SoK: Certified Robustness for Deep Neural Networks

Linyi Li* Tao Xie† Bo Li*
* University of Illinois Urbana-Champaign, {linyi2,lbo}@illinois.edu
† Key Laboratory of High Confidence Software Technologies, MoE (Peking University), taoxie@pku.edu.cn

2023 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-9336-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SP46215.2023.10179303

***Abstract*—Great advances in deep neural networks (DNNs) have led to state-of-the-art performance on a wide range of tasks. However, recent studies have shown that DNNs are vulnerable to adversarial attacks, which have brought great concerns when deploying these models to safety-critical applications such as autonomous driving. Different defense approaches have been proposed against adversarial attacks, including: a) *empirical defenses*, which can usually be adaptively attacked again without providing robustness certification; and b) *certifiably robust approaches*, which consist of *robustness verification* providing the lower bound of robust accuracy against *any* attacks under certain conditions and corresponding *robust training* approaches. In this paper, we systematize certifiably robust approaches and related practical and theoretical implications and findings. We also provide the *first* comprehensive benchmark on existing robustness verification and training approaches on different datasets. In particular, we 1) provide a taxonomy for the robustness verification and training approaches, as well as summarize the methodologies for representative algorithms, 2) reveal the characteristics, strengths, limitations, and fundamental connections among these approaches, 3) discuss current research progresses, theoretical barriers, main challenges, and future directions for certifiably robust approaches for DNNs, and 4) provide an open-sourced unified platform to evaluate 20+ representative certifiably robust approaches.**

***Index Terms*—certified robustness, neural networks, verification**

I. Introduction

Machine learning (ML) techniques, especially deep neural networks (DNNs), have been widely adopted in various applications, such as image classification [1]–[3] and natural language processing [4]–[6]. However, despite their wide applications, both traditional ML models [7]–[9] and DNNs [10], [11] are shown vulnerable to adversarial evasion attacks where carefully crafted *adversarial examples* — inputs with adversarial perturbations — could mislead ML models to make arbitrarily incorrect predictions [12], [13]. The existence of adversarial attacks leads to great safety concerns for DNN-based applications, especially in safety-critical scenarios such as autonomous driving [14], [15].

To defend against such attacks, there are several works proposed to empirically improve the robustness of DNNs [8], [16]–[20]. However, many of such defenses can be adaptively attacked again by sophisticated attackers [12], [21]. The everlasting competition between attackers and defenders motivates studies on the **certifiably robust approaches** for DNNs, which include both **robustness verification** and **robust training** approaches [22]–[28]. The robustness verification approaches aim to evaluate DNN robustness by providing a theoretically certified lower bound of robustness under certain perturbation constraints; the corresponding robust training approaches aim to train DNNs to improve such lower bound.

In this paper, we aim to provide a taxonomy for existing certifiably robust approaches (i.e., robustness verification and robust training approaches) from the first principle, as well as a comprehensive benchmark on different datasets and models to enable the quantitative comparison for the community. Existing surveys discuss general attacks and defenses for traditional ML models [29]–[32] and DNNs [33]–[36], but they mainly focus on empirical defenses without guarantees or some specific verification approaches. To the best of our knowledge, this is the *first* systematic taxonomy for the fast-developing *certifiably robust* approaches on DNNs against *evasion* attacks. The taxonomy reveals characteristics, strengths, limitations, and fundamental connections among these approaches.

To provide quantitative analysis for existing certifiably robust approaches, we develop an open-source unified toolbox for representative verification and training approaches. We benchmark over 20 verification and robust training approaches. As far as we know, it is the *first* large-scale benchmark for the certified robustness of DNNs. Based on the taxonomy, analysis, and benchmark of existing approaches, we further provide discussion and analysis on current research progresses, theoretical barriers, and several promising future directions. We also outline how to extend these approaches to alternative threat models and system models along with their applications.

This SoK is intended for both ML experts, who aim to develop and improve certifiably robust ML approaches, as well as practical users with a focus on applying certifiably robust approaches to different real-world ML applications. For ML experts, this SoK provides (1) systematic taxonomy to contextualize their works, (2) detailed explanation and analysis/comparison for representative certifiably robust approaches, and (3) discussion of research implications, including limitations, challenges, and future directions. For practical users, the SoK provides (1) formal problem definition of robustness verification, (2) comprehensive benchmark and reference implementations of representative approaches to ease the deployment, and (3) practical implications on how to select the most suitable defenses and how to evaluate existing defenses using certifiably robust approaches.

In taxonomizing and analyzing certifiably robust approaches for DNNs, we make the following contributions:

- We provide a general problem definition for the robustness

DOI 10.1109/SP46215.2023.00006

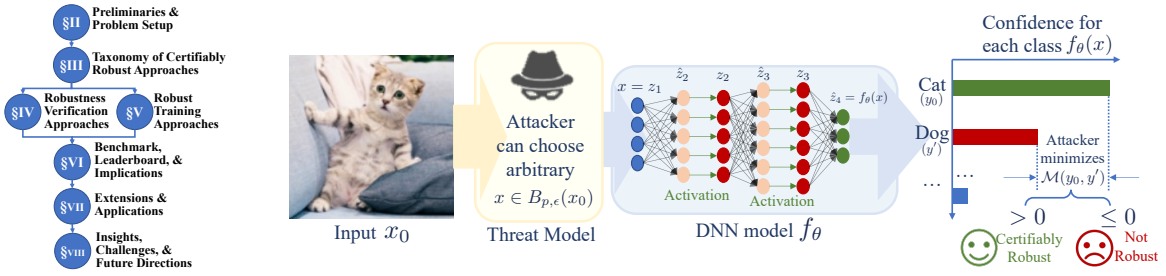


Fig. 1. **(a)** Overview of this SoK; **(b)** Illustration of robustness verification as optimization (Problem 1).

verification problem and the *first* systematic taxonomy of certifiably robust approaches for DNNs (Sec. III), including the robustness verification approaches (Sec. IV), and robust training approaches (Sec. V).

- We conduct extensive quantitative comparisons[1] for different state-of-the-art approaches on robustness verification and robust training, leading to a benchmark and leaderboard, from which we summarize practical implications for deploying certifiably robust approaches (Sec. VI).
- We provide an open-source unified evaluation toolbox for over 20 verification and training approaches, which we believe will facilitate the development and evaluation of research on certified robustness for DNNs.[1]
- We discuss and analyze current research progresses, theoretical barriers, challenges, extensions, and further provide several potential future research directions (Secs. VII and VIII).

II. Preliminaries and Problem Setup

In this section, we provide the preliminaries and a general problem definition for robustness verification. We denote $[n]$ as set $\{1, 2, \ldots, n\}$. To represent the region of adversarially perturbed input, when measured by ℓ_p norm ($p \in \mathbb{N}_+ \cup \{+\infty\}$) we use $B_{p,\epsilon}(x_0)$ to denote the perturbed input which is drawn from the region centered at x_0 with ϵ radius, i.e., $B_{p,\epsilon}(x_0) := \{x : \|x - x_0\|_p < \epsilon\}$, where ϵ is called perturbation radius.

A. System Model

We focus on the certified robustness of DNNs for classification tasks for brevity and ease of exposition. Extensions to other system models and other tasks are discussed in Sec. VII.

A (classification) DNN model f_θ is formulated as a function: $\mathcal{X} \to \mathbb{R}^C$, where the input data $x \sim \mathcal{D}$ is in a bounded n-dimensional subspace $\mathcal{X} \subseteq [0, 1]^n$, and the model provides confidence scores for all C classes. $F_\theta(x) := \arg\max_{i \in [C]} f_\theta(x)_i$ is the predicted class of model f_θ given input x. θ is the set of trainable parameters for f_θ. For brevity, we may omit θ when there is no ambiguity. There are many different DNN architectures. One common system model is feed-forward ReLU networks as defined in Def. 1.

Definition 1 (Feed-Forward ReLU Networks)**.** *An l-layer feed-forward ReLU network f_θ is defined as such:*

$$\begin{cases} z_1 := x, \\ \hat{z}_{i+1} := \boldsymbol{W}_i z_i + b_i, & \text{for } i = 1, 2, \ldots, l-1 \\ z_i := \mathrm{ReLU}(\hat{z}_i), & \text{for } i = 2, \ldots, l \\ f_\theta(x) := \hat{z}_l, \end{cases} \tag{1}$$

where $\mathrm{ReLU}(z, 0) = \max\{z, 0\}$. *Each z_i and $\hat{z}_i$ is a vector in $\mathbb{R}^{n_i}$. In particular, $n_1 = n$ and $n_l = C$. The trainable parameters $\theta := \{\boldsymbol{W}_i, b_i : i \in [l-1]\}$.*

In Table I, the "System Model" column lists some other system models which we will define when illustrating their corresponding verification approaches in Sec. IV.

B. Threat Model

Existing studies on certified robustness [18], [22], [27], [37], [38] mainly aim to defend against *white-box* ***evasion*** attacks, which indicate the strongest adversaries who have full knowledge of the target model, including its parameters and architecture. In particular, the adversary would carefully craft a bounded perturbation to the original input, generating an *adversarial example* [11] to fool the model into making incorrect predictions. Formally, we define (ℓ_p, ϵ)*-adversary*.

Definition 2 ((ℓ_p, ϵ)-Adversary)**.** *For given input (x_0, y_0), where $x_0 \in \mathcal{X}$ is the input instance and $y_0 \in [C]$ is its true label, the* (ℓ_p, ϵ)-adversary *will generate a perturbed input $x \in B_{p,\epsilon}(x_0)$, such that $F_\theta(x) \neq y_0$. When there is no ambiguity, we will call it* ℓ_p adversary.

We focus on certifiably robust approaches against (ℓ_p, ϵ)-adversary, since approaches for this adversary are well-developed, and approaches for other threat models can be extended from those for (ℓ_p, ϵ)-adversary. We will discuss other threat models in Sec. VII. The above definition conforms to untargeted attack whose goal is to deviate the model prediction from the ground truth. The targeted attack which aims to mislead the model to output a specific label y', can be defined similarly. The literature also refers to an (ℓ_p, ϵ)-adversary as an ℓ_p*-bounded attack* (bounded by ϵ). To the best of our knowledge, existing ℓ_p-bounded attacks only consider $p = 1, 2, \infty$. The inputs generated by these adversaries are within ϵ distance to clean input x_0 measured by ℓ_1 norm (i.e., Matthattan distance), ℓ_2 norm (i.e., Euclidean distance), and ℓ_∞ norm (i.e., maximum difference among all dimensions) respectively. We illustrate the region from which the attacker picks the perturbed input in Fig. 5 in App. B-A. For 2D input, the region shapes are diamond, circle, and square for ℓ_1, ℓ_2, and ℓ_∞ adversaries respectively.

[1]The benchmark website with open-source toolbox, including full results are available at *https://sokcertifiedrobustness.github.io*.

C. Robustness Verification and Robust Training

Robustness verification. A *robustness verification* approach certifies the lower bound of model's performance against *any* adversary under certain constraints, e.g., ℓ_∞-bounded attack. We can categorize the verification approaches into *complete verification* and *incomplete verification*. When the verification approach outputs "not verified" for a given x_0, if it is guaranteed that an adversarial example x around x_0 exists we call it *complete verification*; and otherwise *incomplete verification*.

We can also categorize the verification approaches into *deterministic verification* and *probabilistic verification*. When the given input is non-robust against the attack, deterministic verification is guaranteed to output "not verified"; and the probabilistic verification is guaranteed to output "not verified" with a certain probability (e.g., 99.9%) where the randomness is independent of the input. Formal definitions are as follows.

Definition 3 (Robustness Verification)**.** *An algorithm $\mathcal{A}$ is called a* robustness verification*, if for any (x_0, y_0), as long as there exists $x \in B_{p,\epsilon}(x_0)$ with $F_\theta(x) \neq y_0$ (adversarial example), $\mathcal{A}(f_\theta, x_0, y_0, \epsilon) = \text{false}$ (*deterministic verification*) or $\Pr[\mathcal{A}(f_\theta, x_0, y_0, \epsilon) = \text{false}] \geq 1 - \alpha$ (*probabilistic verification*), where α is a pre-defined small threshold. If $\mathcal{A}(f_\theta, x_0, y_0, \epsilon) = \text{true}$, we call $\mathcal{A}$ provides **robustness certification** for model f_θ on (x_0, y_0) against (ℓ_p, ϵ)-adversary. Whenever $\mathcal{A}(f_\theta, x_0, y_0, \epsilon) = \text{false}$, if there exists $x \in B_{p,\epsilon}(x_0)$ with $F_\theta(x) \neq y_0$, $\mathcal{A}$ is called **complete verification**, otherwise **incomplete verification**.*

If we view "certifying a truly robust instance" as the true positive, then a robustness verification approach produces false positives with small (probabilistic) or zero (deterministic) probability, and complete verification produces no false negatives. If the verification cannot certify an instance, it is possible that either the instance is not robust or the verification approach is too loose to certify it. We can also view robustness verification from optimization perspective.

Problem 1 (Robustness Verification as Optimization)**.** *Given a neural network $f_\theta : \mathcal{X} \to \mathbb{R}^C$, input instance $x_0 \in \mathcal{X}$, ground-truth label $y_0 \in [C]$, any other label $y' \in [C]$ and the radius $\epsilon > 0$, we define the following optimization problem:*

$$\mathcal{M}(y_0, y') = \text{minimize}_x \; f_\theta(x)_{y_0} - f_\theta(x)_{y'} \;\; s.t. \;\; x \in B_{p,\epsilon}(x_0).$$

If $\mathcal{M}(y_0, y') > 0, \forall y' \in [C] \setminus \{y_0\}$, f_θ is certifiably robust at x_0 within radius ϵ w.r.t. ℓ_p norm.

Intuitively, Problem 1 searches for the minimum margin between the model confidence for the true class y_0 and any other class y'. For any $y' \neq y_0$, if we can certify $\mathcal{M}(y_0, y') > 0$, the margin $f_\theta(x)_{y_0} - f_\theta(x)_{y'}$ is always positive. Since the model will predict the class with the highest confidence, this means for any possible perturbed input $x \in B_{p,\epsilon}(x_0)$, the predicted class is always y_0, and therefore the robustness is certified. Fig. 1b illustrates this process.

The robustness verification then boils down to deciding whether $\mathcal{M}(y_0, y') > 0$. If a procedure exactly solves Problem 1, the corresponding verification approach is complete. If a procedure conservatively provides a lower bound of $\mathcal{M}$, the corresponding verification approach is usually incomplete.

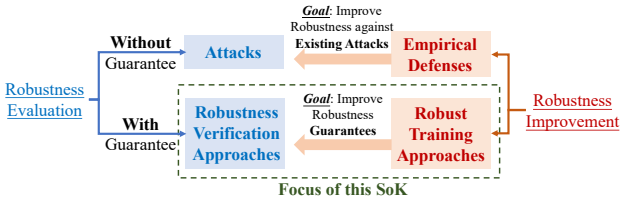


Fig. 2. Different approaches for evaluating and improving DNN robustness against evasion attacks.

Although complete verification sounds attractive, it is NP-Complete [23], [39]. This intrinsic barrier, which we identify as **scalability challenge**, impedes complete verification approaches from scaling up to common DNN sizes. To overcome this scalability challenge, incomplete verification is studied, aiming to solve the relaxed problem, i.e., computing the lower bound of $\mathcal{M}$ which is more tractable. However, the relaxations in existing approaches are typically too loose, which induces another problem identified as the **tightness challenge**. For example, the widely-used linear relaxations are shown significantly looser than complete verification in practice [40]. Theoretically, if complete verification can certify robustness radius ϵ_0, unless NP = P, there is no polynomial-time verification that can guarantee a constant fraction between its certified robustness radius and ϵ_0 [39]. The trade-off between scalability and tightness, i.e., either scalability or tightness can be achieved but not both, constitutes the main obstacle for robustness verification.

Robust training. Given the scalability and tightness challenges, vanilla DNNs are challenging to verify, where verification approaches either need a long running time or output trivial bounds. To enhance the certifiability, many robust training approaches are proposed, which are typically related to or derived from corresponding verification by optimizing verification-inspired regularization terms or injecting specific data augmentation during training. In practice, after robust training, the model usually achieves high certified robustness. Thus, robust training is a strong complement to robustness verification approaches.

Relationship with empirical attacks and defenses. Towards evaluating and improving DNN robustness, another active line of research is attacks and empirical defenses. Strong white-box attacks, such as CW attack [103], PGD attack [18], and AutoAttack [104], are widely used to evaluate DNN robustness (e.g., [105], [106]). To improve model robustness against these attacks, many empirical defenses are proposed, such as adversarial training [18], [107]–[109] and TRADES [106]. As illustrated in Fig. 2, both attacks and verification approaches can be used to evaluate DNN robustness, but verification approaches can provide robustness guarantees against any possible future attacks; both empirical defenses and robust training approaches can improve DNN robustness, but empirical defenses aim to improve robustness against existing attacks and robust training approaches aim to improve robustness guarantees. We note that: (1) The strongest attack (which always discovers adversarial

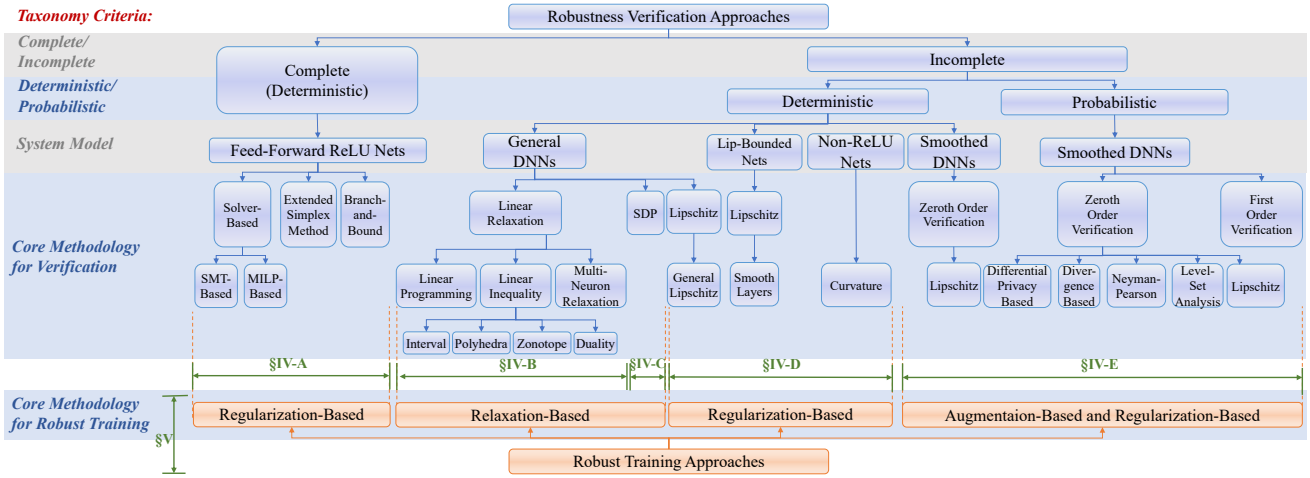


Fig. 3. **Taxonomy** of certifiably robust approaches. Blue boxes show the taxonomy of verification approaches. Orange boxes show the taxonomy of robust training approaches, and vertical dotted lines show the suitable verification for corresponding training approaches. Left columns list taxonomy criteria per level. Sections illustrating corresponding approach category are shown as green segments.

TABLE I

PROPERTIES AND REFERENCES OF ROBUSTNESS VERIFICATION APPROACHES. NOTATIONS ARE EXPLAINED IN SEC. III.

Complete/ Incomplete	Deterministic/ Probabilistic	System Model	Robustness Verification Approaches			Supported ℓ_p ℓ_∞	ℓ_2	ℓ_1	Scalability (Scale up to)	Scalability (Complexity)	Tightness	References
Complete	Deterministic	for Feed-Forward ReLU Nets	Solver-Based	SMT-Based		✓	✓	✓	MNIST	$O(2^{lw})$	Complete	[41], [42]
				MILP-Based		✓			CIFAR-10	$O(2^{lw})$	Complete	[43]–[46]
			Extended Simplex Method			✓			MNIST	$O(2^{lw})$	Complete	[23], [47]
			Branch-and-Bound			✓	✓		CIFAR-10	$O(2^{lw})$	Complete	[26], [37], [48]–[54] [55]–[60]
Incomplete	Deterministic	for General DNNs[1]	Linear Relaxation	Linear Programming (LP)		✓	(✓)	(✓)	CIFAR-10	$O(\text{poly}(l,w))$	T_5[2]	[39], [40]
				Linear Inequality	Interval	✓	(✓)	(✓)	Tiny ImageNet	$O(lw^2)$	T_2	[61]
					Polyhedra	✓	(✓)	(✓)	Tiny ImageNet	$O(lw^3)$	T_4[2]	[38], [39], [62]–[65]
					Zonotope	✓	(✓)	(✓)	Tiny ImageNet	$O(lw^3)$	T_3[2]	[25], [66]–[68]
					Duality	✓	(✓)	(✓)	Tiny ImageNet	$O(lw^3)$	T_4[2]	[27], [69]–[71]
				Multi-Neuron Relaxation		✓	(✓)	(✓)	CIFAR-10	$O(lw^3)$ - $O(2^{lw})$[6]	T_7	[24], [72]–[74]
			Semidefinite Programming (SDP)			✓			CIFAR-10	$O(\text{poly}(l,w))$	T_6	[75]–[79]
		for Lip-Bounded Nets	Lipschitz	General Lipschitz			✓		Tiny ImageNet	$O(lw^2)$	T_1[3]	[11], [39], [80]–[84]
				Smooth Layers		✓	✓		Tiny ImageNet	$O(lw^2)$	[3]	[85]–[89]
		for Non-ReLU Nets[4]	Curvature				✓		CIFAR-10	$O(lw^3)$	[4]	[90]
		for Smoothed DNNs	Zeroth Order	Lipschitz				✓	ImageNet	$O(Slw^2)$	[5]	[91]
	Probabilistic	for Smoothed DNNs	Zeroth Order	Differential Privacy Inspired			✓	✓	ImageNet	$O(Slw^2)$	ST_1	[92]
				Divergence Based			✓	✓	ImageNet	$O(Slw^2)$	ST_2	[93], [94]
				Neyman Pearson			✓		ImageNet	$O(Slw^2)$	ST_3	[22]
				Level-Set Analysis		(✓)	✓	✓	ImageNet	$O(Slw^2)$	ST_3	[28], [95], [96]
				Lipschitz		(✓)	✓		ImageNet	$O(Slw^2)$	ST_3	[97], [98]
			First Order			(✓)	✓	✓	ImageNet	$O(Slw^2)$	ST_4	[99], [100]

1. Typical approaches mainly support feed-forward ReLU networks, but extensions to general DNNs are available [38], [101], [102], which are discussed in Sec. VII.
2. Tightness depends on intermediate layer bounds. If they share the same intermediate layer bounds, the tightness order is Zonotope $<$ Polyhedra = Duality $<$ LP [40].
3. Lipschitz bound is loose for typical DNNs, but can be tight for specially regularized DNNs which have small Lipschitz bounds.
4. Only available for networks whose activation functions have nonzero second-order derivatives, which exclude ReLU networks. Thus, tightness is incomparable with others.
5. The approach is designed for some specific smoothing distributions that are not supported by other smoothed DNN oriented approaches.
6. Tunable time complexity dependent on the upper limit of number of linear constraints.

example if exists) is the strongest verification (complete verification). For robustness evaluation, attack and verification can be viewed as approaching from two sides (over-estimation and under-estimation) to the same goal (precise evaluation). (2) Complete verification approaches can be used to evaluate and compare empirical defenses on small models (not on large models due to scalability challenges). In practice, models trained with strong empirical defenses can be certified to have high robustness by complete verification [58]. In contrast, most incomplete verification cannot certify high robustness for empirically defended models. More discussion is in Sec. V.

III. TAXONOMY OF CERTIFIABLY ROBUST APPROACHES

In this section, we provide a comprehensive taxonomy of existing robustness verification and robust training approaches (Fig. 3), and characterize their properties (Table I).

Taxonomy of robustness verification and robust training. In Fig. 3, we present a taxonomy of existing robustness verification and robust training approaches. In the taxonomy, the first-level is "complete vs. incomplete", and the second-level is "deterministic vs. probabilistic". These concepts are as defined in Sec. II-A. Note that there is no complete and probabilistic verification approach yet. In the third-level, we categorize verification approaches based on the system model. Under the third level, we categorize verification approaches by their core methodologies. We will illustrate verification approaches in detail in Sec. IV. The robust training approaches are shown in orange. Based on their core methodologies, there are three categories: regularization-based, relaxation-based, and augmentation-based approaches. We will illustrate robust training approaches in detail in Sec. V.

Properties of verification approaches. In Table I, we summarize the key properties of each verification approach, including the system model, the supported ℓ_p adversary types, scalability, and its verification tightness.

For the **supported** ℓ_p in Table I, "✓" means well-supported ℓ_p adversaries, "(✓)" means supported adversaries but the

verification is not as tight as others, and empty means unsupported adversaries. To measure the **scalability**, we use "*the largest dataset (in terms of input dimension) that has been demonstrated feasible to certify by existing work using the corresponding verification approach under radius* $\epsilon \geq 1/255$" as the criterion. The threshold $1/255$ is the smallest considered ϵ we have seen in the literature for these image datasets. The dataset effectively measures scalability. For example, the approach scaling up to ImageNet is more scalable than the one to MNIST. We also provide a quantitative measure: the best known time complexity for verifying an arbitrary input, given an arbitrary network with depth l, width w in terms of neurons, and sampling number S (for smoothed DNNs). For **tightness**, the tightest verification approaches are complete ones. For incomplete approaches, we rank the tightness by our benchmark results shown in Sec. VI, or empirical observations and theoretical results from published papers. General DNNs are ranked by T_n and smoothed DNNs are ranked by ST_n where larger n means tighter approaches. T_n- and ST_n-denoted approaches are incomparable since their system models are different. More discussion on the scalability and tightness measurements are in App. A.

From Table I, we observe that for general DNNs, complete or tight deterministic approaches ($\geq T_5$) can only handle CIFAR-10-sized models, and only looser verification can go beyond this scale. For large ImageNet-sized models, verification for general DNNs cannot support such a scale yet; only approaches for smoothed DNNs can, while they cannot provide nontrivial verification against ℓ_∞ adversary yet. This reflects the fundamental trade-off between scalability and tightness challenge. We will discuss this further in Sec. VI.

IV. Robustness Verification Approaches

We illustrate representative verification approaches in this section: complete verification (Sec. IV-A); incomplete verification, including linear relaxation-based (Sec. IV-B), SDP (Sec. IV-C), Lipschitz-/curvature-based (Sec. IV-D), and probabilistic approaches (Sec. IV-E). We conclude each subsection by highlighting the implications. We summarize practical and research implications in Secs. VI-C and VIII respectively.

A. Complete Verification

Here we illustrate complete robustness verification approaches, which usually consider ℓ_∞ adversary and support feed-forward ReLU networks (see Def. 1). All these complete verification approaches have worst-case exponential time complexity due to the hardness of verification [23], [39], but some of them perform well in practice, being able to verify DNNs with several thousands of neurons [46], [58]. Many complete verification approaches rely on the neuron activation patterns, so below we first categorize neurons by their activation patterns.

Definition 4 (Stable and Unstable Neurons)**.** *Let* $z = \mathrm{ReLU}(\hat{z})$ *be a neuron in a feed-forward ReLU network. For a given input* x*, if the input* $\hat{z} < 0$*, we call* z inactive*, otherwise* active*. Let* $\mathcal{S}$ *be an input region, for any* $x \in \mathcal{S}$*, if we can certify that input* $\hat{z}$ *is always* > 0 *or* ≤ 0*, we call neuron* z stable *in region* $\mathcal{S}$*; otherwise we call* z unstable*.*

Remark. When a neuron z is stable, it serves as a linear mapping $x \mapsto 0$ (inactive neuron) or $x \mapsto x$ (active neuron).

*1) **Solver-Based Verification** [41]–[46]:* By inspecting the definition of feed-forward neural networks (Def. 1), we can observe that the DNN $f_\theta(x)$ is defined by the sequential composition of affine transformations and ReLU operations. Both affine transformations and ReLU operations can be encoded by a conjunction of linear inequalities. For example, $z_{i,j} = \mathrm{ReLU}(\hat{z}_{i,j}) \iff ((\hat{z}_{i,j} < 0) \wedge (z_{i,j} = 0)) \vee ((\hat{z}_{i,j} \geq 0) \wedge (z_{i,j} = \hat{z}_{i,j}))$. Thus, general-purpose SMT solvers such as Z3 [110] can be directly applied to solve the satisfiability problem of boolean predicate $\bigwedge_{y' \in [C]: y' \neq y_0} (\mathcal{M}(y_0, y') > 0)$ (see Problem 1), which yields a solution to complete verification. However, SMT-based verification is generally not scalable [41], [42] and can verify DNNs with only hundreds of neurons, which is too small even for the simple MNIST dataset.

Another way is to encode the verification problem as a mixed-integer linear programming (MILP) problem. In MILP, the constraints are linear inequalities and the objective is a linear function. However, different from linear programming (LP), in MILP we can constrain some variables to take only integer values instead of real numbers. This additional expressive power allows MILP constraints to encode the non-linear ReLU operations and the whole DNN model [43], [45]. Thus, the verification problem can be precisely encoded as an MILP problem. By leveraging efficient MILP solvers such as Gurobi [111], MILP-based verification is feasible on medium-sized CIFAR-10 models if the model is specifically trained to favor certifiability [61], [71], [112], [113]. However, the naturally trained or empirically defended DNNs are still hard to verify by these approaches even on MNIST [46].

*2) **Extended Simplex Method** [23], [47]:* The DNN model is composed of affine transformations and ReLU operations which correspond to linear constraints and ReLU constraints respectively. When there are only linear constraints, the verification problem is a linear programming problem and can be effectively solved by the simplex method [114]. In [23], [47], the simplex method is extended to handle ReLU constraints. The core idea is to iteratively check whether the ReLU constraints are violated and fix them. If the violation cannot be easily fixed, we split the neuron into active and inactive and solve subproblems respectively.

*3) **Branch-and-Bound** [26], [37], [48]–[50], [52]–[55], [57], [58], [60], [73]:* Another line of complete verification is branch-and-bound. Most competitors in VNN-COMP, an annual DNN verification competition, build their verification tools based on branch-and-bound [115], and the winner tool of VNN-COMP 2021 and 2022, α-β-CROWN [58], [60], is based on branch-and-bound. The branch-and-bound verification relies on the **piecewise-linear property** of DNNs: Since each ReLU neuron outputs $\mathrm{ReLU}(x) = \max\{x, 0\}$, it is always locally linear within some region around input x. Since feed-forward ReLU networks are the composition of these piecewise linear neurons and (linear) affine mappings, *the output is locally linear w.r.t. input* x. This property is formally stated and proved in [55].

It serves as the foundation for branch-and-bound verification.

Given an input x_0 with true class label y_0 and perturbation radius ϵ, recall that the verification problem can be reduced to deciding whether $\mathcal{M}(y_0, y') > 0$ for any $y' \in [C] \setminus \{y_0\}$ (see Problem 1). A branch-and-bound verification approach first applies incomplete verification to derive a lower bound and an upper bound of $\mathcal{M}(y_0, y')$: if the lower bound is positive then terminate with "verified"; if the upper bound is non-positive then terminate with "not verified"—**bounding**. Otherwise, the approach recursively chooses a neuron $z_{i,j} = \mathrm{ReLU}(\hat{z}_{i,j})$ to split into two branches: $\hat{z}_{i,j} < 0$ (inactive branch) and $\hat{z}_{i,j} \geq 0$ (active branch)—**branching**. For inactive branch, we have the constraint $\hat{z}_{i,j} < 0, z_{i,j} = 0$; and for active branch, we have the constraint $\hat{z}_{i,j} \geq 0, z_{i,j} = \hat{z}_{i,j}$. Therefore, for each branch the neuron brings only linear constraints, and we again apply incomplete verification to determine whether $\mathcal{M}(y_0, y') > 0$ for each branch. If for both branches, we can verify that $\mathcal{M}(y_0, y')$ is always positive/negative or the branching condition is infeasible, the verification terminates; otherwise, we further split other neurons recursively. When all neurons are split, the branch will contain only linear constraints, and thus the approach applies linear programming to compute the precise $\mathcal{M}(y_0, y')$ and verify the branch. The branch-and-bound framework is formalized in [49], [50], [52], and opens a wide range of design choices, leading to approaches with different implementation and scalabilities. Some verification approaches efficiently traverse the piecewise linear regions around the clean input x_0 to exhaustively search adversarial examples in the region $B_{p,\epsilon}(x_0)$ [48], [54], [55], which work better under ℓ_2 adversary while other branch-and-bound approaches work better under ℓ_∞ adversary, because under ℓ_2 adversary input region has special geometric properties that can be exploited for traversal-based approaches [54].

Practical implications. Though complete verification approaches have worst-case exponential time complexity, they, especially branch-and-bound approaches such as α-β-CROWN [58], [60], can verify DNNs with up to 10^5 neurons like ResNet [1] in practice within several minutes, if the model is specifically trained to favor certifiability. This model size is of moderate level on CIFAR-10. For models that are not specifically trained, complete verification can handle DNNs with up to 10^4 neurons and roughly 6 layers, corresponding to small models on CIFAR-10 and large models on MNIST. Therefore, for simple tasks and simple DNNs, such as those for aircraft control systems [23], [25], [26], [47], [48], [116], [117], it is feasible to deploy complete verification to verify the robustness. The branch-and-bound approaches are more scalable than solver-based and extended simplex approaches. A more comprehensive comparison of existing verifiers can be found in VNN-COMP 2021 report [115].

Research implications. For complete verification, the primary research goal is to develop more scalable approaches. For solver-based verification, it is important to find a more solver-friendly problem encoding or design specific optimizations tailored for DNN verification inside the solver [113]. For branch-and-bound verification, it is a popular direction to find an efficient incomplete verification heuristic for the bound computation that has a better trade-off between tightness and efficiency [49], [58], [60], [73]. In addition, finding a new branching heuristic, either rule-based or learning-based, is a promising direction to boost the scalability of complete verification.
Recently, Zombori et al [118] and Jia and Rinard [119] discovered that some complete verification approaches are unsound under floating-point arithmetic and such unsoundness can be exploited to fool the verifier. Thus, future approach developers should consider floating-point rounding errors.

B. Incomplete Verification via Linear Relaxation

Due to the scalability barrier of complete verification, many incomplete verification approaches based on relaxations are proposed. Among them, linear relaxations are well studied. This category of verification approaches runs much faster and many can scale up to large ResNet models on Tiny ImageNet, which contain around 10^5 neurons [65].

Linear relaxation based approaches rely on ReLU polytope, which we define below and illustrated in App. B-B.

Definition 5 (Polytope for Unstable ReLU)**.** *For neuron $z_{i,j} = \mathrm{ReLU}(\hat{z}_{i,j})$, let $l_{i,j}$ and $u_{i,j}$ be the lower bound and upper bound of its output when the input region is $B_{p,\epsilon}(x_0)$:*

$$l_{i,j} \leq \min_{x \in B_{p,\epsilon}(x_0)} \hat{z}_{i,j}(x) \leq \max_{x \in B_{p,\epsilon}(x_0)} \hat{z}_{i,j}(x) \leq u_{i,j}. \tag{2}$$

*Then, if $l_{i,j} < 0 < u_{i,j}$, the **unstable neuron** $z_{i,j} = \mathrm{ReLU}(\hat{z}_{i,j})$ can be bounded by following linear constraints:*

$$z_{i,j} \geq 0,\ z_{i,j} \geq \hat{z}_{i,j},\ z_{i,j} \leq \frac{u_{i,j}}{u_{i,j} - l_{i,j}}(\hat{z}_{i,j} - l_{i,j}). \tag{3}$$

These constraints define a region called ReLU polytope*.*

When both $l_{i,j}$ and $u_{i,j}$ are tight, the polytope is the tightest convex hull for this neuron. For stable ReLU, linear constraint $z_{i,j} = 0$ or $z_{i,j} = \hat{z}_{i,j}$ defines its linear relaxation.

In general, all linear relaxation based approaches require computing $l_{i,j}$ and $u_{i,j}$ (see Def. 5) for each neuron $z_{i,j}$, then they compute an over-approximation bound $\mathcal{S}$ for the region $f_\theta\left(B_{p,\epsilon}(x_0)\right) := \{f_\theta(x) : x \in B_{p,\epsilon}(x_0)\}$, i.e., $\mathcal{S} \supseteq f_\theta\left(B_{p,\epsilon}(x_0)\right)$. $\mathcal{S}$ is described by linear constraints so that it is easy to verify whether all points in $\mathcal{S}$ lead to the true class y_0. If it is true, the region $f_\theta\left(B_{p,\epsilon}(x_0)\right)$ is robust.

*1) **Linear Programming** [39], [40]:* Based on Def. 5, we can directly use the polytope shown in Fig. 6a in App. B-B as the relaxation for verification, which results in the approach named LP-FULL [39], [40]. In LP-FULL, l and u are computed layer by layer, the polytope relaxation (Def. 5) is then applied for each ReLU neuron, and finally, the verification is performed by solving the resulting linear programming (LP) problem. Due to the relaxation, we obtain a lower bound of $\mathcal{M}(y_0, y')$ in Problem 1. Even though LPs can be solved in polynomial time, in practice, solving LP is still expensive. Applying LP-FULL on a typical model on CIFAR-10 for verifying a single instance takes several hours to several days [40]. Moreover, although LP-FULL is the tightest verification using single neuron linear relaxations, compared with complete verification, the certified robustness radius ϵ is usually $1.5 - 5$ times smaller, which indicates the intrinsic tightness barrier of linear relaxations.

*2) **Linear Inequality** [25], [27], [38], [39], [61]–[67], [69]–[71]:* To circumvent solving expensive LP, further relaxations are applied, which can be divided into *interval* bound propagation (IBP), *polyhedra* abstraction, *zonotope* abstraction, and *duality*-based approaches.

Inteval bound propagation (IBP). A more straightforward and efficient but much looser approach comes from directly propagating l and u defined in Eqn. (2) through the layers of the given DNN model. Given perturbed input region $B_{p,\epsilon}(x_0)$, for the first layer, we have $z_1 = x \in [x_0 - \epsilon, x_0 + \epsilon]$. We let $[l_1, u_1]$ to represent this numerical interval for the first layer z_1. Then, we derive $[l_{i+1}, u_{i+1}]$ for layer z_{i+1} from $[l_i, u_i]$: If $l_k \leq z_k \leq u_k$, based on $\hat{z}_{k+1} = \boldsymbol{W}_k z_k + b_k$, $\hat{z}_{k+1}$ can be bounded by $\hat{l}_{k+1} \leq \hat{z}_{k+1} \leq \hat{u}_{k+1}$ where

$$\hat{l}_{k+1} = \boldsymbol{W}_k^+ l_k + \boldsymbol{W}_k^- u_k + b_k,\ \hat{u}_{k+1} = \boldsymbol{W}_k^+ u_k + \boldsymbol{W}_k^- l_k + b_k.$$

$\boldsymbol{W}^+$ sets negative elements in $\boldsymbol{W}$ to 0; and $\boldsymbol{W}^-$ sets positive elements to 0. Then, $z_{k+1} = \mathrm{ReLU}(\hat{z}_{k+1})$ can be bounded by $l_{k+1} = \max\{\hat{l}_{k+1}, 0\} \leq z_{k+1} \leq \max\{\hat{u}_{k+1}, 0\} = u_{k+1}$.

Through each layer, this bound propagation performs only four matrix-vector products, which are in the same order of model inference. As a result, the approach is very scalable for verifying large models on ImageNet, but on ImageNet it yields trivial bounds due to its looseness. The approach is called IBP [61] or interval arithmetic [57].

As we will discuss in Sec. V, though for normal DNNs, IBP is usually loose. For the models that are specifically trained with IBP, IBP can verify close-to-best certified robustness among linear-relaxation-based approaches. Some work [120] conjectures that IBP bound, though loose, is smoother than other linear relaxations and thus more suitable for training.

Polyhedra abstraction. The polyhedra abstraction based verification approaches, such as FAST-LIN [39], CROWN [38], and DEEPPOLY [64], replace the two lower bounds in the ReLU polytope shown in Eqn. (3) by a single lower bound, resulting in one lower and one upper bound for each neuron respectively. The idea is illustrated in Fig. 6b to 6d in App. B-B. The advantage of using a single linear lower bound is that: (1) the linear bounds can be propagated through layers efficiently instead of solving LP problem—the verification is more scalable than LP; and (2) linear bounds maintain interactions between different components to some degree—the verification is typically tighter than IBP. We call these approaches "polyhedra abstraction based" approaches since they essentially compute polyhedra domain abstraction interpretation [37] for DNNs. We defer technical details along with the illustration of zonotope abstraction and duality-based approaches to App. C.

For all linear inequality based verification approaches, Salman et al [40] prove the *convex barrier*: these approaches cannot be tighter than linear programming based approaches (introduced in Sec. IV-B1).

*3) **Multi-Neuron Relaxation** [24], [72]–[74]:* To circumvent the convex barrier mentioned above, Singh et al [24] and Tjandraatmadja et al [74] found that for ReLU that takes multiple input variables (e.g., $z = \mathrm{ReLU}(x + y)$ takes scalars x and y), if considering multiple input variables together, the tightest convex polytope is tighter than applying single-neuron polytope (Def. 5) along the base direction (i.e., $(x+y)$-direction for $z = \mathrm{ReLU}(x+y)$). Fig. 7 in App. B-C illustrates this observation. Multi-neuron relaxation based approaches are proposed to leverage the multivariate convex relaxations to tighten the verification. Among them, K-RELU [24] and PRIMA [72] consider k ($k \leq 5$) inputs at once. Tjandraatmadja et al [74] point out that tightest convex polytope may contain exponential number of linear constraints, and propose C2V to heuristically find out and only preserve more useful constraints. ACTIVE-SET [73] improves upon C2V with gradient-based optimization and better heuristics on constraint selection. GCP-CROWN [60] extracts convex constraints from MILP solvers and integrate them in linear inequality propagation, which can be viewed as leveraging multi-neuron relaxations in branch-and-bound complete verification.

Practical implications. When the DNN model is too large to be verified by complete verification approaches, linear relaxation based approaches are good options. Among these approaches, IBP is the most scalable and typically the loosest one [61], yielding trivial bounds on normal DNNs. However, on models that are specifically trained for IBP, the IBP certified robustness can be quite satisfactory on large CIFAR-10 and Tiny ImageNet datasets [65], [112], [121]. Multi-neuron relaxation based approaches are the tightest but least scalable ones. Effective heuristics enable multi-neuron relaxation based approaches to significantly improve the scalability at small tightness loss, so they can verify decent robustness on medium-sized CIFAR-10 models with around 10^5 neurons more efficiently than complete verification [72]. But for very deep neural networks (layers $\geq$ 10), due to the amplification of over-approximation, linear relaxation based approaches cannot certify nontrivial robustness. Besides those mentioned above, there are linear relaxation based verification approaches and implementations aiming to efficiently support specific DNN architectures, including CNN [101], residual blocks [71], and other activation functions [38], [65], [72].

Research implications. For linear relaxation based approaches, a tighter and more scalable multi-neuron relaxation based approach is in need. Due to the worst-case exponential number of linear constraints in the tightest convex relaxation, it is important to develop efficient heuristics to synthesize the critical constraints and solve the verification problem with these constraints. Both rule-based heuristics performance [38], [72], [74] and learning-based heuristics [122] are shown effective and are promising directions. Extracting constraints from existing solvers is effective as well [60]. Furthermore, towards the ultimate goal of improving certified robustness, it is also an important topic to develop more efficient robust training approaches for linear relaxation based verification, which we will discuss more in Sec. V.

C. Incomplete Verification via SDP

Semidefinite programming (SDP) can be applied for incomplete verification: VERIFY [79] formulates the robustness verification as an SDP problem, which is a convex optimization problem, where the decision variable is a symmetric and semi-positive matrix whose elements can be linearly constrained. The key formulation in VERIFY is

$$z_{ij} = \mathrm{ReLU}(\hat{z}_{ij}) \Longleftrightarrow \begin{cases} z_{ij} \geq 0, \quad z_{ij} \geq \hat{z}_{ij}, \\ z_{ij} z_{ij}^\mathsf{T} = z_{ij} \hat{z}_{ij}^\mathsf{T}. \end{cases} \tag{4}$$

To handle the quadratic constraint, it defines the vector $v := (1, z_1, \ldots, z_l)^\intercal$ which encodes all ReLU activations, and considers the matrix $\boldsymbol{P} = vv^\intercal$ as the SDP decision variable. Since we replace the constraint $\mathrm{rank}(\boldsymbol{P}) = 1$ coming from $\boldsymbol{P} = vv^\intercal$ by semidefinite constraint $\boldsymbol{P} \succeq 0$, it is a relaxation. Then, the constraints in Eqn. (4) can be directly treated as linear inequality constraints on $\boldsymbol{P}$'s elements and thus can be precisely encoded by the SDP problem along with the optimization objective for verification (Problem 1). Several variants of SDP encoding are studied [76]–[78].

Practical implications. SDP-based verification approaches are incomplete but quite tight—empirically the tightness lies between linear programming based and multi-neuron relaxation based approaches. However, the utility of SDP-based verification approaches is limited by the slow SDP solving process. Even with the specialized solver as in [75], verifying a thousand-neuron level DNN model requires several hours to one day, while complete verification and linear relaxation based approaches can terminate within minutes. Therefore, complete verification or linear relaxation based approaches are recommended instead of SDP-based verification at the current stage.

Research implications. The major challenge of SDP-based verification is its scalability. The improvements for SDP-based verification would come from either more efficient and tighter relaxation [123] or more efficient solvers [75]. For example, the specialized first-order solver proposed in [75] greatly boosts the scalability. Further scalability improvements need to be explored.

D. Incomplete Verification via Lipschitz or Curvature Bounds

Some verification approaches use the Lipschitz bound or curvature bound of DNN function f_θ to verify its robustness.

Definition 6 (Lipschitz Constant)**.** *We say scalar function $g : \mathbb{R}^n \supseteq \mathcal{X} \to \mathbb{R}$ has local* Lipschitz constant L *w.r.t. ℓ_q norm in region $B_{p,\epsilon}(x_0)$ if $\forall x_1, x_2 \in B_{p,\epsilon}(x_0)$, $\frac{|g(x_1)-g(x_2)|}{\|x_1-x_2\|_q} \leq L$.*

We can lower bound $f_\theta(x)_{y_0} - f_\theta(x)_{y'}$ for any $x \in B_{p,\epsilon}(x_0)$ given Lipschitz constant, and thus certify robustness [11].

1) ***General Lipschitz*** *[11], [39], [80]–[84]:* Some verification approaches aim at computing a tight Lipschitz bound for general neural networks, and we call them *general Lipschitz based verification approaches*. A commonly-used approach [11], [81], [83] is to compute a global Lipschitz constant w.r.t. ℓ_2 norm by multiplying the spectral norm of all weight matrices $L = \prod_{i=1}^{l-1} \|\boldsymbol{W}_i\|$ of the ReLU neural network, where the spectral norm can be computed by power iteration algorithm [124]. Efforts have been made on tightening this bound [82]. The global Lipschitz constant is usually too loose to provide nontrivial certified robustness in practice, but it can be efficiently regularized during training. When this constant is regularized, this verification can bring non-trivial certified robustness against ℓ_2 norm, which we will discuss in detail in Sec. V. Global Lipschitz constant computation is efficient and thus scalable to models on Tiny ImageNet dataset [80], [81].

Global Lipschitz bound can be improved by finer-grained analysis on convolutional layers [80], by computing local Lipschitz bound [39], [84], [125], or by combining with IBP [80]. Currently, general Lipschitz based verification can certify nontrivial robustness against only ℓ_2 adversary.

2) ***Smooth Layers*** *[85]–[89]:* Besides general Lipschitz based verification, another thread of research proposes specific layer structures which we call *smooth layers* and proves Lipschitz constant for these layer structures. For example, there are different designs of orthogonal convolutional layers [86], [88]. They usually use parameterization or transformation to explicitly construct trainable convolutional layers which are orthogonal and thus have 1 as the Lipschitz constant. When this small Lipschitz constant is proved, general Lipschitz based verification can provide robustness certification. However, these approaches are restricted to ℓ_2 adversary. Recently, Zhang et al [89] propose a novel activation function $x \mapsto \|x - w\|_\infty + b$ which is called ℓ_∞ neuron and is 1-Lipschitz w.r.t. ℓ_∞ norm. This design enables general Lipschitz based verification to certify robustness against ℓ_∞ adversary. Combined with effective training [126], this approach can certify state-of-the-art ℓ_∞ certified robustness.

3) ***Curvature*** *[90]:* If a DNN uses activation functions that have non-zero second-order derivatives, such as sigmoid, Singla and Feizi [90] propose an efficient algorithm to bound the DNN's curvature, i.e., second-order derivatives. Based on the curvature bound, we can compute a lower bound of Problem 1 and thus certify the model's robustness. Compared with others, curvature-based verification cannot be applied to the widely-used ReLU networks and can only certify against ℓ_2 adversary, so the application scenario is a bit limited. Though on small dataset like MNIST, this approach can verify high robustness for some robustly trained models.

Practical implications. When applying Lipschitz- or curvature-based approaches to general DNN models, the resulting robustness verification is usually trivial. However, if the model is regularized to have a small Lipschitz constant or uses smooth layers with effective training, these approaches can provide nontrivial bounds on both small and large datasets (MNIST, CIFAR-10, and Tiny ImageNet) against both ℓ_2 and ℓ_∞ adversaries.

Research implications. Among these Lipschitz and curvature based verification, we believe that the potential of smooth layers based verification has not been fully explored yet. For example, many orthogonal training methods can lead to smooth layers (e.g., [127]–[129]), and smooth variants may exist for recent architectures such as transformers [6]. Improving on these directions may lead to state-of-the-art certified robustness.

E. Incomplete Verification via Probabilistic Approaches

Besides deterministic verification, one recently emerging branch of studies proposes to add random noise to smooth the models, and thus derive the certified robustness for these *smoothed models* (See Def. 7). We call this line of work *probabilistic robustness verification* approaches or *randomized smoothing based* approaches since they provide probabilistic robustness guarantees and all existing probabilistic verification approaches are designed for smoothed models. Currently, only these verification approaches are scalable enough to certify nontrivial robustness on the large-scale ImageNet dataset.

Definition 7 (Smoothed Classifier)**.** *Given a smoothing distribution μ whose support is* $\mathrm{supp}(\mu)$ *and density at point*

δ *denoted by* $\mu(\delta)$. *For a given classifier* F, *the* smoothed classifier F_{smooth} *is defined as:*

$$F_{\text{smooth}}(x) = \arg\max_{i \in [C]} \int_{\delta \in \text{supp}(\mu)} \mathbb{I}\left[F(x+\delta) = i\right] \mu(\delta)\, \mathrm{d}\delta.$$

The integral in Def. 7 cannot be exactly solved. Thus, instead, Monte-Carlo estimation and hypothesis testing [130] are used to approximate the exact solution. As a result, the certification is probabilistic rather than deterministic (Def. 3).

1) ***Approaches with Zeroth-Order Information*** *[22], [28], [92]–[96], [98], [131]:* A majority of verification approaches only use zeroth-order information of the smoothed classifier, i.e., the probabilities $\Pr_{\delta\sim\mu}[F(x_0+\delta) = y]$ for $y \in [C]$ where the clean input is x_0, to compute the robustness certification. Among these approaches, Neyman-Pearson based approaches are proved to be the tightest [28].

Given clean input x_0, when adding noise δ, we suppose the model F predicts true class y_0 with probability $P_A := \Pr_{\delta\sim\mu}[F(x_0+\delta) = y_0]$ and runner-up class with $P_B := \max_{y' \in [C]: y' \neq y_0} \Pr_{\delta\sim\mu}[F(x_0+\delta) = y']$. High-confidence intervals for P_A and P_B can be obtained with Monte-Carlo sampling. The high-level **intuition** for Neyman-Pearson based approaches is: if the attacker's perturbed input x is close to x_0, the distribution of $x+\delta$ would highly overlap the distribution of $x_0+\delta$ where $\delta \sim \mu$ is the added smoothing noise. Therefore, the corresponding P'_A and P'_B for perturbed input x will not change too much from P_A and P_B for clean input x_0. That means, if there is a sufficient margin between P_A and P_B, then P'_A will still be larger than P'_B. Thus, the smoothed classifier will still predict y_0 for perturbed input x according to Def. 7.

Formally, based on Neyman-Pearson lemma [132], one can derive a tight lower bound for P'_A and upper bound for P'_B given P_A, P_B, and input shift (i.e., $x - x_0$). Then, we solve the distance lower bound that guarantees $P'_A > P'_B$ to get robustness certification.

Against ℓ_2 adversary, Cohen et al [22] consider Gaussian smoothing and derive a tight ℓ_2 robustness radius based on Neyman-Pearson lemma. **Against ℓ_1 adversary**, Lecuyer et al [92] and Teng et al [95] consider Laplacian smoothing and derive ℓ_1 robustness radius. **Against ℓ_∞ adversary**, Yang et al [28] empirically show and theoretically justify that it yields the highest ℓ_∞ certified radius by using Gaussian smoothing and transforming Neyman-Pearson-based ℓ_2 robustness radius to ℓ_∞ radius: $r \mapsto \frac{r}{\sqrt{d}}$ where d is the input dimension. However, for dataset where d is large, the certified ℓ_∞ radius is small. Indeed, certifying robustness against ℓ_∞ for high-dimensional input is proven to be intrinsically challenging for zeroth-order information approaches [28], [133]–[135].

Using zeroth-order information, the robustness verification can also be derived from: (1) differential privacy (DP) where Gaussian and Laplace mechanisms in DP can induce certified robustness for models smoothed with Gaussian and Laplace distributions [92]; (2) Lipschitz bound [98]; (3) statistics view [93], [94]; and (4) level-set method [28]. These approaches derive looser [92], [93] or equivalently tight [94]–[96], [98], [131] robustness certification as Neyman-Pearson based approaches.

2) ***Approaches with First-Order Information*** *[99], [100]:* Since zeroth-order information approaches have tightness barriers as discussed before, attempts have been made on querying more information from the smoothed model beyond only P_A and P_B. One example could be the gradient magnitude information $\|\frac{\partial \Pr_{\delta\sim\mu}[F(x_0+\delta)=y_0]}{\partial x_0}\|_p$ which can be estimated via sampling with high-confidence error interval [99], [100]. When using this first-order information together with P_A and P_B, we can derive a tighter robustness verification for smoothed models. Currently, the tightness improvements are pronounced against ℓ_1 adversary but not significant against ℓ_2 or ℓ_∞ adversaries.

3) ***Choice of Smoothing Distributions****:* To achieve satisfactory certified robustness, besides verification approaches, the choice of smoothing distribution is also important for these randomized smoothing based approaches.

In general, for ℓ_2 adversary, Gaussian smoothing distribution is most commonly used [22], [28]. Some work argues that Gaussian may not be the optimal smoothing distribution [96] but significantly better alternatives have not been found yet. For ℓ_1 adversary, Yang et al [28] show that uniform distribution is significantly better than others [92], [93], [95]. Recently, a specific non-additive discrete smoothing distribution is proposed [91], which enables Lipschitz-bound-based *deterministic* certification for smoothed models against ℓ_1 adversary. Since deterministic verification does not need to consider sampling error, it provides better robustness certification than [28].

The smoothing distributions can control trade-offs between certified robustness and accuracy, where distribution with larger variance can lead to a larger certified radius under the same P_A and P_B, but hurts the clean accuracy since input signal is more severely corrupted by noise [134], [135].

Practical implications. The confidence level of probabilistic verification is usually set to be high (e.g., 99.9%) to maintain the rigor of robustness certification. Probabilistic verification approaches are very strong against ℓ_1 and ℓ_2 adversaries, being the only type that can provide certification on large-scale datasets (e.g., ImageNet) and achieve the state-of-the-art on almost all other datasets. Moreover, they support arbitrary model architectures since only black-box access to final predictions is required. However, they fail to provide robustness certification that is tight enough against the ℓ_∞ adversary on ImageNet. Furthermore, the smoothing procedure adds additional overhead for inference and requires specific training to ensure good model performance which usually hurts the clean accuracy [135].

Research implications. For verification approaches for smoothed DNNs, promising research directions include: (1) Tighter verification: though zeroth-order verification fails to certify a large radius against ℓ_∞ adversary, other information can be leveraged to tighten it. For example, approaches using first-order information are visibly tighter than zeroth-order verification against ℓ_1 adversary [100]. However, there is still a visible gap between the current certified robustness radius and the empirically attackable radius [136], so improvement rooms exist. Thus, it would be a promising direction to derive tighter verification by using effective information in addition to zeroth-order information, especially for the challenging ℓ_2 and ℓ_∞ adversaries. A recent work [137] follows this direction and proposes double sampling randomized smoothing that achieves tighter verification against ℓ_2 adversary. In App. F, we extend their method and achieves tighter verification against ℓ_∞ adversary. Researchers can think about what types of information

from the smoothed models are useful and how to efficiently obtain them—novel sampling methods are needed. (2) Better smoothing distributions: suitable smoothing distributions have significantly improved ℓ_1 certified robustness [28], [91]. The same degree of improvements may exist against other ℓ_p adversaries. (3) Better training approaches for smoothed classifiers. Discussion is in Sec. V.

V. Robust Training Approaches

Normally-trained DNNs are usually non-robust where effective attacks can find adversarial examples with almost 100% probability [11], [18], [19]. To achieve high certified robustness, DNNs need to be trained with robust training approaches which aim to improve robustness guarantees, as illustrated in Fig. 2.

Current robustness verification approaches usually favor certain properties of DNNs to achieve high certified robustness. For instance: (1) Branch-and-bound verification (Sec. IV-A3) uses incomplete verification such as linear relaxation to reduce explored branches and boost the certification efficiency so it favors DNNs whose linear relaxations are tight and for other DNNs the certification process is significantly slower [58], [73]. (2) Linear relaxation verification (Sec. IV-B) can certify only models whose specific linear relaxations are tight. (3) Lipschitz or curvature verification (Sec. IV-D) can certify only models where a small Lipschitz or curvature constant can be computed. (4) Verification for smoothed DNNs (Sec. IV-E) certifies larger radius for models with higher correct-prediction probability under noise. These favored properties are not directly promoted by standard training or empirical defenses. Therefore, to improve *certified* robustness, robust training approaches are proposed to promote these properties during training.

We divide existing robust training approaches into three categories: regularization-based, relaxation-based, and augmentation-based. We defer the illustration to App. D.

Discussion. Robust training approaches can improve model robustness and at the same time remarkably enhance desired properties of models for corresponding verification approaches. Thus, models trained with a robust training approach usually achieve much better certified robustness based on corresponding verification, as reflected by evaluation in Sec. VI-A.

Models trained by one robust training approach are often verified to have poor robustness by a mismatched verification approach (as shown in Sec. VI-A). This is because the models do not inherit the desired property of the verification. For example, models trained for randomized smoothing based approach can predict well for noisy input but may have many unstable neurons and loose linear relaxations, making them difficult to be verified by complete or linear relaxation verification. Models trained for linear relaxation are not specialized for predicting noisy input and are challenging for randomized smoothing based verification [22].

As a result, an important research goal is to develop verification that does not heavily rely on specific model properties so it can verify most robustly trained or empirically defended models. Recently, some complete verification approaches [58], [73] are shown tractable for verifying empirically defensed (e.g., PGD adversarially trained [18]) models though they are still limited to small models. Thus, proposing more practically efficient complete verification approaches may be a viable path towards this goal. Another research goal is to improve certified robustness for a given task. For this goal, advances on both verification and robust training are valuable, such as tighter [74] or more training-friendly relaxation [120], or more effective training methods [121], [126], [138], which we will discuss further in Sec. VIII.

VI. Benchmark, Leaderboard, and Implications

In this section, we introduce an open-source toolbox to systematically benchmark $20+$ certifiably robust approaches. Based on benchmark results and the leaderboard on representative datasets, we outline practical implications for deploying certifiably robust approaches for DNNs.

A. Benchmark Evaluation

We present the following evaluation: (1) for representative **deterministic verification** approaches, we compare their certified robustness over a diverse set of trained models of different scales; (2) for representative **probabilistic verification** approaches and their corresponding robust training approaches, we compare the best certified robustness they jointly achieve. We do such separation because deterministic and probabilistic certificates have different semantics and their supported system models are different. The evaluation is made possible by our open-source unified toolbox—a first toolkit integrating a wide range of verification approaches.

We present the findings here and introduce the experiment protocol and representative results in App. E. Full results are on our benchmark website: *https://sokcertifiedrobustness.github.io*. In the evaluation, we use *certified accuracy* to measure certified robustness, which is the fraction of test set samples verified to be robust against the corresponding (ℓ_p, ϵ)-adversary.

1) Findings from Comparing Deterministic Verification Approaches: (1) On relatively small models, complete verification approaches can effectively verify robustness, thus they are the best choice. (2) On larger models, usually linear relaxation based verification approaches perform the best since the complete verification approaches are too slow and other approaches are too loose, yielding almost 0% certified accuracy. However, linear relaxation based verification still cannot handle large DNNs and they are still too loose compared with the upper bound provided by PGD attack. (3) On robustly trained models, if the robust training approach is CROWN-IBP which is tailored for IBP and CROWN (two linear relaxation verification approaches), IBP and CROWN can certify high certified accuracy while others fail to certify. Indeed, robust training approaches can usually boost the certified accuracy but the models must be verified with corresponding verification approaches as discussed in Sec. V. (4) SDP approaches usually take too long and thus are less practical.

2) Findings from Comparing Probabilistic Verification Approaches: (1) For both ℓ_1 and ℓ_2 adversaries, Neyman-Pearson based verification achieves the highest certified robustness. (2) Robust training approaches effectively enhance the models' certified robustness. Among these existing robust training approaches, adversarial training usually achieves the best

TABLE II

LEADERBOARD: TOP-5 CERTIFIED ACCURACY UNDER EACH SETTING ACHIEVED BY CORRESPONDING ROBUST TRAINING (SHOWN BY REFERENCE BRACKET) AND VERIFICATION (SHOWN BY NAME). ϵ DENOTES THE ATTACK RADIUS. * INDICATES CERTIFIED ACCURACY BY PROBABILISTIC VERIFICATION, OTHERWISE BY DETERMINISTIC VERIFICATION. "NYM.-PRSN." MEANS NEYMAN-PEARSON-BASED OR EQUIVALENTLY TIGHT VERIFICATION.

	ℓ_∞ Adversary				ℓ_2 Adversary				ℓ_1 Adversary			
	$\epsilon = 0.1$		$\epsilon = 0.3$		$\epsilon = 0.5$		$\epsilon = 1.58$		Existing approaches for ℓ_1 are all randomized-smoothing-based, which are generally evaluated on CIFAR-10 and ImageNet datasets.			
MNIST	**97.95%**	[121], Interval	**93.20%**	[126], Smooth Layers	**98.2%***	[138], Nym.-Prsn.	**70.7%***	[138], Nym.-Prsn.($\epsilon = 1.75$)				
	97.91%	[139], Duality	93.10%	[121], Interval	98.0%*	[140], Nym.-Prsn.	70.5%*	[140], Nym.-Prsn.($\epsilon = 1.75$)				
	97.77%	[61], Interval	93.09%	[89], Smooth Layers	78.45%	[90], Curvature	69.79%	[90], Curvature				
	97.76%	[112], Polyhedra	92.98%	[112], Polyhedra			69.0%*	[93], Divergence Based				
	97.26%	[71], Duality	91.95%	[61], Interval			62.8%	[81], General Lipschitz				
	$\epsilon = 2/255$		$\epsilon = 8/255$		$\epsilon = 36/255$		$\epsilon = 0.25$		$\epsilon = 1.0$		$\epsilon = 2.0$	
CIFAR-10	**68.2%***	[98], Nym.-Prsn.	**40.06%**	[126], Smooth Layers	**65.6%***	[93], Divergence Based	**81%***	[98], Nym.-Prsn.	**63.07%**	[91], Lipschitz	**51.33%**	[91], Lipschitz
	63.8%*	[141], Nym.-Prsn.	35.42%	[89], Smooth Layers	59.16%	[88], Smooth Layers	72%*	[141], Nym.-Prsn.	63%*	[28], Nym.-Prsn.	48%*	[28], Nym.-Prsn.
	60.5%	[142], Polyhedra	34.97%	[121], Interval	58.4%	[81], General Lipschitz	71%*	[143], Nym.-Prsn.	39%*	[95], Nym.-Prsn.	17%*	[96], Nym.-Prsn.
	54.12%	[126], Smooth Layers	33.38%	[65], Polyhedra	51.96%	[71], Duality	68.8%*	[140], Nym.-Prsn.	34%*	[96], Nym.-Prsn.	16%*	[95], Nym.-Prsn.
	53.97%	[112], Polyhedra	33.06%	[112], Polyhedra	51.30%	[80], General Lipschitz	67.9%*	[138], Nym.-Prsn.	18%*	[92], Differential Privacy	5%*	[92], Differential Privacy
	$\epsilon = 1/255$		No work achieves $> 0\%$ certified accuracy under large ϵ yet.		$\epsilon = 1.0$		$\epsilon = 2.0$		$\epsilon = 1.0$		$\epsilon = 2.0$	
ImageNet	**38.2%***	[98], Nym.-Prsn.			**45%***	[98], Nym.-Prsn.	**30.4%***	[144], Nym.-Prsn.	**55%***	[28], Nym.-Prsn.	**48%***	[28], Nym.-Prsn.
	28.6%*	[22], Nym.-Prsn.			44.4%*	[144], Nym.-Prsn.	28%*	[98], Nym.-Prsn.	49%	[91], Lipschitz	45%	[91], Lipschitz
					44%*	[140], Nym.-Prsn.	27%*	[143], Nym.-Prsn.	42%*	[96], Nym.-Prsn.	30%*	[96], Nym.-Prsn.
					43%*	[138], Nym.-Prsn.	26%*	[138], Nym.-Prsn.	40%*	[95], Nym.-Prsn.	26%*	[95], Nym.-Prsn.
					43%*	[143], Nym.-Prsn.	24%*	[140], Nym.-Prsn.	25%*	[92], Differential Privacy	16%*	[92], Differential Privacy

performance. (3) The choice of smoothing distribution can greatly affect the certified accuracy. Under ℓ_1 adversary, the superior result is achieved by uniform smoothing distribution. (4) For probabilistic verification approaches, certifying robustness under ℓ_∞ adversary is challenging, and would become more challenging when the data dimension increases, which coincides with theory [28], [133], [134].

B. Leaderboard on Certified Robustness

What is the state-of-the-art certified accuracy achieved on representative datasets? Table II shows a leaderboard of certified accuracy under different settings from peer-reviewed publications till Aug. 1, 2022. The high certified accuracy is jointly achieved by robust training (shown by reference bracket) and verification (shown by name).

As we can see, much progress has been made in the certified robustness field in recent years. On MNIST, the certified accuracy against ℓ_∞ adversary with 0.3 radius has reached over 93%. This is remarkable since the limit is 0.5 radius where any input image can be perturbed to indistinguishable half-gray. This is achieved by robust training for interval verification [121] and Lipschitz verification based on smooth layers [89]. On more challenging CIFAR-10 and ImageNet datasets, however, certified accuracy is still low. On CIFAR-10, against ℓ_∞ adversary with $2/255$ radius, the certified accuracy is only around 68% [98]; with radius $8/255$, it is only 40.06% [126]. These are far from state-of-the-art $90\%+$ clean accuracy or $65\%+$ accuracy under strong attacks [145]. On ImageNet, only approaches for smoothed DNNs can provide non-zero robust accuracy: 30.4% under ℓ_2 radius 2.0 [144]; and around 38% under ℓ_∞ radius $1/255$ [98].

C. Practical Implications

In practice, what are the most suitable certifiably robust approaches for users to deploy? Based on the benchmark results and the leaderboard, we present practical implications in Fig. 4, where we envision two scenarios: 1) users want to improve certified robustness for their tasks at hand; 2) users want to evaluate or certify the robustness of given models.

When users want to improve certified robustness for their tasks, they need to achieve this by choosing a robust training approach and certifying the robustness with the corresponding verification approach as discussed in Sec. V. The upper part of Fig. 4 shows the recommended combinations of verification and

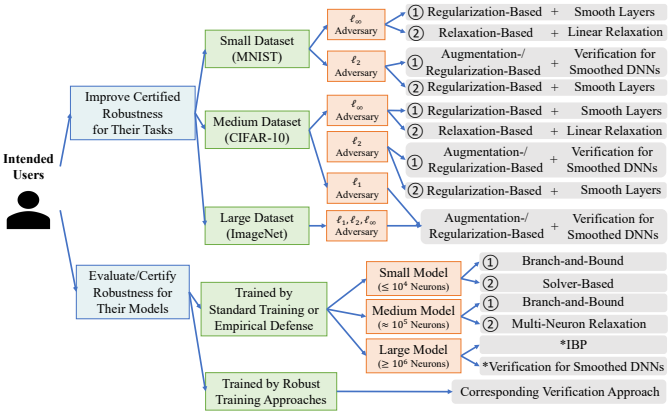


Fig. 4. Practical implications for users to select certifiably robust approaches. Gray boxes depict suitable "(verification) + (robust training)" approach combinations or verification approaches for given scenarios. If exist, "①" and "②" label the most and runner-up suitable ones. Details in Sec. VI-C.

robust training approaches in light gray boxes. Depending on the dataset size and the type of ℓ_p adversary to defend against, we recommend the corresponding approach combinations which achieve high certified accuracy in practice based on our leaderboard (Table II). When multiple choices are available, we show the top-2 choices and label them with "①" and "②" respectively. In summary, linear relaxation and smooth layer based verification and corresponding robust training perform well on small and medium datasets against ℓ_∞ adversary, while for other datasets and other ℓ_p adversaries, approaches for smoothed DNNs are better. In particular, on large datasets like ImageNet, only approaches for smoothed DNNs can provide robustness certification at the current stage.

When users want to evaluate or certify the robustness of certain models, they need to choose a suitable verification approach. Inspired by our benchmark, we present the implications in the lower part of Fig. 4. For small and medium models trained by standard training or empirical defenses, the branch-and-bound based complete verification [53], [58] and multi-neuron relaxation verification [73] can certify the robustness efficiently. Specifically, for small models, the solver-based (concretely, MILP-based) verification approaches can certify good robustness. But for large models, none of these methods cannot finish in a feasible time (one day per input). Therefore, we must use more efficient but loose verification such as IBP (Sec. IV-B2) and verification approaches for

smoothed DNNs (Sec. IV-E), which usually yield trivial certified robustness radius and it is an active research area to make tighter verification approaches scalable for these large models. In addition, we find that the ranking of empirical defenses for small/medium models based on certified robustness is consistent with that evaluated by strong empirical attacks such as PGD [58], [75], [146]. If models are trained by robust training approaches, as discussed in Sec. V, using the corresponding verification approaches targeted by the training approach would be the best choice.

VII. Extensions and Applications

The methodologies derived from certifiably robust DNNs have recently been applied to much broader areas.

Extensions to other threat models. Though certifiably robust approaches mainly focus on the (ℓ_p, ϵ)-adversary, extending the related techniques to other threat models has drawn much attention. (1) **Local evasion attacks**: in local evasion attacks, the adversary slightly perturbs the in-distribution data to mislead the model. Our (ℓ_p, ϵ)-adversary adds pixelwise perturbations bounded by ℓ_p norm within ϵ. Now we elaborate on some other effective local evasion attacks. (a) *Semantic adversary* picks an arbitrary but bounded transformation parameter, such as rotation angle, and applies the transformation to perturb the input [147]–[149]. Neyman-Pearson approaches and linear relaxation approaches can be extended to provide certification [150]–[153]. The core methodology is to split the low-dimensional parameter space into tiny intervals and then bound the input changes in each interval. (b) *Generative model based adversary* uses generative models such as GAN [154] to generate input perturbation. Similar to certification against the semantic adversary, Neyman-Pearson approaches and linear relaxation approaches can be extended to provide certification against this adversary [155], [156]. (c) ℓ_0 *adversary* picks a bounded number of pixels to arbitrarily change and *patch adversary* picks a region of pixels with a bounded area to arbitrarily change. To defend against ℓ_0 adversary, Neyman-Pearson approaches can be deployed [131], [157], [158]. To defend against patch adversary, the core idea of Neyman-Pearson approaches, prediction aggregation on several noisy inputs which are patched inputs here, is leveraged to develop customized certification and corresponding training approaches [65], [159]–[163]. (2) **Distributional evasion attacks**: in distributional evasion attacks, the attacker shifts the whole test data distribution within some bounded distance to maximize the expected loss. This threat model can be used to characterize the out-of-distribution generalization ability of ML models [164]. The certification under this threat model is an upper bound of the expected loss, which can be derived from duality under Lipschitz and curvature assumptions [165] or from extensions of Neyman-Pearson approaches [166], [167]. (3) **Global evasion attacks**: global evasion attacks can perturb any valid input example to mislead the model, whereas local evasion attacks can only perturb in-distribution data. Thus, the robustness against global evasion attacks means that the robustness property holds for the whole input domain. An example of a robustness property is that for any high-confident prediction, small perturbations cannot change the predicted label [81]. In the security domain, Chen et al [168] recently proposed several domain-specific robustness properties such as requiring all low-cost features to be robust. To verify these properties, they propose a specific solver-based verification (Sec. IV-A1) to verify *logic ensemble models*, and then use the found adversarial example as an augmentation for robust training. The verification and robust training *for DNNs* against global evasion attacks can be a promising direction. (4) **Training-time attacks**: training-time attacks can manipulate some training data to reduce the trained model's performance or inject some backdoors. Against this threat model, verification approaches extended from Neyman-Pearson can provide robustness certification [169]–[172].

Extensions to diverse types of system models. There are efforts on generalizing existing DNN verification approaches to deal with more types of system models. For example: (1) Some approaches that are designed for feed-forward ReLU networks, such as linear relaxation based approaches, have been extended to support general DNNs [38], [64], [71], recurrent networks [173]–[175], transformers [102], [176], generative models [177], and model ensembles [178]. The main methodology is to derive the corresponding linear bounds for activation functions or attention mechanisms in these system models. Some complete verification approaches, e.g., branch-and-bound based ones [58], also support general DNNs. However, these complete verification approaches become incomplete when applied on general DNNs. (2) Verification approaches for Lipschitz-bounded networks and non-ReLU networks have not been generalized to other system models yet. (3) Verification approaches for smoothed DNNs typically need access to only the final prediction label, so they are applicable to any classification models. However, the model must follow the corresponding smoothing-based inference protocol. (4) There are also verification approaches for decision trees [179]–[181], decision stumps [181], and logic ensembles [168]. However, there is no verification and robust training approach that supports all these system models yet. This is because verification and robust training approaches need to exploit properties (piecewise linearity, Lipschitz bound, smoothness, etc) of specific system models to achieve certified robustness.

Certified robustness for concrete applications. Beyond the classification task, the discussed methodologies, such as linear relaxation and Neyman-Pearson approaches, have been extended to certify DNNs in many concrete applications. In natural language processing, extensions include certification for recurrent neural networks against embedding perturbations [173]–[175], word substitutions [182], and word transformations [183]–[185]. Extensions have also been studied for object detection [186], segmentation [187], and point cloud models [187]–[189] in computer vision, and speech recognition [150], [190]. Verification and robust training approaches have also been proposed for reinforcement learning [146], [172], [191]–[193].

VIII. Insights, Challenges, and Future Directions

In this section, we summarize characteristics, strengths, limitations, and fundamental connections among certifiably robust approaches, then discuss barriers, main challenges, and future directions for DNN certification.

A unified view: characteristics, strengths, limitations, and connections of certifiably robust approaches. To reveal the fundamental connections, we adopt a unified view of robustness verification: all existing verification approaches provide an abstraction of given DNN models to verify the robustness. For example, the branch-and-bound verification views the model as the union of several sub-domains where the model output in each domain can be bounded, e.g., by linear inequalities. The branching process is essentially refining the abstraction by splitting sub-domains whose current abstractions are not precise enough. The linear relaxation based verification uses some linear constraints to abstract the possible behavior of the model in the whole perturbation region. The probabilistic verification uses the queried information, such as zeroth-order information, to abstract the model behavior. This view is closely related to the concept of abstract interpretation in traditional program analysis [194]. Therefore, the scalability and tightness trade-off of verification mentioned in Sec. II-C is essentially the inherent trade-off between preciseness and efficiency of abstraction: more precise abstraction enables tighter robustness certification, whereas has higher time and space complexity. Thus, for a model that is not specifically trained, the most suitable verification approach is the most precise one that can be computed for this model size. Concrete approach selection guidelines are in Sec. VI-C. We note that, under this unified view, the favored properties of each verification (listed in Sec. V) are tight conditions of the corresponding abstraction domain. Thus, robust training approaches that promote these properties can boost verification tightness for the model to improve certified robustness. More concrete strengths and limitations of each verification are discussed in "practical implications" and "research implications" boxes in Sec. IV.

Challenges and barriers. Although there has been remarkable progress towards certifiably robust DNNs, scalability and tightness challenges persist. For example: (1) Complete verification is NP-complete [23], [39]. (2) Multi-neuron based linear relaxation needs exponential number of constraints [74]. (3) Probabilistic certification based on zeroth-order information cannot certify high robustness against ℓ_∞ adversary for real-world high-dimensional inputs [28], [133]–[135]. These theoretical barriers are intrinsic challenges for further improvements in these verification approaches. There are also practical issues to solve, such as guaranteeing verification soundness under floating-point arithmetic [118], [119], [195] and safeguarding robust training against training-time attacks [196].

Future directions. Despite the challenges and barriers, there are also several potential future directions: (1) **Scalable and tight verification:** There are still hopes for more scalable and tighter verification for DNNs *in practice* despite theoretical barriers. For example, good heuristics have boosted complete verification to handle DNNs with over 10^5 neurons [58]. It is promising to explore other better heuristics. For instance, a recent work [53] improves the complete verification by proposing better bounding heuristics based on multi-neuron relaxation. For SDP verification, better formulation and solvers can lead to better verification [75], [123]. For smoothed DNNs, although only using zeroth-order information cannot certify high robustness against ℓ_∞ adversary, this barrier may be circumvented by leveraging more information as in [99], [100] which improve ℓ_1 and ℓ_∞ certification tightness; or leveraging non-additive smoothing distribution as in [91] which improves ℓ_1 certification tightness. More details are discussed in Sec. IV-E. (2) **Effective robust training with theoretical understanding:** Unlike verification where theoretical barriers exist, robust training can empirically boost the certified robustness without known theoretical limitations. Indeed, even the empirically loose interval relaxations (see Sec. IV-B2) are universal approximators [197], [198] and achieve training convergence (under some assumptions) [199], which implies that with effective and generalizable robust training the certified accuracy could be on par with benign accuracy. However, theoretical understanding of robust training, such as why robust training generalizes, is still lacking [120]. Recent work shows that when an efficient complete verification approach exists, generalizable robust training is achievable [200]. Extending this result to broader scenarios, e.g., the generalization of robust training with *incomplete* verification, would significantly advance our understanding of ML robustness. (3) **Design certifiably robust DNN architectures**: Based on the model properties required for different verification approaches, it is promising to design novel DNN architectures to further improve the certified robustness. In addition, it is also possible to design sparse DNNs following the model compression literature [201] to achieve efficient and certifiably robust models. (4) **Certification for other ML utilities:** Techniques of certified robustness can be extended to certify other ML utilities such as fairness [202]–[204] and generalization [166], [167]. It is an emerging trend to provide certification for generic ML utilities, such as model bias, toxicity, and model unlearning, or train models to achieve such certifications [205]. (5) **Certification for different ML models:** Current robustness certification mainly focuses on classification models, and it would be critical to extend such certification to other ML models, such as reinforcement learning, federated learning, and large language models, which have demonstrated their real-world usage in safety-critical domains. (6) **Integrate domain knowledge and logic reasoning ability into ML to improve certified robustness:** It has been shown that joint inference with knowledge rules can improve model benign accuracy [206]–[208], and therefore it would be promising to integrate domain knowledge, causal analysis, and security rules into ML pipeline to further improve and tighten its end-to-end certified robustness. (7) **Bring certified robustness to real-world applications:** Besides achieving higher certified robustness on standard benchmarks, we believe that adaptation of verification and robust training approaches for real-world

applications is also critical. For example, security threats are found on DNNs in autonomous vehicles [14], [209] which may lead to severe consequences [148], [210]. Designing a certifiably robust autonomous driving system would be an important, timely, and promising direction.

IX. Conclusions

We presented an SoK for certifiably robust approaches for DNNs, including both robustness verification approaches and robust training approaches. We show characteristics, strengths, limitations, and fundamental connections among these approaches. Our discussion summarizes the current research status both theoretically and empirically, reveals limitations, and highlights future directions.

Acknowledgment

We would like to thank Xiangyu Qi for conducting the benchmark evaluation on some probabilistic verification approaches for smoothed DNNs. We thank Dr. Ce Zhang, Dr. Sasa Misailovic, and Dr. Gagandeep Singh for their thoughtful feedback. We also thank the support of NSF grant No.1910100, NSF CNS 2046726, C3 AI, the Alfred P. Sloan Foundation, and the AWS Research Awards.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[7] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[8] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma, "Adversarial classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 99–108.

[9] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 641–647.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[12] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018, pp. 274–283.

[13] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *International Conference on Learning Representations*, 2018.

[14] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, "Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 176–194.

[15] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[16] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[17] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations*, 2018.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[19] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[20] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *International Conference on Learning Representations*, 2018.

[21] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *arXiv preprint arXiv:2002.08347*, 2020.

[22] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *International Conference on Machine Learning*, 2019.

[23] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2017, pp. 97–117.

[24] G. Singh, R. Ganvir, M. Püschel, and M. Vechev, "Beyond the single neuron convex barrier for neural network certification," in *Advances in Neural Information Processing Systems*, 2019, pp. 15 072–15 083.

[25] G. Singh, T. Gehr, M. Püschel, and M. Vechev, "Boosting robustness certification of neural networks," in *International Conference on Learning Representations*, 2019.

[26] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Efficient formal safety analysis of neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 6367–6377.

[27] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*, 2018, pp. 5286–5295.

[28] G. Yang, T. Duan, E. Hu, H. Salman, I. Razenshteyn, and J. Li, "Randomized smoothing of all shapes and sizes," in *International Conference on Machine Learning*, 2020.

[29] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[30] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.

[31] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, pp. 43–58.

[32] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "Sok: Security and privacy in machine learning," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2018, pp. 399–414.

[33] A. Albarghouthi, "Introduction to neural network verification," *arXiv preprint arXiv:2109.10317*, 2021.

[34] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020.

[35] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer *et al.*, "Algorithms for verifying deep neural networks," *Foundations and Trends® in Optimization*, vol. 4, no. 3-4, pp. 244–404, 2021.

[36] D. J. Miller, Z. Xiang, and G. Kesidis, "Adversarial learning targeting deep neural network classification: A comprehensive review of defenses

against attacks," *Proceedings of the IEEE*, vol. 108, no. 3, pp. 402–433, 2020.

[37] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "AI^2: Safety and robustness certification of neural networks with abstract interpretation," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 3–18.

[38] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in neural information processing systems*, 2018, pp. 4939–4948.

[39] L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, "Towards fast computation of certified robustness for relu networks," in *International Conference on Machine Learning*, 2018, pp. 5276–5285.

[40] H. Salman, G. Yang, H. Zhang, C.-J. Hsieh, and P. Zhang, "A convex relaxation barrier to tight robustness verification of neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 9832–9842.

[41] L. Pulina and A. Tacchella, "An abstraction-refinement approach to verification of artificial neural networks," in *International Conference on Computer Aided Verification*. Springer, 2010, pp. 243–257.

[42] ——, "Challenging smt solvers to verify neural networks," *Ai Communications*, vol. 25, no. 2, pp. 117–135, 2012.

[43] C.-H. Cheng, G. Nührenberg, and H. Ruess, "Maximum resilience of artificial neural networks," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2017, pp. 251–268.

[44] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, "Output range analysis for deep feedforward neural networks," in *NASA Formal Methods - 10th International Symposium*, vol. 10811, 2018, pp. 121–138.

[45] A. Lomuscio and L. Maganti, "An approach to reachability analysis for feed-forward relu neural networks," *arXiv preprint arXiv:1706.07351*, 2017.

[46] V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in *International Conference on Learning Representations*, 2019.

[47] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljić *et al.*, "The marabou framework for verification and analysis of deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2019, pp. 443–452.

[48] S. Bak, H.-D. Tran, K. Hobbs, and T. T. Johnson, "Improved geometric path enumeration for verifying relu neural networks," in *International Conference on Computer Aided Verification*. Springer, 2020, pp. 66–96.

[49] R. Bunel, P. Mudigonda, I. Turkaslan, P. Torr, J. Lu, and P. Kohli, "Branch and bound for piecewise linear neural network verification," *Journal of Machine Learning Research*, vol. 21, no. 2020, 2020.

[50] R. R. Bunel, I. Turkaslan, P. Torr, P. Kohli, and P. K. Mudigonda, "A unified view of piecewise linear neural network verification," in *Advances in Neural Information Processing Systems*, 2018, pp. 4790–4799.

[51] A. De Palma, R. Bunel, A. Desmaison, K. Dvijotham, P. Kohli, P. H. Torr, and M. P. Kumar, "Improved branch and bound for neural network verification via lagrangian decomposition," *arXiv preprint arXiv:2104.06718*, 2021.

[52] R. Ehlers, "Formal verification of piece-wise linear feed-forward neural networks," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2017, pp. 269–286.

[53] C. Ferrari, M. N. Mueller, N. Jovanović, and M. Vechev, "Complete verification via multi-neuron relaxation guided branch-and-bound," in *International Conference on Learning Representations*, 2021.

[54] A. Fromherz, K. Leino, M. Fredrikson, B. Parno, and C. Pasareanu, "Fast geometric projections for local robustness certification," in *International Conference on Learning Representations*, 2021.

[55] M. Jordan, J. Lewis, and A. G. Dimakis, "Provable certificates for adversarial examples: Fitting a ball in the union of polytopes," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 059–14 069.

[56] J. Lu and M. P. Kumar, "Neural network branching for neural network verification," in *International Conference on Learning Representations*, 2020.

[57] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *27th USENIX Security Symposium (USENIX) Security 18)*, 2018, pp. 1599–1614.

[58] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and J. Z. Kolter, "Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[59] K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin, and C.-J. Hsieh, "Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers," in *International Conference on Learning Representations*, 2021.

[60] H. Zhang, S. Wang, K. Xu, L. Li, B. Li, S. Jana, C.-J. Hsieh, and J. Z. Kolter, "General cutting planes for bound-propagation-based neural network verification," *arXiv preprint arXiv:2208.05740*, 2022.

[61] S. Gowal, K. D. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, "Scalable verified training for provably robust image classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4842–4851.

[62] Z. Lyu, C.-Y. Ko, Z. Kong, N. Wong, D. Lin, and L. Daniel, "Fastened crown: Tightened neural network robustness certificates," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5037–5044.

[63] Z. Lyu, M. Guo, T. Wu, G. Xu, K. Zhang, and D. Lin, "Towards evaluating and training verifiably robust neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4308–4317.

[64] G. Singh, T. Gehr, M. Püschel, and M. Vechev, "An abstract domain for certifying neural networks," *Proceedings of the ACM on Programming Languages*, vol. 3, no. POPL, p. 41, 2019.

[65] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, "Automatic perturbation analysis for scalable certified robustness and beyond," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[66] G. Anderson, S. Pailoor, I. Dillig, and S. Chaudhuri, "Optimization and abstraction: a synergistic approach for analyzing neural network robustness," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2019, pp. 731–744.

[67] M. Mirman, T. Gehr, and M. Vechev, "Differentiable abstract interpretation for provably robust neural networks," in *International Conference on Machine Learning*, 2018, pp. 3575–3583.

[68] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev, "Fast and effective robustness certification," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 802–10 813.

[69] K. Dvijotham, S. Gowal, R. Stanforth, R. Arandjelovic, B. O'Donoghue, J. Uesato, and P. Kohli, "Training verified learners with learned verifiers," *arXiv preprint arXiv:1805.10265*, 2018.

[70] K. Dvijotham, R. Stanforth, S. Gowal, T. A. Mann, and P. Kohli, "A dual approach to scalable verification of deep networks." in *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, vol. 1, 2018, pp. 550–559.

[71] E. Wong, F. Schmidt, J. H. Metzen, and J. Z. Kolter, "Scaling provable adversarial defenses," in *Advances in Neural Information Processing Systems*, 2018, pp. 8400–8409.

[72] M. N. Müller, G. Makarchuk, G. Singh, M. Püschel, and M. Vechev, "PRIMA: Precise and general neural network certification via multi-neuron convex relaxations," *Proceedings of the ACM on Programming Languages*, vol. 6, no. POPL, pp. 1–33, 2022.

[73] A. D. Palma, H. Behl, R. R. Bunel, P. Torr, and M. P. Kumar, "Scaling the convex barrier with active sets," in *International Conference on Learning Representations*, 2021.

[74] C. Tjandraatmadja, R. Anderson, J. Huchette, W. Ma, K. K. PATEL, and J. P. Vielma, "The convex relaxation barrier, revisited: Tightened single-neuron relaxations for neural network verification," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 675–21 686, 2020.

[75] S. Dathathri, K. Dvijotham, A. Kurakin, A. Raghunathan, J. Uesato, R. R. Bunel, S. Shankar, J. Steinhardt, I. Goodfellow, P. S. Liang, and P. Kohli, "Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, 2020, pp. 5318–5331.

[76] K. D. Dvijotham, R. Stanforth, S. Gowal, C. Qin, S. De, and P. Kohli, "Efficient neural network verification with exactness characterization," in *Proc. Uncertainty in Artificial Intelligence, UAI*, 2019, p. 164.

[77] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and

semidefinite programming," *IEEE Transactions on Automatic Control*, 2020.

[78] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *International Conference on Learning Representations*, 2018.

[79] A. Raghunathan, J. Steinhardt, and P. S. Liang, "Semidefinite relaxations for certifying robustness to adversarial examples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 877–10 887.

[80] S. Lee, J. Lee, and S. Park, "Lipschitz-certifiable training with a tight outer bound," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[81] K. Leino, Z. Wang, and M. Fredrikson, "Globally-robust neural networks," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 6212–6222.

[82] S. Singla and S. Feizi, "Fantastic four: Differentiable and efficient bounds on singular values of convolution layers," in *International Conference on Learning Representations*, 2021.

[83] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Lipschitz-margin training: scalable certification of perturbation invariance for deep neural networks," in *Advances in Neural Information Processing Systems*, 2018.

[84] H. Zhang, P. Zhang, and C.-J. Hsieh, "Recurjac: An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5757–5764.

[85] M. Hein and M. Andriushchenko, "Formal guarantees on the robustness of a classifier against adversarial manipulation," in *Advances in Neural Information Processing Systems*, 2017, pp. 2266–2276.

[86] Q. Li, S. Haque, C. Anil, J. Lucas, R. B. Grosse, and J.-H. Jacobsen, "Preventing gradient attenuation in lipschitz constrained convolutional networks," *Advances in neural information processing systems*, vol. 32, pp. 15 390–15 402, 2019.

[87] S. Singla, S. Singla, and S. Feizi, "Improved deterministic l2 robustness on CIFAR-10 and CIFAR-100," in *International Conference on Learning Representations*, 2022.

[88] A. Trockman and J. Z. Kolter, "Orthogonalizing convolutional layers with the cayley transform," in *International Conference on Learning Representations*, 2021.

[89] B. Zhang, T. Cai, Z. Lu, D. He, and L. Wang, "Towards certifying l-infinity robustness using neural networks with l-inf-dist neurons," in *International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 12 368–12 379.

[90] S. Singla and S. Feizi, "Second-order provable defenses against adversarial attacks," in *International Conference on Machine Learning*, 2020.

[91] A. J. Levine and S. Feizi, "Improved, deterministic smoothing for l_1 certified robustness," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 6254–6264.

[92] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 656–672.

[93] B. Li, C. Chen, W. Wang, and L. Carin, "Certified adversarial robustness with additive noise," in *Advances in Neural Information Processing Systems*, 2019, pp. 9459–9469.

[94] K. D. Dvijotham, J. Hayes, B. Balle, Z. Kolter, C. Qin, A. Gyorgy, K. Xiao, S. Gowal, and P. Kohli, "A framework for robustness certification of smoothed classifiers using f-divergences," in *International Conference on Learning Representations*, 2020.

[95] J. Teng, G.-H. Lee, and Y. Yuan, "ℓ_1 adversarial robustness certificates: a randomized smoothing approach," 2020. [Online]. Available: https://openreview.net/forum?id=H1lQIgrFDS

[96] D. Zhang, M. Ye, C. Gong, Z. Zhu, and Q. Liu, "Black-box certification with randomized smoothing: A functional optimization based framework," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 2316–2326.

[97] P. Awasthi, H. Jain, A. S. Rawat, and A. Vijayaraghavan, "Adversarial robustness via robust low rank representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 391–11 403, 2020.

[98] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang, "Provably robust deep learning via adversarially trained smoothed classifiers," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 289–11 300.

[99] A. Levine, A. Kumar, T. Goldstein, and S. Feizi, "Tight second-order certificates for randomized smoothing," *arXiv preprint arXiv:2010.10549*, 2020.

[100] J. Mohapatra, C.-Y. Ko, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel, "Higher-order certification for randomized smoothing," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[101] A. Boopathy, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel, "Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3240–3247.

[102] Z. Shi, H. Zhang, K.-W. Chang, M. Huang, and C.-J. Hsieh, "Robustness verification for transformers," in *International Conference on Learning Representations*, 2020.

[103] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[104] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International conference on machine learning*. PMLR, 2020, pp. 2206–2216.

[105] Z. Yang, L. Li, X. Xu, S. Zuo, Q. Chen, P. Zhou, B. I. P. Rubinstein, C. Zhang, and B. Li, "Trs: Transferability reduced ensemble via promoting gradient diversity and model smoothness," in *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021.

[106] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*. PMLR, 2019, pp. 7472–7482.

[107] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. A. Mann, "Improving robustness using generated data," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[108] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu, "Bag of tricks for adversarial training," in *International Conference on Learning Representations*, 2020.

[109] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *International Conference on Learning Representations*, 2020.

[110] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.

[111] L. Gurobi Optimization, "Gurobi - the fastest solver - gurobi," Gurobi Optimization, LLC., 2020, https://www.gurobi.com/.

[112] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, and C.-J. Hsieh, "Towards stable and efficient training of verifiably robust neural networks," in *International Conference on Learning Representations*, 2020.

[113] M. König, H. H. Hoos, and J. N. van Rijn, "Speeding up neural network verification via automated algorithm configuration," in *ICLR 2021 Workshop on Security and Safety in Machine Learning Systems*, 2021.

[114] J. P. Ignizio and T. M. Cavalier, *Linear programming*. Prentice-Hall, Inc., 1994.

[115] S. Bak, C. Liu, and T. Johnson, "The second international verification of neural networks competition (vnn-comp 2021): Summary and results," *arXiv preprint arXiv:2109.00498*, 2021.

[116] K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen, and M. J. Kochenderfer, "Policy compression for aircraft collision avoidance systems," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference*. IEEE, 2016, pp. 1–10.

[117] D. Shriver, S. Elbaum, and M. B. Dwyer, "Dnnv: A framework for deep neural network verification," in *International Conference on Computer Aided Verification*. Springer, 2021, pp. 137–150.

[118] D. Zombori, B. Bánhelyi, T. Csendes, I. Megyeri, and M. Jelasity, "Fooling a complete neural network verifier," in *International Conference on Learning Representations*, 2020.

[119] K. Jia and M. Rinard, "Exploiting verified neural networks via floating point numerical error," in *International Static Analysis Symposium*. Springer, 2021, pp. 191–205.

[120] N. Jovanović, M. Balunović, M. Baader, and M. Vechev, "Certified defenses: Why tighter relaxations may hurt training?" *arXiv preprint arXiv:2102.06700*, 2021.

[121] Z. Shi, Y. Wang, H. Zhang, J. Yi, and C.-J. Hsieh, "Fast certified robust training with short warmup," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[122] B. Paulsen and C. Wang, "Linsyn: Synthesizing tight linear bounds for arbitrary neural network activation functions," in *28th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2022.

[123] B. Batten, P. Kouvaros, A. Lomuscio, and Y. Zheng, "Efficient neural network verification via layer-based semidefinite relaxations and linear cuts," in *International Joint Conference on Artificial Intelligence*, 2021, pp. 2184–2190.

[124] R. Mises and H. Pollaczek-Geiringer, "Praktische verfahren der gleichungsauflösung." *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 9, no. 1, pp. 58–77, 1929.

[125] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and accurate estimation of lipschitz constants for deep neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 423–11 434.

[126] B. Zhang, D. Jiang, D. He, and L. Wang, "Boosting the certified robustness of l-infinity distance nets," in *International Conference on Learning Representations*, 2022.

[127] N. Bansal, X. Chen, and Z. Wang, "Can we gain more from orthogonality regularizations in training deep networks?" in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[128] L. Huang, L. Liu, F. Zhu, D. Wan, Z. Yuan, B. Li, and L. Shao, "Controllable orthogonalization in training dnns," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6429–6438.

[129] J. Wang, Y. Chen, R. Chakraborty, and S. X. Yu, "Orthogonal convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 505–11 515.

[130] K. Hung, W. Fithian *et al.*, "Rank verification for exponential families," *The Annals of Statistics*, vol. 47, no. 2, pp. 758–782, 2019.

[131] G.-H. Lee, Y. Yuan, S. Chang, and T. Jaakkola, "Tight certificates of adversarial robustness for randomly smoothed classifiers," in *Advances in Neural Information Processing Systems*, 2019, pp. 4911–4922.

[132] J. Neyman and E. S. Pearson, "Ix. on the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, no. 694-706, pp. 289–337, 1933.

[133] A. Blum, T. Dick, N. Manoj, and H. Zhang, "Random smoothing might be unable to certify ℓ_∞ robustness for high-dimensional images," *J. Mach. Learn. Res.*, vol. 21, pp. 211–1, 2020.

[134] A. Kumar, A. Levine, T. Goldstein, and S. Feizi, "Curse of dimensionality on randomized smoothing for certifiable robustness," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5458–5467.

[135] J. Mohapatra, C.-Y. Ko, L. Weng, P.-Y. Chen, S. Liu, and L. Daniel, "Hidden cost of randomized smoothing," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 4033–4041.

[136] J. Hayes, "Extensions and limitations of randomized smoothing for robustness guarantees," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 786–787.

[137] L. Li, J. Zhang, T. Xie, and B. Li, "Double sampling randomized smoothing," in *International Conference on Machine Learning*, 2022.

[138] J. Jeong, S. Park, M. Kim, H.-C. Lee, D. Kim, and J. Shin, "Smoothmix: Training confidence-calibrated smoothed classifiers for certified robustness," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[139] L. Li, Z. Zhong, B. Li, and T. Xie, "Robustra: training provable robust neural networks over reference adversarial space," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 4711–4717.

[140] J. Jeong and J. Shin, "Consistency regularization for certified robustness of smoothed classifiers," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[141] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 190–11 201.

[142] M. Balunovic and M. Vechev, "Adversarial training and provable defenses: Bridging the gap," in *International Conference on Learning Representations*, 2020.

[143] R. Zhai, C. Dan, D. He, H. Zhang, B. Gong, P. Ravikumar, C.-J. Hsieh, and L. Wang, "Macer: Attack-free and scalable robust training via maximizing certified radius," in *International Conference on Learning Representations*, 2020.

[144] Z. Yang, L. Li, X. Xu, B. Kailkhura, T. Xie, and B. Li, "On the certified robustness for ensemble models and beyond," in *International Conference on Learning Representations*, 2022.

[145] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "RobustBench: a standardized adversarial robustness benchmark," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1*, J. Vanschoren and S. Yeung, Eds., 2021.

[146] F. Wu, L. Li, Z. Huang, Y. Vorobeychik, D. Zhao, and B. Li, "CROP: Certifying robust policies for reinforcement learning through functional smoothing," in *International Conference on Learning Representations*, 2022.

[147] L. Engstrom, D. Tsipras, L. Schmidt, and A. Madry, "A rotation and a translation suffice: Fooling cnns with simple transformations," *arXiv preprint arXiv:1712.02779*, vol. 1, no. 2, p. 3, 2017.

[148] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[149] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," in *International Conference on Learning Representations*, 2018.

[150] M. Fischer, M. Baader, and M. Vechev, "Certified defense to image transformations via randomized smoothing," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, 2020, pp. 8404–8417.

[151] L. Li, M. Weber, X. Xu, L. Rimanic, B. Kailkhura, T. Xie, C. Zhang, and B. Li, "TSS: Transformation-specific smoothing for robustness certification," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, p. 535–557.

[152] M. Pautov, N. Tursynbek, M. Munkhoeva, N. Muravev, A. Petiushko, and I. Oseledets, "CC-Cert: A probabilistic approach to certify general robustness of neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 7975–7983.

[153] Z. Hao, C. Ying, Y. Dong, H. Su, J. Song, and J. Zhu, "GSmooth: Certified robustness against semantic transformations via generalized randomized smoothing," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 8465–8483.

[154] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[155] M. Mirman, A. Hägele, P. Bielik, T. Gehr, and M. Vechev, "Robustness certification with generative models," in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 2021, pp. 1141–1154.

[156] E. Wong and J. Z. Kolter, "Learning perturbation sets for robust machine learning," in *International Conference on Learning Representations*, 2020.

[157] A. Levine and S. Feizi, "Robustness certificates for sparse adversarial attacks by randomized ablation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4585–4593.

[158] J. Jia, B. Wang, X. Cao, H. Liu, and N. Z. Gong, "Almost tight l0-norm certified robustness of top-k predictions against adversarial perturbations," in *International Conference on Learning Representations*, 2022.

[159] H. Han, K. Xu, X. Hu, X. Chen, L. Liang, Z. Du, Q. Guo, Y. Wang, and Y. Chen, "Scalecert: Scalable certified defense against adversarial patches with sparse superficial layers," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[160] A. Levine and S. Feizi, "(de) randomized smoothing for certifiable defense against patch attacks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6465–6475, 2020.

[161] H. Salman, S. Jain, E. Wong, and A. Madry, "Certified patch robustness via smoothed vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 137–15 147.

[162] C. Xiang, A. N. Bhagoji, V. Sehwag, and P. Mittal, "PatchGuard: A provably robust defense against adversarial patches via small receptive fields and masking," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2237–2254.

[163] C. Xiang, S. Mahloujifar, and P. Mittal, "PatchCleanser: Certifiably robust defense against adversarial patches for any image classifier," in *31st USENIX Security Symposium (USENIX Security)*, 2022.

[164] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, "Towards out-of-distribution generalization: A survey," *arXiv preprint arXiv:2108.13624*, 2021.

[165] A. Sinha, H. Namkoong, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," in *International Conference on Learning Representations*, 2018.

[166] A. Kumar, A. Levine, T. Goldstein, and S. Feizi, "Certifying model accuracy under distribution shifts," *arXiv preprint arXiv:2201.12440*, 2022.

[167] M. G. Weber, L. Li, B. Wang, Z. Zhao, B. Li, and C. Zhang, "Certifying out-of-domain generalization for blackbox functions," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 527–23 548.

[168] Y. Chen, S. Wang, Y. Qin, X. Liao, S. Jana, and D. Wagner, "Learning security classifiers with verified global robustness properties," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, p. 477–494.

[169] A. Levine and S. Feizi, "Deep partition aggregation: Provable defenses against general poisoning attacks," in *International Conference on Learning Representations*, 2021.

[170] E. Rosenfeld, E. Winston, P. Ravikumar, and J. Z. Kolter, "Certified robustness to label-flipping attacks via randomized smoothing," in *International Conference on Machine Learning*, 2020.

[171] M. Weber, X. Xu, B. Karlas, C. Zhang, and B. Li, "Rab: Provable robustness against backdoor attacks," in *44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, 22-26 May 2023*. IEEE, 2023.

[172] F. Wu, L. Li, C. Xu, H. Zhang, B. Kailkhura, K. Kenthapadi, D. Zhao, and B. Li, "COPA: Certifying robust policies for offline reinforcement learning against poisoning attacks," in *International Conference on Learning Representations*, 2022.

[173] T. Du, S. Ji, L. Shen, Y. Zhang, J. Li, J. Shi, C. Fang, J. Yin, R. Beyah, and T. Wang, "Cert-rnn: Towards certifying the robustness of recurrent neural networks," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 516–534.

[174] C.-Y. Ko, Z. Lyu, L. Weng, L. Daniel, N. Wong, and D. Lin, "Popqorn: Quantifying robustness of recurrent neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3468–3477.

[175] W. Ryou, J. Chen, M. Balunovic, G. Singh, A. Dan, and M. Vechev, "Scalable polyhedral verification of recurrent neural networks," in *International Conference on Computer Aided Verification*. Springer, 2021, pp. 225–248.

[176] G. Bonaert, D. I. Dimitrov, M. Baader, and M. Vechev, "Fast and precise certification of transformers," in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 2021, pp. 466–481.

[177] M. Mirman, A. Hägele, P. Bielik, T. Gehr, and M. Vechev, "Robustness certification with generative models," in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 2021, pp. 1141–1154.

[178] H. Zhang, M. Cheng, and C.-J. Hsieh, "Enhancing certifiable robustness via a deep model ensemble," *arXiv preprint arXiv:1910.14655*, 2019.

[179] M. Andriushchenko and M. Hein, "Provably robust boosted decision stumps and trees against adversarial attacks," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 997–13 008.

[180] H. Chen, H. Zhang, S. Si, Y. Li, D. Boning, and C.-J. Hsieh, "Robustness verification of tree-based models," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 317–12 328.

[181] Y. Wang, H. Zhang, H. Chen, D. Boning, and C.-J. Hsieh, "On lp-norm robustness of ensemble decision stumps and trees," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 104–10 114.

[182] R. Jia, A. Raghunathan, K. Göksel, and P. Liang, "Certified robustness to adversarial word substitutions," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 4127–4140.

[183] M. Ye, C. Gong, and Q. Liu, "Safer: A structure-free approach for certified robustness to adversarial word substitutions," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3465–3475.

[184] Y. Zhang, A. Albarghouthi, and L. D'Antoni, "Robustness to programmable string transformations via augmented abstract training," in *International Conference on Machine Learning*, 2020.

[185] Y. Zhang, A. Albarghouthi, and L. D'Antoni, "Certified robustness to programmable transformations in LSTMs," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1068–1083.

[186] P.-y. Chiang, M. Curry, A. Abdelkader, A. Kumar, J. Dickerson, and T. Goldstein, "Detection as regression: Certified object detection with median smoothing," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, 2020, pp. 1275–1286.

[187] M. Fischer, M. Baader, and M. Vechev, "Scalable certified segmentation via randomized smoothing," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3340–3351.

[188] W. Chu, L. Li, and B. Li, "TPC: Transformation-specific smoothing for point cloud models," in *Proceedings of the 39th International Conference on Machine Learning*, 2022, pp. 4035–4056.

[189] H. Liu, J. Jia, and N. Z. Gong, "Pointguard: Provably robust 3d point cloud classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6186–6195.

[190] R. Olivier and B. Raj, "Sequential randomized smoothing for adversarially robust speech recognition," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6372–6386.

[191] A. Kumar, A. Levine, and S. Feizi, "Policy smoothing for provably robust reinforcement learning," in *International Conference on Learning Representations*, 2022.

[192] B. Lütjens, M. Everett, and J. P. How, "Certified adversarial robustness for deep reinforcement learning," in *Conference on Robot Learning 2019*, ser. Proceedings of Machine Learning Research, vol. 100. PMLR, 2019.

[193] Y.-S. Wang, T.-W. Weng, and L. Daniel, "Verification of neural network control policy under persistent adversarial perturbation," *arXiv preprint arXiv:1908.06353*, 2019.

[194] P. Cousot and R. Cousot, "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints," in *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, 1977, pp. 238–252.

[195] V. Voráček and M. Hein, "Sound randomized smoothing in floating-point arithmetics," *arXiv preprint arXiv:2207.07209*, 2022.

[196] A. Mehra, B. Kailkhura, P.-Y. Chen, and J. Hamm, "How robust are randomized smoothing based defenses to data poisoning?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 244–13 253.

[197] M. Baader, M. Mirman, and M. Vechev, "Universal approximation with certified networks," in *International Conference on Learning Representations*, 2020.

[198] Z. Wang, A. Albarghouthi, G. Prakriya, and S. Jha, "Interval universal approximation for neural networks," *Proceedings of the ACM on Programming Languages*, vol. 6, no. POPL, pp. 1–29, 2022.

[199] Y. Wang, Z. Shi, Q. Gu, and C.-J. Hsieh, "On the convergence of certified robust training with interval bound propagation," in *International Conference on Learning Representations*, 2022.

[200] H. Ashtiani, V. Pathak, and R. Urner, "Black-box certification and learning under adversarial perturbations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 388–398.

[201] V. Sehwag, S. Wang, P. Mittal, and S. Jana, "Hydra: Pruning adversarially robust neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 655–19 666, 2020.

[202] A. Ruoss, M. Balunovic, M. Fischer, and M. Vechev, "Learning certified individually fair representations," *Advances in Neural Information Processing Systems 33 pre-proceedings*, 2020.

[203] C. Urban, M. Christakis, V. Wüstholz, and F. Zhang, "Perfectly parallel fairness certification of neural networks," *Proceedings of the ACM on Programming Languages*, vol. 4, no. OOPSLA, pp. 1–30, 2020.

[204] M. Kang, L. Li, M. Weber, Y. Liu, C. Zhang, and B. Li, "Certifying some distributional fairness with subpopulation decomposition," *arXiv preprint arXiv:2205.15494*, 2022.

[205] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 141–159.

[206] M. Qu and J. Tang, "Probabilistic logic neural networks for reasoning," *Advances in neural information processing systems*, vol. 32, 2019.
[207] Y. Xie, Z. Xu, M. S. Kankanhalli, K. S. Meel, and H. Soh, "Embedding symbolic knowledge into deep networks," *Advances in neural information processing systems*, vol. 32, 2019.
[208] Y. Zhang, X. Chen, Y. Yang, A. Ramamurthy, B. Li, Y. Qi, and L. Song, "Efficient probabilistic logic reasoning with graph neural networks," in *International Conference on Learning Representations*, 2019.
[209] R. S. Hallyburton, Y. Liu, Y. Cao, Z. M. Mao, and M. Pajic, "Security analysis of camera-lidar fusion against black-box attacks on autonomous vehicles," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022.
[210] D. Wakabayashi, "Self-driving uber car kills pedestrian in arizona, where robots roam," https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html, accessed: 2021-12-02.
[211] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge, England: Cambridge university press, 2004.
[212] R. Bunel, A. De Palma, A. Desmaison, K. Dvijotham, P. Kohli, P. Torr, and M. P. Kumar, "Lagrangian decomposition for neural network verification," in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 370–379.
[213] K. Y. Xiao, V. Tjeng, N. M. M. Shafiullah, and A. Madry, "Training for faster adversarial robustness verification via inducing reLU stability," in *International Conference on Learning Representations*, 2019.
[214] F. Croce, M. Andriushchenko, and M. Hein, "Provable robustness of relu networks via maximization of linear regions," in *the 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 2057–2066.
[215] F. Croce and M. Hein, "Provable robustness against all adversarial l_p-perturbations for $p \geq 1$," in *International Conference on Learning Representations*, 2020.
[216] S. Wang, Y. Chen, A. Abdou, and S. Jana, "Mixtrain: Scalable training of verifiably robust neural networks," *arXiv preprint arXiv:1811.02625*, 2018.
[217] S. Lee, W. Lee, J. Park, and J. Lee, "Towards better understanding of training certifiably robust models against adversarial examples," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
[218] M. Z. Horváth, M. N. Mueller, M. Fischer, and M. Vechev, "Boosting randomized smoothing with variance reduced classifiers," in *International Conference on Learning Representations*, 2022.
[219] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.
[220] N. Carlini, F. Tramer, J. Z. Kolter *et al.*, "(certified!!) adversarial robustness for free!" *arXiv preprint arXiv:2206.10550*, 2022.
[221] H. Feng, C. Wu, G. Chen, W. Zhang, and Y. Ning, "Regularized training and tight certification for randomized smoothed classifier with provable robustness," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2020.
[222] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.

APPENDIX A
SCALABILITY AND TIGHTNESS MEASUREMENTS

This appendix contains more discussion on the scalability and tightness characterization in Sec. III.

Scalability measured by time complexity. Note that the time complexity for DNN inference is $O(lw^2)$. As we can see in Table I, all complete verification approaches have exponential time complexity $O(2^{lw})$ which coincides with the theoretical scalability barriers [23], [39]. The $\text{poly}(l, w)$ means a time complexity higher than $O(lw^3)$. All approaches for smoothed DNNs have complexity $O(Slw^2)$, which is because the sampling time cost is much higher than the actual bound computation whose time complexity is subsumed.

Deails on tightness ranks. For general DNNs, we rank the tightness from T_1 to T_7 where T_7 is the tightest. $T_1 < T_2 < T_3$ comes from benchmark results, $T_3 < T_4 < T_5$ comes from theoretical analyses [40], and $T_5 < T_6$ and $T_6 < T_7$ come from empirical observations in [75] and [72] respectively. For smoothed DNNs we rank the tightness from ST_1 to ST_4 based on existing theoretical analyses: $ST_1 < ST_2$ comes from [93], [94], $ST_2 < ST_3$ comes from [28], [94], [98], and $ST_3 < ST_4$ comes from [100].

APPENDIX B
OMITTED ILLUSTRATIONS

This appendix includes the omitted figure illustrations.

A. Perturbation Region of ℓ_p Adversary

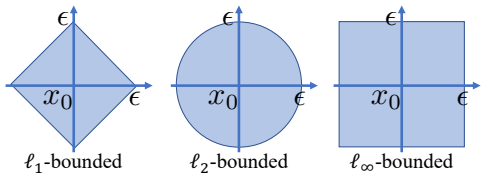


Fig. 5. An (ℓ_p, ϵ)-adversary crafts perturbed input from ℓ_p-bounded region centered at clean input x_0. From left to right are ℓ_1-, ℓ_2-, and ℓ_∞-bounded perturbation regions in 2D space with radius ϵ.

B. ReLU Relaxation with Single Input Variable

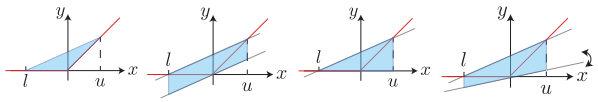
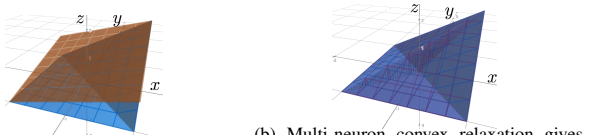


Fig. 6. Different linear relaxations for ReLU shown as blue region. (a) shows the tightest single-neuron polytope for ReLU which is used in [27], [46], [52], [71]; (b) is used in [38], [39], [64]; (c) is used in [38], [64]; (d) shows that the slope of lower bound can be dynamically adjusted or optimized [38], [59], [62], [64].

C. ReLU Relaxation with Multiple Input Variables



(a) Single ReLU polytope (shown in brown) gives looser upper bound for z.

(b) Multi-neuron convex relaxation gives tighter upper bound for z (shown as two dark blue facets).

Fig. 7. Comparison of convex relaxation for $z = \text{ReLU}(x + y)$ (shown as bottom blue surface in (a)), where $x, y \in [-1, 1]$. Vertical axis is the z-axis.

APPENDIX C
DETAILS ON LINEAR INEQUALITY BASED VERIFICATION

This appendix entails the omitted details of linear inequality verification approaches introduced in Sec. IV-B2.

More details on polyhedra abstraction. FAST-LIN [39] uses a parallel line as the lower bound as shown in Fig. 6b. CROWN [38] and DEEPPOLY [64] both support adjustable lower bound. They both use $y = \lambda x$ with adjustable $\lambda \in [0, 1]$ as the lower bound, while their heuristics for determining λ are slightly different. FROWN [62] and α-CROWN [59] deploy

gradient-based optimization on lower bound slope λ to improve tightness.

These approaches maintain the linear bound for each layer k in the form of $\boldsymbol{L}_k x + b_{L,k} \leq z_k(x) \leq \boldsymbol{U}_k x + b_{U,k}$ for any $x \in B_{p,\epsilon}(x_0)$. From the bound for layer k, we can deduct the bound after affine mapping $\hat{z}_{k+1} = \boldsymbol{W}_k z_k + b_k$:

$$\begin{aligned}&(\boldsymbol{W}_k^+ \boldsymbol{L}_k + \boldsymbol{W}_k^- \boldsymbol{U}_k)x + \boldsymbol{W}_k^+ b_{L,k} + \boldsymbol{W}_k^- b_{U,k} + b_k \\ \leq &\hat{z}_{k+1}(x) \\ \leq &(\boldsymbol{W}_k^+ \boldsymbol{U}_k + \boldsymbol{W}_k^- \boldsymbol{L}_k)x + \boldsymbol{W}_k^+ b_{U,k} + \boldsymbol{W}_k^- b_{L,k} + b_k.\end{aligned} \tag{5}$$

Then, they compute the activation value bound l_{k+1} and u_{k+1} for $\hat{z}_{k+1}$, and compute the linear bound for $z_{k+1}(x) = \mathrm{ReLU}(\hat{z}_{k+1}(x))$ using ReLU lower and upper bound respectively. By repeating the process, they finally bound the last layer $\hat{z}_l$, i.e., the model f itself.

Zonotope abstraction. Zonotope is another type of over-approximation or abstract interpretation domain that can be propagated layer by layer efficiently [25], [37], [66]–[68]. Zonotope abstraction has the same efficiency and slightly inferior tightness compared to polyhedra abstraction [40].

Duality-based approaches. Since the robustness verification can be viewed as an optimization problem (Problem 1), we can consider its Lagrangian dual problem. Especially, since Problem 1 is a minimization problem, any feasible dual solution provides a valid lower bound of the primal problem and therefore a valid verification. Moreover, the dual problem is always convex [211]. Typical duality-based approaches are WK [27], [71], D-LP [70], PVT [69], and LAGRANGIAN DECOMPOSITION [212] where WK is proved to share equivalent tightness with polyhedra abstraction approaches, and the others are proved to share equivalent tightness with linear programming based approaches [40].

APPENDIX D
ILLUSTRATION OF ROBUST TRAINING APPROACHES

Regularization-based training. For complete verification, Xiao et al [213] find that the number of branches is upper bounded by the number of unstable neurons (see Def. 4) which motivates a regularization term to increase the ReLU neuron's stability for training. For complete verification based on linear region traversal, we can train with a regularization term maximizing the margin to non-robust regions [214], [215]. The Lipschitz and curvature verification favor small Lipschitz constant and small curvature bounds respectively. Therefore, the corresponding robust training approaches explicitly penalize large Lipschitz or curvature bounds [80], [81], [83], [90].

Relaxation-based training. For linear relaxation based verification approaches, models with tight linear relaxation bounds are favored. To train such models, corresponding robust training approaches usually use the computed bounds from linear relaxation as the training objective to explicitly improve the bound tightness. This idea is similar to the powerful empirical defense named adversarial training [18] which uses effective attacks to approximately find "most adversarial" example $\max_{x \in B_{p,\epsilon}(x_0)} \mathcal{L}(f_\theta(x), y_0)$ and minimize model weights θ w.r.t. it. In relaxation-based training, instead, we compute an upper bound of $\max_{x \in B_{p,\epsilon}(x_0)} \mathcal{L}(f_\theta(x), y_0)$ and minimize it. The bound can be derived from IBP [61], [121], polyhedra-based [62], [112], [142], zonotope-based [67], or duality-based verification [27], [69], [139]. Some useful training tricks are: combining relaxation-based loss with standard loss to improve benign accuracy [61], [112], [216], applying relaxation on some layers but not all to balance benign accuracy and certified robustness [142], specialized weight initialization and training scheduling [121], and using reference space to guide the relaxation [139]. An intriguing phenomenon of relaxation-based training is that tighter relaxation, when used as the training objective, may not lead to more certifiably robust models [120], while the loosest IBP relaxation can achieve almost the highest certified robustness. A conjecture is that tighter relaxation may lead to a less smooth loss landscape containing discontinuities or sensitive regions which poses challenges for gradient-based training [120], [217]. Theoretical understanding of relaxation-based training is still lacking. Note that solver based and branch-and-bound based complete verification usually use linear relaxations for bounding. Therefore, models trained with these relaxation-based training approaches can usually be efficiently certified by these complete verification approaches [46], [58].

Augmentation-based training. Since randomized smoothing based verification favors models to perform well for noisy inputs, to obtain high certified robustness, we can train the DNNs with noisy inputs, resulting in augmentation-based training [22], [92], [93]. Built upon such augmentation-based training, later approaches combine augmentation with regularization terms to encourage the prediction stability/consistency when the input noise is added [138], [140], [143]. Strategic training regularization combined with augmentation and ensemble is effective and achieves the state-of-the-art certified robustness against ℓ_2 adversary [144], [218]. Adversarial training combined with augmentation [98], and training unlabeled data [141] are also shown effective. Recently, diffusion models [219], which intrinsically possess the denoising ability, are leveraged to build models for randomized smoothing [220]. They achieve competitive performance though require large model size which results in large inference overhead.

APPENDIX E
BENCHMARK EVALUATION DETAILS

Experiment environment. Our toolkit implementation is based on PYTORCH [222]. In the toolkit, we tend to integrate the original implementations released by the authors when it is available; otherwise, we implement and optimize them to match the reported performance. We run the evaluation on a 24-core Intel Xeon Platinum 8259CL CPU running at 2.50 GHz with a single NVIDIA Tesla T4 GPU.

A. Comparison of Deterministic Verification

We present a thorough comparison of representative *deterministic verification approaches* in Table III.

We evaluate on 7 different DNNs on CIFAR-10. Among them, 3 models (FCNNA - FCNNC) are fully-connected

TABLE III

Certified accuracy ON CIFAR-10 CERTIFIED BY *deterministic verification approaches* FOR DNNS. FAILING TO VERIFY OR EXCEEDING 60 s VERIFICATION TIME LIMIT IS COUNTED AS "NON-ROBUST". 0% CERTIFIED ACCURACY MEANS TOO LOOSE OR TOO SLOW TO VERIFY ALL INPUT SAMPLES. THE VERIFICATION IS AGAINST ℓ_∞ ADVERSARY WITH RADIUS $\epsilon = 8/255$. WE INCLUDE MODEL ACCURACY UNDER PGD ATTACK AS THE UPPER BOUND OF THE CERTIFIED ACCURACY. THE **BOLDED** NUMBERS MARK THE HIGHEST ONES AMONG VERIFICATION APPROACHES.

Verification Approach					FCNNA		FCNNB		FCNNC		CNNA		CNNB		CNNC		CNND	
Category				Name	adv	cadv	adv	cadv	adv	cadv	adv	cadv	adv	cadv	adv	cadv	adv	cadv
Complete	Solver-Based			BOUNDED MILP [46]	**19%**	**27%**	1%	**25%**	0%	0%	0%	**34%**	0%	**36%**	0%	0%	0%	0%
	Branch-and-Bound			AI^2 [37]	**19%**	**27%**	**7%**	23%	0%	22%	**8%**	**34%**	0%	20%	0%	14%	0%	0%
Incomplete	Linear Relaxtion	Linear Programming		LP-FULL [39], [40]	15%	**27%**	6%	**25%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
		Linear Inequality	Interval	IBP [61]	0%	**27%**	0%	**25%**	0%	**30%**	0%	**34%**	0%	35%	0%	**38%**	0%	**28%**
			Polyhedra	FAST-LIN [39]	15%	25%	4%	18%	0%	19%	3%	26%	0%	15%	0%	7%	0%	0%
				CROWN [38]	15%	**27%**	6%	20%	0%	22%	**8%**	33%	**1%**	20%	0%	0%	0%	0%
				CNN-CERT [101]	15%	**27%**	5%	20%	0%	0%	7%	33%	0%	20%	0%	0%	0%	0%
				CROWN-IBP [112]	9%	**27%**	0%	22%	0%	28%	0%	**34%**	0%	31%	0%	32%	0%	25%
				DEEPPOLY [64]	15%	**27%**	6%	20%	0%	22%	**8%**	33%	**1%**	20%	0%	7%	0%	0%
				REFINEZONO [25]	0%	**27%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
			Duality	WK [27], [71]	15%	25%	4%	18%	0%	19%	3%	26%	0%	15%	0%	7%	0%	5%
		Multi-Neuron Relaxation		K-RELU [24]	15%	**27%**	2%	23%	0%	0%	0%	32%	0%	0%	0%	0%	0%	0%
	SDP			SDPVERIFY [79]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
				LMIVERIFY [77]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	Lipschitz	General Lipschitz		OP-NORM [11], [83]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
				FASTLIP [39]	12%	**27%**	0%	17%	0%	17%	0%	24%	0%	0%	0%	0%	0%	0%
				RECURJAC [84]	14%	**27%**	2%	17%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Accuracy under PGD (Upper Bound of Robust Accuracy)					22%	28%	23%	26%	19%	34%	34%	34%	33%	39%	36%	40%	41%	31%
Clean Accuracy					33%	31%	37%	30%	26%	39%	44%	46%	53%	48%	52%	46%	66%	46%

TABLE IV

Certified accuracy ON CIFAR-10 CERTIFIED BY *probabilistic approaches* FOR SMOOTHED DNNS. THE MODELS ARE TRAINED WITH DIFFERENT "ROBUST TRAINING" APPROACHES AND SMOOTHED WITH DISTRIBUTIONS LABELLED AS "SMOOTH DIST.". NUMBERS WITHIN EACH BIG CELL UNDER THE RIGHTMOST COLUMN ARE COMPARABLE. THE **BOLDED** NUMBERS MARK THE HIGHEST ONES WITHIN EACH GROUP.

Adversary	Model Structure	Verification Approach	Robust Training	Smooth Dist.	Certified Robust Accuracy under Perturbation Radius ϵ					
				$\epsilon =$	0.25	0.50	0.75	1.00	1.25	1.50
ℓ_2	Wide ResNet 40-2	Differential Privacy Based [92]	Data Augmentation [22], [28]	Gaussian	34.2%	14.8%	6.8%	2.2%	0.0%	0.0%
		Neyman-Pearson [22], [28], [95], [96]			**68.8%**	**46.8%**	**36.0%**	**25.4%**	**19.8%**	**15.6%**
		f-Divergence [94]			62.2%	41.8%	27.2%	19.2%	14.2%	11.4%
	ResNet-110	Neyman-Pearson [22], [28], [95], [96]	Data Augmentation [22], [28]		61.2%	43.2%	32.0%	22.4%	17.2%	14.0%
			Adversarial Training [98]		73.0%	57.8%	48.2%	**37.2%**	33.6%	**28.2%**
			Adversarial + Pretraining [98], [141]		**81.8%**	**62.6%**	**52.4%**	37.2%	**34.0%**	30.2%
			MACER [143]		68.8%	52.6%	40.4%	33.0%	27.8%	25.0%
			ADRE [221]		68.0%	50.2%	37.8%	30.2%	23.0%	17.0%
				$\epsilon =$	0.5	1.0	1.5	2.0	3.0	4.0
ℓ_1	Wide ResNet 40-2	Differential Privacy Based [92]	Data Augmentation [22], [28]	Laplace	43.0%	20.8%	12.2%	7.2%	1.4%	0.0%
		Rényi Divergence [93]			58.2%	39.4%	27.0%	16.8%	9.2%	4.0%
		Neyman-Pearson [22], [28], [95], [96]			58.4%	39.6%	27.0%	17.2%	9.2%	4.2%
				Uniform	**69.2%**	**56.6%**	**48.0%**	**39.4%**	**26.0%**	**20.4%**
				$\epsilon =$	1/255	2/255	4/255	8/255		
ℓ_∞	Wide ResNet 40-2	Neyman-Pearson [22], [28], [95], [96]	Data Augmentation [22], [28]	Gaussian	71.4%	52.0%	29.0%	12.8%		
			Adversarial Training [98]		**83.2%**	**65.0%**	**49.6%**	**25.4%**		

networks, and 4 models (CNNA - CNND) are convolutional neural networks. The number of neurons ranges from 50 (FCNNA) to about 200,000 (CNND). For each DNN structure, we train two sets of weights: `adv`—PGD adversarial training with $\epsilon = 8/255$; `cadv`—CROWN-IBP training with $\epsilon = 8/255$, where ϵ is the ℓ_∞ attack radius. The PGD adversarial training [18] is a strong empirical defense, and CROWN-IBP [112] is a strong robust training approach. For PGD adversarial training, following the literature [18], [21], we set the attack step size to be $\epsilon/50$, attack iterations to be 100 with random initialization, and train for 40 epochs with 0.1 learning rate and SGD optimizer. For CROWN-IBP, we use the official code release [112] and default hyperparameters: 100 epochs with Adam optimizer and 5×10^{-4} learning rate on MNIST, and 200 epochs with SGD optimizer and 0.001 learning rate on CIFAR-10. More hyperparameters can be found in our open-source toolbox. We choose these training configurations to reflect two common types of models on which verification approaches are used: empirically defended models and robustly trained models. All models are trained to reach their expected robustness as reported in the corresponding papers. We defer the detailed model structure and statistics to our website.

Evaluation protocol. We measure the performance of verification approaches by their **certified accuracy** w.r.t. ℓ_∞ radius $\epsilon = 8/255$. ℓ_∞ adversary is supported by most deterministic verification approaches. The certified accuracy, as a measurement of certified robustness, is defined as

$$\text{certacc} := \frac{\text{\# samples verified to be robust}}{\text{\# number of all samples}}. \tag{6}$$

On each dataset, we uniformly sample 100 test samples as the fixed set for evaluation. We limit the running time to 60 s per instance (so that verifying all 14 benchmark models with each approach takes about one day) and count timeout instances as "not verified" to favor efficient and practical verification approaches. This time limit is aligned with common settings. For example, the recent competition (VNN-COMP 2021 [200]) for complete verification sets 6-hour as the time limit. For a fair comparison, we relax this time limit from 6 hours to one day since we benchmark multiple models together. Moreover, running tools with the one-day time limit per approach takes overall around 2.5 months considering around 20 approaches and all settings. Therefore, for time and energy concerns we did not benchmark with longer time limits. Practical users can explore other time limits with our open-source toolkit. We also report the robust accuracy under empirical attack (PGD attack with 100 steps, step size $\epsilon/50$, and random starts following [18], [21]), which upper bounds certified accuracy.

Table III shows certified accuracy on CIFAR-10 for deterministic approaches. Each row corresponds to a verification approach, PGD attack, or clean accuracy. More results such as average certified robustness radius, average running time,

TABLE V
ℓ_∞ CERTIFIED ROBUST ACCURACY W.R.T. DIFFERENT RADII r'S (OUR METHOD SHOWN IN GRAY).

Dataset	Model	Certification Approach	Clean Accuracy	Certified Accuracy under Radius r											
				1/255	2/255	3/255	4/255	5/255	6/255	7/255	8/255	9/255	10/255	11/255	12/255
MNIST	Gaussian Augmentation	Neyman-Pearson	99.1%	**98.1%**	97.4%	**96.6%**	95.8%	**95.2%**	92.4%	89.4%	85.2%	80.8%	73.2%	64.0%	50.7%
		Our Method		**98.1%**	**97.5%**	**96.6%**	**96.1%**	**95.2%**	**92.7%**	**90.5%**	**86.8%**	**82.8%**	**77.6%**	**68.8%**	**60.0%**
	Consistency [140]	Neyman-Pearson	98.5%	**98.3%**	**98.2%**	**97.2%**	**96.4%**	95.4%	93.9%	91.5%	88.3%	83.9%	78.7%	71.2%	62.7%
		Our Method		**98.3%**	**98.2%**	**97.2%**	**96.4%**	**95.6%**	**94.3%**	**92.7%**	**89.0%**	**86.0%**	**81.9%**	**75.5%**	**67.5%**
			Clean Accuracy	Certified Accuracy under Radius r											
				0.5/255	1/255	1.5/255	2/255	2.5/255	3/255	3.5/255	4/255	4.5/255	5/255	5.5/255	6/255
CIFAR-10	Gaussian Augmentation	Neyman-Pearson	65.6%	52.0%	45.3%	41.1%	36.3%	32.6%	26.7%	21.9%	18.1%	15.1%	10.9%	8.9%	6.1%
		Our Method		**52.3%**	**45.6%**	**41.5%**	**37.6%**	**33.8%**	**28.8%**	**23.7%**	**19.5%**	**17.2%**	**13.9%**	**10.5%**	**8.1%**
	Consistency [140]	Neyman-Pearson	52.6%	47.1%	**45.5%**	**43.6%**	40.6%	38.3%	36.0%	33.4%	30.5%	28.5%	25.2%	22.0%	20.3%
		Our Method		**47.2%**	**45.5%**	**43.6%**	**40.9%**	**38.9%**	**36.9%**	**34.5%**	**31.9%**	**29.5%**	**28.1%**	**24.9%**	**22.0%**

and results on MNIST are on our website. Findings from our evaluation are discussed in Sec. VI-A.

B. Comparison of Probabilistic Verification

We present a thorough comparison of representative *probabilistic verification approaches* for smoothed DNNs with different smoothing distributions and robust training approaches. We either fix the robust training part and vary the verification approaches or the other way around.

Evaluation protocol. We use ResNet-110 and Wide ResNet 40-2 as the model architecture. $n = 1,000$ samples are used for selecting the top label; $N = 100,000$ samples are used for certification. For all robust training approaches, we adopt default hyperparameters as reported in corresponding papers. The failure probability is set to $1 - \alpha = .001$. We uniformly draw 500 samples from the test set for evaluation. All the above settings follow common practice in [22], [28].

Comparison results and discussion. We show results on CIFAR-10 in Table IV. Results on ImageNet can be found on our benchmark website. Findings from our evaluation are discussed in Sec. VI-A.

APPENDIX F
TIGHTER CERTIFICATION AGAINST ℓ_∞ ADVERSARY

We extend the very recent double sampling randomized smoothing in [137] to provide robustness certification for smoothed DNNs by sampling the statistics of the smoothed DNNs' prediction using both the original smoothing distribution $\mathcal{P}$ and an additional smoothing distribution $\mathcal{Q}$ that shares the same form but a different variance from $\mathcal{P}$'s variance. Note that we leverage additional information—the prediction probability under $\mathcal{Q}$. In contrast, the zeroth-order methods only leverage the sampling probability information from $\mathcal{P}$. The extension methodology is listed in Appendix H.3 of [137].

Now we systematically evaluate our extension of the double sampling method and demonstrate that it achieves tighter certification than the classical Neyman-Pearson-based certification (the tightest zeroth-order information approach) against ℓ_∞-bounded perturbations on MNIST and CIFAR-10.

Smoothing Distributions. For a given distribution $\mathcal{D}$, we let $\mathrm{Std}(\mathcal{D})$ be its average component-wise standard deviation: $\mathrm{Std}(\mathcal{D}) := \sqrt{\frac{1}{d}\mathbb{E}_{\delta\sim\mathcal{D}}[\|\delta\|_2^2]}$ as first used in [28]. We set $\mathrm{Std}(\mathcal{P}) = 0.75$, $\mathrm{Std}(\mathcal{Q}) = 0.6$ on both MNIST and CIFAR-10. We use generalized Gaussian as the smoothing distribution following [137] where $d-k = 8$ on MNIST and $d-k = 12$ on CIFAR-10. Note that we did not finetune these hyperparameters and we expect the existence of better hyperparameters.

Models. We train the models using both commonly-used Gaussian augmentation [22] and state-of-the-art Consistency training [140]. On all datasets, we use the default model structures and hyperparameters. All models are trained with the original smoothing distribution $\mathcal{P}$.

Baselines. We consider the Neyman-Pearson-based certification method as the baseline. For both baseline and our method, we set the certification confidence to be $1 - 2\alpha = 99.8\%$. We use 10^5 samples for estimating P_A and Q_A per instance. Note that Neyman-Pearson certification does not use the information from additional distribution and all 10^5 samples are used to estimate the interval of P_A. In our method, we use 5×10^4 samples to estimate the interval of P_A and the rest 5×10^4 samples for Q_A.

Metric. We uniformly draw 1000 samples from the test set, and report the *certified accuracy* under each radius r as defined in Eqn. (6). We also report the benign accuracy of the smoothed classifier. Both settings and the metric follow the standard evaluation protocol in literature [22], [28].

Main Results. The experimental results for certification against ℓ_∞ adversary are shown in Table V. We observe that, for *all* evaluated models, our method yields significantly higher certified accuracy. For example, when $r = 12/255$ our method improves the MNIST robust accuracy from 50.7% to 60.0%; when $r = 5/255$ our method improves CIFAR-10 robust accuracy from 25.2% to 28.1%. Thus, leveraging additional information can indeed provide tighter robustness certification over zeroth-order certification approaches for smoothed DNNs not only against ℓ_1 and ℓ_2 adversaries but also against ℓ_∞ adversary.