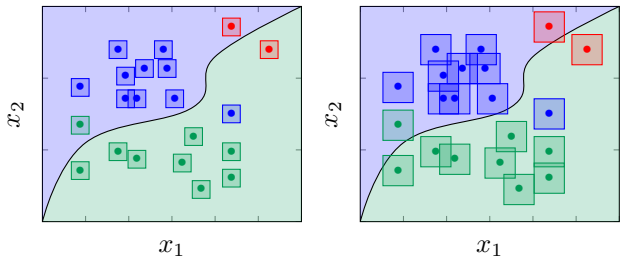


Figure 1. Bi-dimensional visualization of ℓ_∞-norm robustness certificates for ID (●, ●) & OOD (●). This visualization illustrates how ϵ (degree of perturbation) increases from left to right, approaching the decision boundaries of the neural network.

Motivation This study focuses on understanding how formal verification can be used at operational time, i.e., when the labels are not given (*label-free* [36]). In this context, we define as *robustness certification* the application of formal verification to ensure the stability of the prediction in the

vicinity of the input for a predefined range of perturbation. The research evaluates the effectiveness of robustness certification in enhancing prediction confidence and its potential as a safety evaluation criterion. It also examines the validity of robustness certificates for misclassified or Out-Of-Distribution (OOD) samples and their impact on prediction accuracy, both for In-Distribution (ID) and OOD instances. Previous research has mainly focused on improving robustness verification in qualitative or quantitative terms. For example, increasing the number of certified samples within the correctly classified ones, or speeding up the verification process [32, 34, 38]. Another line explores the tension between adversarial robustness and accuracy from an *empirical* [27, 43] or *provable* [21, 33] training perspective.

In this paper, we examine it through another perspective. We evaluate certified robustness independently of the classification results. Specifically, we address the following questions:

- **Certified and Correctly Classified** (●, ● in Fig. 1). Can we guarantee that a certified test sample has been classified correctly? This question wants to clarify if it possible to verify whether a test sample is classified correctly or not. And if there exists an optimal adversarial budget for achieving a good trade-off between accuracy and robustness.
- **In- or Out-Of-Distribution (ID or OOD)** (● in Fig. 1). Given an OOD sample that gets high confidence (i.e., is not detected by standard OOD detectors), there are two possible outcomes: certified or not certified. If the number of certified OOD samples is greater than the number of certified ID samples, then we cannot safely use formal verification methods. So, the question arise: can we detect if a test sample is from a different distribution with respect to the training distribution? Is it possible to verify whether a test sample is ID or OOD?

Contribution In this work, we conduct an in-depth analysis on ID and OOD data for various networks and certificate types, e.g. geometric or norm-based.
Our core contributions are summarized as follows:

- First evaluation on the relationship between correctness and certified robustness for clean and perturbed ID and OOD samples. We empirically show that the number of certified samples is directly related to the accuracy of the network and that robustness certificates are a powerful safety metric for ID data.
- Formal proof that robustness certificates are valid for samples sufficiently far from the training distribution in case of piecewise linear classifiers, e.g. ReLU networks.
- In the task of OOD detection, we show that the performance of verification methods is relatively lower than standard OOD detection approaches on normally and adversarially trained networks, and significantly lower for networks trained with OOD samples.

2. Background & Related Work

We define a neural network by a function $f : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{K}|}$ which maps input samples $x \in \mathbb{R}^d$ to output $y \in \mathbb{R}^{|\mathcal{K}|}$, where $\mathcal{K} = \{1, \dots, K\}$ is the set of K classes. We assume a feedforward architecture composed by affine transformations, $f^{(l)}(x) = W^l \sigma^{(l-1)}(x) + b^{(l)}$, for $l = 1, \dots, L$, and followed by ReLU activation functions, $\sigma^{(k)}(x) = \max\{0, f^{(k)}(x)\}$, for $k = 1, \dots, L-1$. In the end, the resulting classifier is obtained as composition of pre- and post-activations, i.e. $f^{(L)}(x) = W^{(L)}\sigma^{L-1}(x) + b^{(L)}$. In addition, we define all network parameters $(W^{(l)}, b^{(l)})$ as θ.

2.1. Adversarial robustness

Adversarial robustness refers to a model's ability to resist being fooled. Formally, given an input $x \in \mathbb{R}^d$, an adversary is allowed to choose any point $\tilde{x}$ from a convex set $\mathbb{S}(x) \subseteq \mathbb{R}^d$, such that $\arg\max_j f(x)_j \neq \arg\max_j f(\tilde{x})_j$. The set $\mathbb{S}(x)$ can be defined for different specifications, e.g. ℓ_p-norm perturbation [39], geometric transformations [3], randomized smoothing [9] and others.

Definition 2.1 (adversarial training)**.** In order to decrease the susceptibility of a network to adversarial perturbations, the prevalent strategy involves training the network based on the following minimax optimization problem:

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{in}}} \max_{\tilde{x}\in\mathbb{S}(x)} \mathcal{L}(f(\tilde{x}), y).$$

In the outer minimization, we consider training f on an ID dataset $\mathcal{D}_{\text{in}}$, while in the inner maximization we look for the maximum value of the loss function $\mathcal{L}$ that may give us an adversarial sample. As the inner maximization problem results intractable, most of the existing methods rely on approximations. For example, Projected Gradient Descent (PGD) [29] and Fast Gradient Sign Method (FGSM) [13] are commonly used techniques for improving robustness of a neural network, accomplished by generating adversarial examples and retraining the network with corrected labels.

2.2. Robustness certificate

As previously mentioned, a neural network f is certifiably robust for the input $x \in \mathbb{R}^d$ if the prediction for all perturbed versions remains unchanged.

Definition 2.2 (certified robustness)**.** An input x is considered certifiably robust for a neural network f if the prediction remains unchanged for all perturbed versions:

$$\arg\max_j f(x)_j = \arg\max_j f(\tilde{x})_j, \quad \forall \tilde{x} \in \mathbb{S}^p_\epsilon(x).$$

Formal verification methods model the previous definition as a mathematical optimization problem:

$$\min_{\tilde{x},\, t} \{f(\tilde{x})_k - f(\tilde{x})_t \mid \tilde{x} \in \mathbb{S}(x)\ ,\ t \in \mathcal{K} \setminus \{k\}\}\,.$$

This optimization process examines the differences by comparing the outputs of the neural network to predict any class other than the one initially predicted. If the result is positive, the input sample is certified as robust in $\mathbb{S}(x)$. Conversely, if an input exists that can deceive the network's prediction, the certification fails.

2.3. Convex relaxation

To reduce the runtime of the verification process, convex relaxation propagates the input set $\mathbb{S}(x)$ through the network producing lower and upper bounds at every layer. This speeds up the entire verification but sacrifices exactness, resulting in a lower bound at the output layer:

$$\underline{f}_k(\tilde{x}) - \overline{f}_t(\tilde{x}) \leq f_k^*(\tilde{x}) - f_t^*(\tilde{x}),$$

where f^* denotes the optimal result of the verification, and $\underline{f}$, $\overline{f}$ the lower and upper bounds, respectively. Current state-of-the-art methods, e.g. GPUPoly [32] or β-CROWN [38], parallelize the computation and propagation of boundaries on the GPU.

2.4. Robust OOD detection

OOD detection aims to determine whether a sample is originated from a learned distribution or not. Recently, researchers have investigated formal robustness guarantees for low network confidence on OOD samples. This entails verifying that a predictor assigns low values to all labels for OOD inputs within a specified neighborhood. An early approach integrates the softmax layer with density estimators based on Gaussian mixture models to differentiate between ID and OOD samples [30]. While the method achieves comparable OOD detection performance to previous approaches, such as Outlier Exposure (OE) [18], it guarantees a decrease in confidence when moving away from the training distribution. In this vein, [5] propose a training approach that employs interval bound propagation (IBP) to derive a provable upper bound on the maximal network confidence within an ℓ_∞-norm of ϵ around a given point. Although this method results in classifiers with pointwise guarantees for near-OOD samples, IBP generates loose bounds that lead to reduced network accuracy. More recently, [31] combined a binary discriminator to differentiate between ID and OOD samples with previous approaches, maintaining high clean accuracy while providing adversarial OOD guarantees. Despite achieving state-of-the-art performance across various OOD metrics and test distributions, their results are still not practically useful, as most remain below 50%.

Most of the existing literature focuses on improving empirical robustness to adversarial attacks inside [6, 13, 29] and outside [5, 15, 30, 31] the distribution or on formally demonstrating network stability in the input neighborhood [3, 14, 32, 34, 38]. Another line of work deals with the trade-off between accuracy and robustness from a training perspective [21, 27, 43] or on specific benchmarks [10, 16, 40]. Unlike these, in this work we evaluate how robustness certificates relate to accuracy on ID samples and how they perform on OOD samples.

3. In-Distribution Analysis

In this section, we assess various formal verification methods for different networks, perturbation types, and training approaches. A benchmark analysis is performed on clean and perturbed ID data to identify the number of samples that were successfully *certified and correctly classified* (CC) versus those that were *certified but incorrectly classified* (CI). Through this analysis, we aim to determine whether or not incorrectly classified samples will be robustly certified. Ultimately, if the false positive ratio is significantly lower or near zero, it would be reasonable to rely on the robustness verification process as an indicator of correct classification. Otherwise, it would not be a reliable measure.

Table 1. Summary of the metrics.

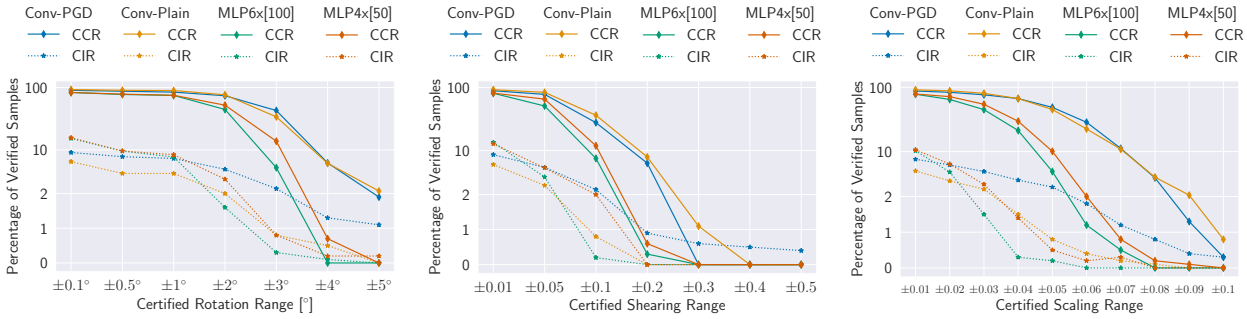
# of Samples	Certified Correct (CC)	Certified Incorrect (CI)
Total (N)	CCR = $^{CC}/_{N}$	CIR = $^{CI}/_{N}$
Relative	TPR = $^{CC}/_{C}$	FPR = $^{CI}/_{I}$

Following the approach in [19], similar evaluation metrics are presented in Tab. 1. The Certified Correct Ratio (CCR) and Certified Incorrect Ratio (CIR) are defined as the number of CC and CI samples divided by the total number of samples N, respectively. Likewise, the ratio of CC over the total number of correctly classified samples C is termed the True Positive Rate (TPR), and the ratio of CI over the total number of incorrectly classified samples I is called the False Positive Rate (FPR).

In this investigation, the Receiver Operating Characteristic (ROC) curve is generated by altering the size of the convex set calculated around the input sample. The CC and CI performance of robustness certificates are visually represented for an increasing certification range. Furthermore, the area under the receiver operating characteristic (AUC) is computed as an evaluation metric using TPR and FPR. To certify the robustness for geometric and norm-based perturbations we select the convex verifier GPUPoly [32].

Table 2. Networks trained on the first ten classes of the GTSRB dataset. The accuracy is computed on 4800 test samples.

Network	**Architecture**	**Activation**	**Training**	**Acc**	**# Neurons**
MLP4x[50]	4 FC	ReLU	Plain	84.8	210
MLP6x[100]	6 FC	ReLU	Plain	85.4	610
Conv	2 Conv. & 2 FC	ReLU	Plain/PGD	92.7/90.8	4852



(a) Rotation (b) Shearing (c) Scaling

Figure 2. Comparison of network architectures and training methods for rotation, shearing and scaling. We run robustness certificates on 1000 clean samples of the first 10 classes of the GTSRB test set.

3.1. Architectures & Training

We train a total of four neural networks on the first ten classes of the GTSRB dataset [20]. Two Fully Connected (FC) Multilayer Perceptron (MLP): MLP4x[50] and MLP6x[100] normally trained (plain), and two convolutional neural networks: one trained with PGD [29] attacks (ℓ_∞-norm attacks with $\epsilon = 0.01$ for a maximum of 40 steps), and the other normally trained, which are denoted as Conv-PGD and Conv-Plain, respectively. The clean accuracy (ACC) and other parameters are reported in Tab. 2. To achieve higher accuracy with such small networks, the number of classes was reduced to 10, which in turn decreased the quantity of features the network needed to learn.

3.2. Geometric robustness

Within this framework, DeepG [3] is employed to calculate linear inequality constraints surrounding the set of geometrically transformed images. In the context of DeepG, the number of samples (1000) used for the LP solver and the tolerance (0.01) in Lipschitz optimization were constant for increased perturbation values. In this experiment, clean (unperturbed) test samples and three geometric perturbations are considered: rotation, shearing, and scaling.

In Fig. 2, we show a comparison between architectures in terms of percentage of CCR and CIR for increasing perturbation values. The results reveal a relatively stable difference between CCR and CIR for small perturbation intervals, even as the range expands. Due to the lengthy computation time required to generate the convex set for each perturbation interval, a wide range of points could not be explored. This analysis visually demonstrates the relationship between robustly certified samples and accuracy. Differently from rotations, shearing and scaling result to be less prone to be certified. Ultimately, the objective is to identify a certified interval value that reduces CIR while maintaining a high CCR, thus ensuring certification reliability.

3.3. Norm-based robustness

In this analysis, the evaluation focuses on ℓ_∞-norm robustness certificates for clean test samples, which are defined as $\mathbb{S}(x) = \{\tilde{x} \in \mathbb{R}^d : \|x - \tilde{x}\|_\infty \leq \epsilon,\ \epsilon \geq 0\}$. In Fig. 3, The ROC curve is plotted for each network with 400 ϵ values between zero and 0.2, where ϵ represents the adversarial perturbation budget. In contrast to Fig. 3a, where the curve starts from the left-hand side for ϵ equal to zero, the curves in Fig. 3b and Fig. 3c begin on the right-hand side and shift to the left as ϵ increases.

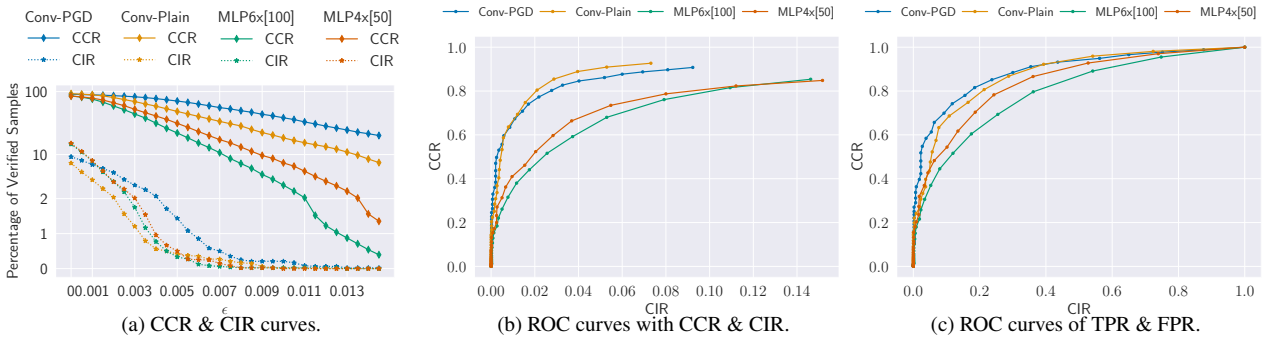
As noticing in Fig. 3a, convolutional neural networks yield better results with respect to fully connected ones and Conv-PGD turns out to be the best. In addition, we see that for very small $\epsilon \sim 0.003$, the $\sim 1\%$ of CI is comparatively very small respect to the $\sim 75\%$ of CC for Conv-Plain. Instead, at $\sim 2\%$ of CI we have $\sim 50\%$ of CC for fully connected models.

In Fig. 3b, we see that for small CIR$\sim$0.02, the CCR$\sim$0.8 remains surprisingly high. Within this range, the ROC of Conv-Plain stays mostly higher than that of Conv-PGD. We can associate this result to the fact that the plain model has higher accuracy (92.7) with respect to the adversarially trained one (90.8). In contrast, MLP6x[100], while having a slightly higher accuracy, leads to lower ROC than MLP4x[50]. This highlights that larger fully connected models are less likely to be certified and reduce the performance of robustness certificates.

In Fig. 3c, we plot the ROC curves for TPR and FPR. We observe that the ROC of Conv-PGD remains mostly higher than that of Conv-Plain. We attribute this result to the fact that adversarially trained networks are more easily certifiable than simple models, which leads to generally higher TPR results. Similarly to the case of CCR & CIR curves, MLP6x[100] demonstrates to be less prone to be certified and results in lower AUC with respect to MLP4x[50].

3.4. Distributional shift

In this section, we examine geometrically manipulated ID samples or *distributional shifts*. Each network is run on



(a) CCR & CIR curves. (b) ROC curves with CCR & CIR. (c) ROC curves of TPR & FPR.

Figure 3. Comparison of network architectures and training methods for varying ϵ of ℓ_∞-norm based robustness certificates on 4800 samples of the first 10 classes of the GTSRB test set.

perturbed samples, and the robustness verification results for the predicted class are evaluated. Testing neural networks for distributional shifts, practically assesses their use for real-world applications.

In Tab. 3, the results of AUCs for different networks and perturbations are presented. The ROC curves are roughly estimated using 400 ϵ values between 0 and 0.02, which is 20% of the maximum adversarial budget, causing the number of certified samples to reach zero for all tests and models. TPR and FPR, used to generate ROC curves, are calculated with ℓ_∞-norm robustness certificates, as in the previous section. The analysis reveals a decrease in accuracy on perturbed test sets, along with a similar decrease in AUC. Convolutional networks maintain higher AUCs compared to fully connected models, with Conv-PGD achieving the best results despite its lower accuracy. Adversarial training of ℓ_∞-norm samples benefits the same type of verification, resulting in more certified true positives and higher AUC.

Both metrics exhibit a similar trend, highlighting the relationship between accuracy and certified robustness. This finding indicates that distributional shifts (or perturbed ID samples) are as challenging to verify as the network's generalization ability is weaker. This holds true irrespective of the training procedure. Although adversarially-trained networks attain higher AUCs than plain models (consistent with results for unperturbed samples), the AUC of adversarially-trained networks decreases proportionally with respect to accuracy, confirming the correlation between accuracy and robustness.

Discussion In summary, the analysis on clean and perturbed ID samples highlights a strong relationship between robustness and accuracy. This is evident in Fig. 3b, where increasing the certification range leads to a reduction in both correctly and incorrectly classified samples. Luckily, we obtain more CC than CI samples for small perturbation budgets, demonstrating that robustness certificates serve as an effective safety metric for ID data.

As a numerical example, when accuracy decreases by approximately 10% (from 90% to 80%), the error rate drops by around 6% (from 8% to 2%). Similar results are obtained for both types of certification (geometric and norm-based). Thus, analogous conclusions can be drawn for other verification and training methods, such as randomized smoothing. In essence, the more inclined a network is towards certification, the better we can use verification methods as a metric to differentiate between correctly and incorrectly classified samples.

4. Out-Of-Distribution Analysis

In this section, we provide a theoretical analysis showing that robustness certificates apply to samples that are sufficiently distant from the training distribution. Hein et al. [15] demonstrated that piecewise linear classifiers maintain high confidence for samples outside the training distribution, and post-processing techniques for softmax scores are unable to reduce this confidence. This inherent issue with network architecture further results in the improper use of formal verifiers. A critical problem with adopting such methods is that OOD samples not only exhibit high confidence but are also easily verifiable and end up being certified as correct.

4.1. Theoretical analysis

Here, we formally demonstrate that robustness certificates are always valid for piecewise linear classifiers and for samples significantly distant from the training distribution. This finding is derived from a more general result shown in [15]. Let us introduce some definitions essential for the main proof. We briefly revisit the definition of continuous piecewise affine classifiers [2], which applies to feedforward neural networks with piecewise affine activation functions, such as ReLU, and linear at the output layer.

Definition 4.1. A function $f : \mathbb{R}^d \to \mathbb{R}$ is called piecewise affine if there exists a finite set of polytopes $\{Q_r\}_{r=1}^M$ (referred to as linear regions of f) such that $\cup_{r=1}^M Q_r = \mathbb{R}^d$ and f is an affine function when restricted to every Q_r.

Table 3. **AUC / ACC:** Comparison between plain and perturbed test samples. The ROCs were calculated with ℓ_∞-norm robustness certificates by varying ϵ. Random perturbation sizes inside the defined ranges are applied to the 4800 test samples of the first 10 classes of the GTSRB test set.

Perturbation		**Conv-PGD**		**Conv-Plain**		**MLP6x[100]**		**MLP4x[50]**	
Type	**Size**	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
Unperturbed	-	87.7	90.8	85.7	92.7	75.8	85.4	81.4	84.8
Gaussian Blur	$K = 3, \sigma \in [1, 2]$	82.4	85.7	79.7	91.7	69.3	80.2	66.9	77.9
Rotation	$[-30°, +30°]$	77.2	71.9	69.0	63.8	66.5	59.7	70.4	59.0
Scaling	$[0.1, 1]$	54.4	38.6	50.9	38.6	49.6	24.9	53.1	27.4

This definition applies to all layers performing linear mappings, e.g. fully connected, convolutional, residual layers, skip connections and further maximum and average pooling. Specifically, given a classifier $f : \mathbb{R}^d \to \mathbb{R}^K$, where K is the number of classes, Definition 4.1 applies to each component $f_i : \mathbb{R}^d \to \mathbb{R}$ and all K components $(f_i)_{i=1}^K$ have the same set of linear regions. We further extend the definition of ReLU networks as piecewise linear classifiers with the fact that all linear regions are polytopes and thus convex sets [15].

Lemma 4.1 (Asymptotic overconfidence [15])**.** Let $\{Q_r\}_{r=1}^R$ be the set of convex polytopes where the ReLU-classifier $f : \mathbb{R}^d \to \mathbb{R}^K$ is an affine function, meaning for every $k \in \{1, \dots, R\}$ and $x \in Q_k$ there exists $V^k \in \mathbb{R}^{K \times d}$ and $c^k \in \mathbb{R}^K$ such that $f(x) = V^k x + c^k$. Thus, for any $x \in \mathbb{R}^d \setminus \{0\}$ there exists $\alpha \in \mathbb{R}$ with $\alpha > 0$ and $r \in \{1, \dots, R\}$ such that $\beta x \in Q_r$ for all $\beta \geq \alpha$.

Given Lemma 4.1, we can state our result.

Theorem 4.1. Let $\cup_{r=1}^M Q_r = \mathbb{R}^d$ and $f(x) = V^r x + a^r$ be the piecewise affine representation of the output of a ReLU network on Q_r. If V^r does not contain identical rows for all $r = 1, \dots, R$, then for almost any $x \in \mathbb{R}^d \setminus \{0\}$, there exists $\alpha \in \mathbb{R}$ with $\alpha > 0$ and a predicted class $k \in \mathcal{K}$ such that:

$$\min_{z,t} \{f_k(z) - f_t(z) \mid z \in \mathbb{S}(\alpha x),\ t \in \mathcal{K} \setminus \{k\}\} > 0,$$

holds for $\mathbb{S}(\alpha x) \subset Q_r$.

Proof. By Lemma 4.1, there exists a region Q_r, with $r \in \{1, \dots, R\}$ and $\beta > 0$ such that for all $\alpha \geq \beta$ we have $\alpha x \in Q_r$. Given that $\mathbb{S}(\alpha x) \subset Q_r$ and since $z \in \mathbb{S}(\alpha x)$ we have that $z \in Q_r$. Let $f(z) = V^r z + a^r$ be the affine form of the ReLU classifier f on Q_r. Let $k^* = \arg\max_k \langle v_k^r, z \rangle$, where v_k^r is the k-th row of V^r. Given the fact that V^r does not has identical rows, i.e. $v_l^r \neq v_m^r$ for $l \neq m$, the maximum is unique up to zero. If the maximum is unique, it holds for sufficiently large $\alpha \geq \beta$:

$$\langle v_{k^*}^r, z \rangle + a_{k^*}^r - \langle v_t^r, z \rangle - a_t^r > 0, \quad \forall t \in \mathcal{K} \setminus \{k^*\}.$$

□

The primary implication of this theorem is that the surrounding area of infinitely many samples, which are far enough from the training distribution, can be easily certified as robust. As highlighted in [15], the constraint on V^r is rather weak. However, the fact that $\mathbb{S}(\alpha x) \subset Q_r$ is not as straightforward, since the definition of $\mathbb{S}(x)$ may vary depending on the type of certificate one is interested in.

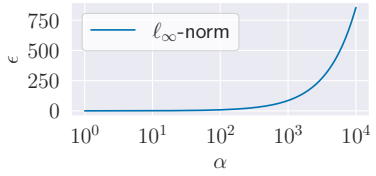


Figure 4. Given a single input αx, we compute the robustness certificate based on the ℓ_∞-norm for increasing α. As noted, α and ϵ are linearly correlated.

In Fig. 4, we show the relationship between α and ϵ for ℓ_∞-norm robustness certificates, where ϵ is the adversarial budget, i.e. $\mathbb{S}(\alpha x) = \{\tilde{x} \in \mathbb{R}^d \mid \|\alpha x - \tilde{x}\|_\infty \leq \epsilon, \epsilon \geq 0\}$. We display the maximum ϵ value for which the certificate holds for increasing α. One can note that this settings is unlikely in practice as all the images are normalized to be inside the interval $[0, 1]^d$ and therefore $\epsilon \in [0, 1]$. Despite this, we theoretically demonstrate that samples far enough from the training distribution are expected to be certified, and the certification range expands as the distance increases.

This issue poses a significant challenge for the practical application of robustness certificates, as samples far from the training distribution are likely to be certified. Therefore, incorporating formal verification methods with OOD detectors is recommended to address this limitation effectively.

4.2. Experimental analysis

In this section, we conduct experiments on OOD samples for different datasets, networks and training methods. The aim is to evaluate the performance of robustness certificates in detecting whether a sample is ID or OOD. To this end, we compare the convex verifier *GPUPoly* [32] against standard OOD detection methods: *MPS* [17], *ODIN* [26], Mahalanobis distance [25] (*Mahala*) and *Energy* [28]. We ran all methods on the entire test set except GPUPoly, which was executed only on the first 1000 test samples. This is due

to the incredibly long run time of validating a large amount of samples for a large range of ϵ values. The ROC curve for GPUPoly has been computed by varying the adversarial budget ϵ. Here, we consider as true positives all certified samples from the ID test set, and as false positives all certified samples from the OOD test set. We define $4\,000$ ϵ values equidistant between 0 and 0.2, i.e. 20% of maximum adversarial budget, which push the amount of certified samples to zero for all tests and models. As an example, verifying $1\,000$ images on our largest network (31360 neurons) with GPUPoly takes about 20 minutes per single ϵ. We conduct our experiments on a Nvidia GPU RTX 3090.

Datasets In our evaluation, we consider three ID datasets: MNIST [24], GTSRB [20], and CIFAR10 [23]. For MNIST, we include as OOD datasets with grayscale images of size 28x28: EMNIST [8], KMNIST (i.e., Kuzushiji-MNIST [7]), and FMNIST (i.e., Fashion-MNIST [41]). For GTSRB [20], which has RGB images of size 32x32, we use CIFAR10/100 [23] and SVHN [35]. Similarly, for CIFAR10, we replace CIFAR10 with GTSRB. Additionally, we employ OrganAMNIST from MedMNIST [42] and ImageNet Cropped (C) [11] for training OOD aware models. For each category, we normalize all datasets using the same mean and standard deviation as the ID training set.

Table 4. Networks architectures for each dataset category.

Input	Network	Architecture	Activation	# Neurons
28x28x1	MLP6x[200]	6 FC	ReLU	1 000
	ConvSmall	2 Conv. & 2 FC	ReLU	3 604
32x32x3	ConvSmall	2 Conv. & 2 FC	ReLU	4 852
	ConvMed	5 Conv. & 3 FC	ReLU	6 756

Network Architectures In Tab. 4, we describe the architectures, activation type and number of neurons for each dataset category. Evaluation is carried out on different training procedures. Networks trained only with clean training data are called **Plain**. Adversarially trained networks are **PGD** [29] or **FGSM** [13], where $\epsilon = 1/255$ is the adversarial perturbation budget. **RS** are networks trained with randomized smoothing [9] where $\sigma = 0.1$ is the standard deviation. Lastly, **OE** stands for Outlier Exposure [18], where we insert the OOD training set in parentheses.

MNIST In Tab. 5, we show the results on grayscale datasets, where we use MNIST as ID dataset. Given the limited size of the models, PGD and FGSM attacks prevent convergence during training, so we evaluate only Plain, OE and Randomized trained networks in this analysis. In case of OE, we consider two datasets: OrganAMNIST (OE-O) and FMNIST (OE-F). As might be expected, networks trained with OE perform significantly better than those trained with Plain or Randomized. The results with FMNIST as OOD training set compared to OrganAMNIST are surprisingly close to optimum in KMNIST for all standard OOD detection methods.

Table 5. **ID: MNIST.** Comparison between standard OOD detection methods and robustness certificates of ℓ_∞-norm: GPUPoly(ℓ_∞) [32]. We report the clean accuracy on MNIST test set. All methods were executed on all samples in the test set except GPUPoly, which was executed on the first 1000 test samples. In the context of GPUPoly, AUC and FPR95 are computed by varying the adversarial budget ϵ.

Network/ Train (Acc.)	Method	EMNIST (letters)		KMNIST		FMNIST	
		AUC↑	FPR95↓	AUC↑	FPR95↓	AUC↑	FPR95↓
MLP6x[200] OE-O (97.9)	MPS	90.6	38.6	**98.2**	8.7	97.9	12.4
	ODIN	**90.7**	**36.2**	98.1	**8.4**	**98.2**	**10.1**
	Mahala	90.5	39.7	97.0	15.1	97.3	12.2
	Energy	90.5	39.0	98.1	9.3	97.4	14.1
	GPUPoly	82.3	55.0	92.1	31.0	87.7	45.0
MLP6x[200] OE-F (98.2)	MPS	**97.0**	**11.7**	**99.8**	**0.8**	-	-
	ODIN	96.5	13.7	**99.8**	0.9	-	-
	Mahala	96.1	15.7	99.7	1.4	-	-
	Energy	96.9	12.3	**99.8**	0.9	-	-
	GPUPoly	89.7	29.3	94.1	25.2	-	-
ConvSmall Plain (98.8)	MPS	79.3	61.1	85.5	51.4	85.7	58.1
	ODIN	80.0	60.1	85.3	51.8	85.0	59.1
	Mahala	**91.4**	**38.9**	**92.0**	43.1	**91.9**	**43.2**
	Energy	80.5	57.7	85.3	51.9	83.5	64.2
	GPUPoly	81.9	48.3	87.1	**39.4**	78.3	64.6
ConvSmall RS (98.7)	MPS	73.5	73.0	87.4	48.0	81.3	65.9
	ODIN	73.5	72.5	86.9	49.7	79.6	67.7
	Mahala	**91.6**	**38.7**	89.7	54.1	**82.7**	61.0
	Energy	75.2	68.6	87.3	47.8	79.9	69.0
	GPUPoly	84.8	50.3	**90.2**	**39.6**	82.4	**58.9**

In the context of GPUPoly, we observe better results compared to other methods for convolutional networks in the KMNIST dataset, and definitely lower results for fully connected models. On the one hand, GPUPoly struggles to certify samples in distribution, leading to inferior results than standard OOD detection methods. On the other hand, for FPR at 95% of true positives, we obtain more certified OOD samples, empirically validating the hypothesis that verification methods easily certified samples far enough from the training distribution. The hardness of verifying OE trained networks should be related to the slightly thinner decision boundaries induced during the training procedure. Surprisingly, the randomized trained convolutional network performed slightly better than its plain counterpart. In App. Sec. 6.1, we report the ROC curves for convolutional networks.

GTSRB In this section, we test ConvMed trained on GTSRB. The accuracy on clean samples is relatively low compared to state-of-the-art models and adversarially trained networks have slightly lower accuracy than plain models. However, this is consistent with related work on verification methods such as [32, 34].

In Tab. 6, we show the results for the ConvMed model. In this setting, we trained each network on all 43 classes

Table 6. **ID: GTSRB.** Comparison between standard OOD detection methods and GPUPoly(ℓ_∞) [32] for different training methods of the ConvMed network. We report the clean accuracy on GTSRB test set. In the context of GPUPoly, the AUC and FPR95 are computed by varying the adversarial budget ϵ of the ℓ_∞-norm based robustness.

Train (Acc.)	Method	CIFAR10		CIFAR100		SVHN	
		AUC↑	FPR95↓	AUC↑	FPR95↓	AUC↑	FPR95↓
OE (83.3)	MPS	97.9	0.6	**97.9**	1.0	**97.7**	2.5
	ODIN	**99.9**	**0.4**	**97.9**	**0.8**	**97.7**	2.4
	Mahala	97.8	0.5	97.7	1.1	97.3	**1.7**
	Energy	97.7	0.6	97.7	0.9	97.5	2.7
	GPUPoly	18.9	99.3	20.1	99.2	34.2	97.3
FGSM (84.1)	MPS	61.4	94.7	64.0	93.0	77.1	81.2
	ODIN	**66.9**	**81.9**	**69.5**	**78.4**	80.9	64.0
	Mahala	62.8	83.2	63.6	82.9	**81.9**	**61.9**
	Energy	62.2	95.8	65.1	94.1	76.2	87.3
	GPUPoly	57.9	95.1	60.5	95.0	70.4	90.7
PGD (81.4)	MPS	58.1	96.0	58.7	92.9	83.7	69.5
	ODIN	64.0	85.1	63.2	80.4	88.2	47.7
	Mahala	**73.8**	**75.1**	**68.1**	**79.1**	**89.0**	**44.7**
	Energy	54.1	97.9	55.2	95.6	80.0	78.1
	GPUPoly	55.4	95.5	58.2	93.8	70.3	90.5
RS (83.7)	MPS	61.6	94.8	62.6	92.6	83.5	71.0
	ODIN	**67.1**	82.9	**67.6**	**79.9**	87.4	51.8
	Mahala	65.6	**80.6**	64.0	83.1	**88.2**	**47.1**
	Energy	60.9	95.3	62.3	93.2	81.7	76.4
	GPUPoly	60.4	92.1	62.7	91.3	72.0	86.9

of the GTSRB dataset. As a consequence, we obtain lower accuracy with respect to the models of Sec. 3 trained on just the first 10 classes. Similarly to the grayscale category, standard OOD detection methods perform likewise. In the case of OE (ImageNet (C)), GPUPoly certifies more OOD than ID samples, drawing the AUC below the random guess value of 0.5. On the one side, adversarial training procedures, such as PGD, FGSM and randomized, do not seem to help the verification process, resulting in substandard performance for GPUPoly. On the other side, standard OOD detection methods are slightly affected.

In Fig. 5, we display the ROC curve of ConvMed, which was trained with OE using GTSRB as the ID dataset and ImageNet cropped as the OOD dataset. We observe that GPUPoly certifies more OOD samples than ID samples. We attribute this behavior to two reasons. First, OE induces an irregular gradient that causes the verification process to fail for both ID and OOD samples, resulting in fewer robustness certified samples and affecting TPR and FPR equally. Second, OE decreases the accuracy of ID samples, resulting in more stable gradients and a larger prediction space for OOD samples. This leads to an increase in the number of OOD certified samples and an increase in FPR. In conclusion, this experiment empirically confirms the theoretical results discussed earlier. Additionally, we present the CIFAR10 results in Appendix 6.

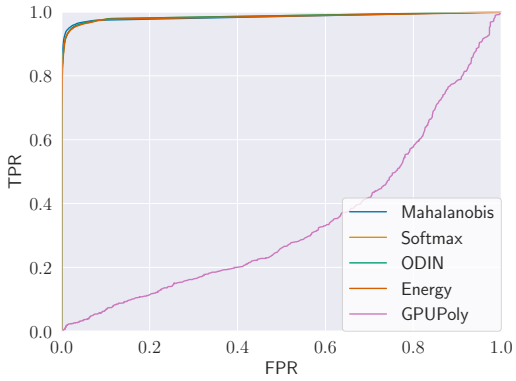


Figure 5. Comparison of ROC curves for standard OOD detection methods and GPUPoly on SVHN dataset. We consider the ConvMed model trained with OE on the GTSRB as ID and ImageNet cropped as OOD training sets.

Discussion In summary, our analysis demonstrates that for adversarially-trained networks, robustness certificates and standard OOD detection methods perform similarly on grayscale and RGB images. However, when using networks trained with OE, the performance of robustness certificates decreases significantly, indicating the limitation of formal verification methods in easily certifying OOD samples for networks trained to be OOD aware. Therefore, to ensure safe deployment of piecewise linear classifiers, additional safety measures should be considered in conjunction with robustness certificates.

5. Conclusion

In this paper, we conduct an in-depth analysis of the robustness of ReLU networks to clean and perturbed samples within and outside the training distribution, using convex verification methods to certify the network predictions. By varying the adversarial perturbation budget, we constructed ROC curves. Our ID analysis showed a strong correlation between certified robustness and accuracy for both clean and perturbed samples, indicating the usefulness of formal verification methods as an error-reduction metric that increases reliability. However, the OOD analysis revealed different results, demonstrating the unreliability of robustness certificates compared to standard OOD detection methods. We proved theoretically that ReLU classifiers can easily certify samples far from the training distribution, which was validated through extensive experiments. These results suggest the need to complement robustness certificates with additional OOD detection measures for practical use in real applications. Overall, verification methods can contribute to trustworthy AI, and future research could explore combining them with standard OOD detection methods to distinguish between the two for a given sample.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision – ECCV 2020*, pages 484–501, Cham, 2020. Springer International Publishing. 1

[2] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018. 5

[3] Mislav Balunovic, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin Vechev. Certifying geometric robustness of neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2, 3, 4

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013. 1

[5] Julian Bitterwolf, Alexander Meinke, and Matthias Hein. Certifiably adversarially robust detection of out-of-distribution data. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 1, 3

[7] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018. 7

[8] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 7

[9] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 2, 7

[10] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 3

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[12] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2018. 1

[13] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2, 3, 7

[14] Divya Gopinath, Guy Katz, Corina S. Păsăreanu, and Clark Barrett. Deepsafe: A data-driven approach for assessing robustness of neural networks. In *Automated Technology for Verification and Analysis*, pages 3–19, Cham, 2018. Springer International Publishing. 3

[15] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 41–50. Computer Vision Foundation / IEEE, 2019. 3, 5, 6

[16] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. 3

[17] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. 6

[18] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 3, 7

[19] Maximilian Henne, Adrian Schwaiger, Karsten Roscher, and Gereon Weiss. Benchmarking uncertainty estimation methods for deep learning with safety-related metrics. In *SafeAI@ AAAI*, pages 83–90, 2020. 3

[20] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013. 4, 7

[21] Nikola Jovanovic, Mislav Balunovic, Maximilian Baader, and Martin T. Vechev. Certified defenses: Why tighter relaxations may hurt training? *CoRR*, abs/2102.06700, 2021. 1, 2, 3

[22] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pages 97–117. Springer, 2017. 1

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009. 7

[24] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 7

[25] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177, 2018. 6

[26] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 6

[27] Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *Advances in Neural Information Processing Systems*, 33:21476–21487, 2020. 2, 3

[28] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 6

[29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2, 3, 4, 7

[30] Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 3

[31] Alexander Meinke, Julian Bitterwolf, and Matthias Hein. Provably robust detection of out-of-distribution data (almost) for free. *arXiv preprint arXiv:2106.04260*, 2021. 3

[32] Christoph Müller, François Serre, Gagandeep Singh, Markus Püschel, and Martin Vechev. Scaling polyhedral neural network verification on gpus. In *Proceedings of Machine Learning and Systems*, pages 733–746, 2021. 1, 2, 3, 6, 7, 8

[33] Mark Niklas Müller, Franziska Eckert, Marc Fischer, and Martin Vechev. Certified training: Small boxes are all you need. *arXiv preprint arXiv:2210.04871*, 2022. 2

[34] Mark Niklas Müller, Gleb Makarchuk, Gagandeep Singh, Markus Püschel, and Martin Vechev. Prima: General and precise neural network certification via scalable convex hull approximations. *Proc. ACM Program. Lang.*, 6(POPL), 2022. 1, 2, 3, 7

[35] Pierre Sermanet, Soumith Chintala, and Yann LeCun. Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 3288–3291. IEEE, 2012. 7

[36] Arvind Kumar Shekar, Liang Gou, Liu Ren, and Axel Wendt. Label-free robustness estimation of object detection cnns for autonomous driving applications. *International Journal of Computer Vision*, 129:1185–1201, 2021. 1

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 1

[38] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In *Advances in Neural Information Processing Systems*, pages 29909–29921. Curran Associates, Inc., 2021. 1, 2, 3

[39] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018. 2

[40] Haoze Wu, Teruhiro Tagomori, Alexander Robey, Fengjun Yang, Nikolai Matni, George Pappas, Hamed Hassani, Corina Pasareanu, and Clark Barrett. Toward certified robustness against real-world distribution shifts. *arXiv preprint arXiv:2206.03669*, 2022. 3

[41] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 7

[42] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021. 7

[43] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in neural information processing systems*, 33:8588–8601, 2020. 1, 2, 3