Data-Driven Assessment of Deep Neural Networks with Random Input Uncertainty

Brendon G. Anderson and Somayeh Sojoudi

***Abstract*— When using deep neural networks to operate safety-critical systems, assessing the sensitivity of the network outputs when subject to uncertain inputs is of paramount importance. Such assessment is commonly done using reachability analysis or robustness certification. However, certification techniques typically ignore localization information, while reachable set methods can fail to issue robustness guarantees. Furthermore, many advanced methods are either computationally intractable in practice or restricted to very specific models. In this paper, we develop a data-driven optimization-based method capable of simultaneously certifying the safety of network outputs and localizing them. The proposed method provides a unified assessment framework, as it subsumes state-of-the-art reachability analysis and robustness certification. The method applies to deep neural networks of all sizes and structures, and to random input uncertainty with a general distribution. We develop sufficient conditions for the convexity of the underlying optimization, and for the number of data samples to certify and localize the outputs with overwhelming probability. We experimentally demonstrate the efficacy and tractability of the method on a deep ReLU network.**

I. INTRODUCTION

Neural networks stand out for their high performance and flexibility in making data-driven predictions and decisions. However, researchers have shown that many networks are highly sensitive to inputs altered by random or adversarial perturbations [1], [2], [3]. This can result in misclassifications or outputs entering an unsafe region of the output space, as well as a large uncertainty propagation from inputs to outputs. When employing neural networks in safety-critical systems, e.g., autonomous vehicles [4], [5], this sensitive behavior is intolerable. Consequently, much effort has been placed on localizing neural network outputs and certifying their safety in the presence of input uncertainty.

In localization, one seeks to find a subset of the output space that contains the possible outputs, whereas certification is the decision problem of assessing whether the outputs enter an unsafe region or not. These two problems are clearly related: exact localization of the network outputs can be used to certify their safety. However, this approach has two problems: 1) the output set is generally intractable to compute [6], and 2) certification typically amounts to solving an NP-hard, nonconvex optimization over the output set [7]. As a result, these assessment methods have largely been treated separately in the settings of output set estimation (see also, reachability analysis) [6], [8], and robustness certification [9], [10], [11], and these remain active areas of research.

The authors are with the University of California, Berkeley. Emails: {bganderson,sojoudi}@berkeley.edu.

This work was supported by grants from AFOSR, ONR, and NSF.

A. Related Works

In this paper, we consider random input uncertainty with a known or sufficiently well-modeled probability distribution. Despite the large body of work on assessing sensitivity to adversarial inputs, random uncertainty often models reality more accurately than worst-case uncertainty [12]. Various methods to localize and certify outputs in the presence of random inputs have been proposed [12], [13], [14], [15], [16]. However, all these approaches rely on trading off theoretical guarantees with computational complexity, and on making restrictive assumptions about either the network structure, e.g., ReLU activations, or the input distribution, e.g., Gaussian or independent coordinates.

To overcome the above limitations, we develop a novel method using a sampling-based approach called *scenario optimization*, which is computationally tractable, provides probabilistic guarantees, and can be applied to arbitrary networks and input distributions. The scenario approach has recently been used in both output set estimation [17] and in robustness certification [18]; however, these methods alone fail to completely assess network sensitivity. In particular, [17] localizes outputs but may fail to determine their safety, as we demonstrate in Section V-B. Furthermore, this method is restricted to localizing outputs into a norm ball, lacking the generality needed to well-approximate the more complicated (and typically nonconvex) outputs sets of neural networks in practice. On the other hand, [18] can efficiently issue robustness certificates, but completely ignores the aspect of localizing the outputs in order to do so.

B. Contributions

In this paper, we formulate a unified framework that simultaneously localizes network outputs and certifies their safety with high-probability guarantees. The assessment procedure is data-driven, and subsumes the output set estimation method in [17] and the robustness certification method in [18] as special cases. Our method is completely general: it may be applied to any neural network and any input distribution. The outputs can be localized into a general class of sets, not just norm balls, and we obtain sufficient conditions on this class to ensure that the procedure amounts to a convex scenario optimization problem. Furthermore, we show that the resulting localization and robustness certification can be made to hold with overwhelming probability upon using a sufficient number of sampled data points in the scenario optimization. We illustrate the assessment procedure on a deep ReLU network, demonstrating the user's control over the strength of the probabilistic guarantees and the varying levels of certification and localization. Finally, we show that our

unified approach of localizing and certifying simultaneously can issue robustness certificates in cases where the two-step process of localizing then certifying cannot.

C. Outline

Various notions of robustness are introduced and used to formalize the problem in Section II. In Sections III and IV, we connect the concepts of certification and localization of network outputs, and show that both can be assessed with guarantees using a single data-driven convex optimization problem with sufficiently many samples. We illustrate the results in Section V and conclude in Section VI.

D. Notations

The set of real numbers is denoted by $\mathbb{R}$. Given a set $\mathcal{X}$, we denote its power set (the set of all subsets of $\mathcal{X}$) by $\mathcal{P}(\mathcal{X})$. The Minkowski sum of sets $\mathcal{X}$ and $\mathcal{Y}$ is defined as $\mathcal{X}+\mathcal{Y}=\{x+y : x\in\mathcal{X},\ y\in\mathcal{Y}\}$. Furthermore, we define $\mathbb{R}_{++}=\{x\in\mathbb{R} : x>0\}$. For a function $f\colon \mathbb{R}^n\to\mathbb{R}^m$, we write the image of a set $\mathcal{X}\subseteq\mathbb{R}^n$ under f as $f(\mathcal{X})=\{f(x)\in\mathbb{R}^m : x\in\mathcal{X}\}$. Finally, for a norm $\|\cdot\|$ on $\mathbb{R}^n$ we denote its dual norm by $\|\cdot\|_*$, where $\|y\|_*=\sup_{\|x\|\le 1} x^\top y$. We assume throughout that optimization problems are attained by a solution.

II. PROBLEM STATEMENT

A. Network Description, Safe Set, and Safety Level

In this paper, we consider a neural network $f\colon \mathbb{R}^{n_x}\to\mathbb{R}^{n_y}$ with arbitrary structure and parameters. We assume that the input to the neural network is a random variable X with a given distribution $\mathbb{P}_X$. The support of $\mathbb{P}_X$ is called the *input set*, which is denoted by $\mathcal{X}\subseteq\mathbb{R}^{n_x}$. The *output set* of the network is defined to be $\mathcal{Y}=f(\mathcal{X})\subseteq\mathbb{R}^{n_y}$.

Next, consider a given convex polyhedral[1] *safe set* $\mathcal{S}=\{y\in\mathbb{R}^{n_y} : Ay+b\ge 0\}$, where $A\in\mathbb{R}^{n_s\times n_y}$ and $b\in\mathbb{R}^{n_s}$. By applying the results of this paper to each row of A and b individually, we may assume without loss of generality that $n_s=1$, henceforth setting $A=a^\top\in\mathbb{R}^{1\times n_y}$ and $b\in\mathbb{R}$. The elements of $\mathcal{S}$ are considered to be *safe*. For a point $y\in\mathbb{R}^{n_y}$, the value $s(y)=a^\top y+b$ is called the *safety level* of y. The point y is safe if and only if its safety level is nonnegative.

B. Various Notions of Robustness

We now use the safety level to define three notions of robustness for the neural network.

1) Deterministic Robustness Level: The *deterministic robustness level* of the network is defined as

$$r^*=\inf_{y\in\mathcal{Y}} a^\top y+b. \tag{1}$$

If the deterministic robustness level is nonnegative, then $\mathcal{Y}\subseteq\mathcal{S}$, which implies that the random output $Y=f(X)$ is safe with probability one. This notion of robustness coincides with that used when considering adversarial inputs [19], [10], [11].

2) Approximate Robustness Level: Although the deterministic robustness level (1) can issue strong guarantees about the safety of the network output, computing its value r^* amounts to solving an intractable nonconvex optimization problem, since $\mathcal{Y}$ is generally a nonconvex set. Instead of computing r^*, we can consider approximating it by

$$\hat{r}(\hat{\mathcal{Y}})=\inf_{y\in\hat{\mathcal{Y}}} a^\top y+b, \tag{2}$$

where $\hat{\mathcal{Y}}\subseteq\mathbb{R}^{n_y}$, termed the *surrogate output set*, is more tractable than $\mathcal{Y}$, and preferably convex. We call (2) the *approximate robustness level* of the network. If $\hat{\mathcal{Y}}$ is chosen to cover the output set $\mathcal{Y}$, then $\hat{r}(\hat{\mathcal{Y}})\le r^*$. In this case, if the approximate robustness level is nonnegative, then the random output $Y=f(X)$ is safe with probability one.

3) Probabilistic Robustness Level: The notion of deterministic robustness is too strong for many applications, particularly those with random input uncertainty [12]. Therefore, for a prescribed probability level $\epsilon\in[0,1]$ we define the *probabilistic robustness level* of the network:

$$\bar{r}(\epsilon)=\sup\{r\in\mathbb{R} : \mathbb{P}_X(a^\top f(X)+b\ge r)\ge 1-\epsilon\}. \tag{3}$$

Intuitively, the condition $\mathbb{P}_X(a^\top f(X)+b\ge r)\ge 1-\epsilon$ states that the random output $Y=f(X)$ has safety level at least r, with probability at least $1-\epsilon$. The probabilistic robustness level of the network is the largest such number r. We remark that (3) is precisely the notion of probabilistic robustness used in [18]. However, [18] only provides a method for certifying that $\bar{r}(\epsilon)\ge 0$, making no effort to localize the random output $Y=f(X)$ in the output space.

In this paper, we aim to localize the neural network output while simultaneously certifying its safety. Mathematically, this amounts to estimating $\mathcal{Y}$ as well as lower bounding $\bar{r}(\epsilon)$. However, as written, these two notions are seemingly disjoint, as the probabilistic robustness level $\bar{r}(\epsilon)$ encodes no information about where in the output space the random output can reach, and the output set $\mathcal{Y}$ cannot be tractably used to ascertain robustness information due to its nonconvexity. In what follows, we bridge this gap by utilizing the approximate robustness level to bound $\bar{r}(\epsilon)$ and localize the output with high probability.

III. CERTIFICATION WITH LOCALIZATION

A. Bounding the Probabilistic Robustness Level

We begin by considering the certification aspect of our problem. It can be easily verified that the probabilistic robustness level is lower bounded by the deterministic robustness level; $r^*\le\bar{r}(\epsilon)$ for all $\epsilon\in[0,1]$, and $r^*=\bar{r}(0)$. Therefore, a natural question is whether one can instead use the easier-to-compute approximate robustness level to lower bound the probabilistic robustness level. As it turns out, this is the case so long as the surrogate output set has high coverage over

[1]The assumption of a polyhedral safe set is not restrictive. For instance, the set of outputs assigned a given label by a classifier is commonly a polyhedral set. Furthermore, when assessing network robustness using an arbitrary safe set, one may always instead use a convex polyhedral inner-approximation.

$\mathcal{Y}$. Before proving this claim in Proposition 1, we formally define this notion of coverage.

Definition 1 (ϵ-cover)**.** Let $\hat{\mathcal{Y}}$ be a subset of $\mathbb{R}^{n_y}$. For $\epsilon \in [0,1]$, the set $\hat{\mathcal{Y}}$ is said to be an *ϵ-cover* of $\mathcal{Y} = f(\mathcal{X})$ if

$$\mathbb{P}_X(f(X) \in \hat{\mathcal{Y}}) \geq 1 - \epsilon.$$

For small ϵ, Definition 1 says that $\hat{\mathcal{Y}}$ is an ϵ-cover of the output set $\mathcal{Y}$ if $\hat{\mathcal{Y}}$ contains the random output $Y = f(X)$ with high probability. In particular, if we can compute an ϵ-cover of $\mathcal{Y}$, then we will have probabilistically localized the output. By restricting the surrogate output set in (2) to be an ϵ-cover of $\mathcal{Y}$, we guarantee that the approximate robustness level takes into account the safety of Y with high probability. In this case, we suspect $\hat{r}(\hat{\mathcal{Y}})$ to well-approximate r^* in a probabilistic sense, thereby giving a lower bound on $\bar{r}(\epsilon)$. We formalize this conclusion as follows.

Proposition 1 (Lower bound from ϵ-cover)**.** *Let $\hat{\mathcal{Y}}$ be an arbitrary subset of $\mathbb{R}^{n_y}$. If $\hat{\mathcal{Y}}$ is an ϵ-cover of $\mathcal{Y} = f(\mathcal{X})$, then the approximate robustness level* (2) *lower bounds the probabilistic robustness level* (3)*, i.e.,*

$$\hat{r}(\hat{\mathcal{Y}}) \leq \bar{r}(\epsilon). \tag{4}$$

Proof. Note that $y \in \hat{\mathcal{Y}}$ implies that $a^\top y + b \geq \hat{r}(\hat{\mathcal{Y}})$ by (2). Therefore, it holds that $\mathbb{P}_X(f(X) \in \hat{\mathcal{Y}}) \leq \mathbb{P}_X(a^\top f(X) + b \geq \hat{r}(\hat{\mathcal{Y}}))$. Since $\hat{\mathcal{Y}}$ is an ϵ-cover of $\mathcal{Y}$, we have that $\mathbb{P}_X(f(X) \in \hat{\mathcal{Y}}) \geq 1 - \epsilon$. Hence,

$$1 - \epsilon \leq \mathbb{P}_X(f(X) \in \hat{\mathcal{Y}}) \leq \mathbb{P}_X(a^\top f(X) + b \geq \hat{r}(\hat{\mathcal{Y}})).$$

This shows that $\hat{r}(\hat{\mathcal{Y}})$ is feasible for the optimization (3). Therefore, $\hat{r}(\hat{\mathcal{Y}}) \leq \bar{r}(\epsilon)$, as desired. □

Proposition 1 can be interpreted as follows. Suppose that $\hat{\mathcal{Y}}$ is chosen to be an ϵ-cover of $\mathcal{Y}$ and the approximate robustness level, $\hat{r}(\hat{\mathcal{Y}})$, is computed using $\hat{\mathcal{Y}}$ as the surrogate output set. Then with high probability, the random output $Y = f(X)$ of the neural network has a safety level at least $\hat{r}(\hat{\mathcal{Y}})$, and Y is contained in $\hat{\mathcal{Y}}$. In particular, if $\hat{r}(\hat{\mathcal{Y}}) \geq 0$, then the random output Y is safe with probability at least $1 - \epsilon$. The proposition thereby shows that the approximate robustness level can be used for certification and localization of the output so long as the surrogate output set is chosen appropriately.

B. Optimizing the Bound

From Proposition 1, we know that ϵ-covers constitute good choices of the surrogate output set $\hat{\mathcal{Y}}$ used to compute the approximate robustness level. This is because the random output $Y = f(X)$ of the neural network is guaranteed to have safety level at least $\hat{r}(\hat{\mathcal{Y}})$ with high probability. However, it is entirely possible that the choice of ϵ-cover results in $\hat{r}(\hat{\mathcal{Y}}) < 0$, even when the network is probabilistically robust, meaning that $\bar{r}(\epsilon) \geq 0$. In this case, the approximate robustness level fails to issue a high-probability certificate for the safety of the random output $Y = f(X)$, despite $\hat{\mathcal{Y}}$ being able to localize it.

To overcome the above problem, we turn our attention to optimizing the lower bound (4). This amounts to finding an ϵ-cover of $\mathcal{Y}$ that maximizes the approximate robustness level. Since optimizing over all possible subsets of $\mathbb{R}^{n_y}$ is generally intractable, we choose to restrict our search to sets within a class $\mathcal{H} = \{h(\theta) : \theta \in \Theta\}$ parameterized by a parameter set $\Theta \subseteq \mathbb{R}^p$ and a set-valued function $h\colon \mathbb{R}^p \to \mathcal{P}(\mathbb{R}^{n_y})$. A concrete example of one such class is given below.

Example 1 (Norm ball class)**.** Let $\|\cdot\|$ be a fixed norm on $\mathbb{R}^{n_y}$ and $\Theta = \mathbb{R}^{n_y} \times \mathbb{R}_{++}$. Defining $p = n_y + 1$, let $h\colon \mathbb{R}^p \to \mathcal{P}(\mathbb{R}^{n_y})$ be defined by $h(\bar{y}, r) = \{y \in \mathbb{R}^{n_y} : \|y - \bar{y}\| \leq r\}$. Then, Θ and h define the class of $\|\cdot\|$-norm balls:

$$\mathcal{H} = \left\{ \{y \in \mathbb{R}^{n_y} : \|y - \bar{y}\| \leq r\} : r > 0,\ \bar{y} \in \mathbb{R}^{n_y} \right\}.$$

The problem of choosing h and Θ (and therefore also $\mathcal{H}$) is discussed in detail in Section IV. By restricting our search for ϵ-covers to within the class $\mathcal{H}$, our search reduces to maximizing the approximate robustness level over the parameter set Θ. By slightly abusing notation, we denote the dependence of the approximate robustness level on the parameter θ explicitly as

$$\hat{r}(\theta) = \inf\{a^\top y + b : y \in h(\theta)\}, \tag{5}$$

and we formulate the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \hat{r}(\theta) - \lambda v(\theta) \\ \text{subject to} \quad & \mathbb{P}_X(f(X) \in h(\theta)) \geq 1 - \epsilon, \\ & \theta \in \Theta, \end{aligned} \tag{6}$$

where the optimization variable is the parameter $\theta \in \mathbb{R}^p$. Here, $\lambda \geq 0$, and $v\colon \mathbb{R}^p \to \mathbb{R}$ can be chosen to be any nonnegative convex function on Θ that increases as the volume of $h(\theta)$ increases.

The objective $\hat{r}(\theta)$ in (6) is the approximate robustness level computed using the set $h(\theta)$ as the surrogate output set. The constraint $\mathbb{P}_X(f(X) \in h(\theta)) \geq 1 - \epsilon$ enforces that we only consider parameters θ such that $h(\theta)$ is an ϵ-cover of the output set $\mathcal{Y}$. The regularization term $-\lambda v(\theta)$ penalizes the size of $h(\theta)$. This makes the set $h(\theta)$ as small as possible while maintaining its ϵ-coverage, thereby yielding the tightest high-probability localization of the output $Y = f(X)$. The regularization is done at the expense of a slightly suboptimal bound (4), and can be eliminated by setting $\lambda = 0$, if only certification is desired. On the other hand, taking $\lambda \to \infty$ amounts to putting all assessment efforts into localizing the output. This certification-localization tradeoff is experimentally explored in Section V-A.

IV. DATA-DRIVEN REFORMULATION

Even when the set $h(\theta)$ is convex for all $\theta \in \Theta$, the probabilistic constraint in (6) is in general nonconvex [20]. Constraints of this form are typically referred to as *chance constraints*, and there exist various approaches to reformulating and relaxing them into convex constraints. Since the problem at hand considers neural networks with complicated or possibly unknown models, we seek a data-driven approach

to approximately enforcing the chance constraint in (6), without losing the certification and localization properties of the solution. The *scenario approach* is a popular method within the stochastic optimization and robust control communities that replaces the chance constraint with hard constraints on a number of randomly sampled data points [20], [21], [22], [23]. As we will soon see, this sampling-based method fits nicely into the framework of our problem, and maintains a lower bound on the probabilistic robustness level with high probability, provided that a sufficiently large number of samples is used.

To implement the scenario approach, suppose that $\{x_j : j \in \{1, 2, \ldots, N\}\}$ is a set of N independent samples of X. For each input x_j, we compute its corresponding output $y_j = f(x_j)$. Then, replacing the chance constraint in (6) with N hard constraints on the samples y_j yields the following scenario optimization problem:

$$\begin{aligned} \text{maximize} \quad & \hat{r}(\theta) - \lambda v(\theta) \\ \text{subject to} \quad & y_j \in h(\theta) \text{ for all } j \in \{1, 2, \ldots, N\}, \\ & \theta \in \Theta, \end{aligned} \tag{7}$$

where the optimization variable is $\theta \in \mathbb{R}^p$. Note that solutions to (7) are random due to the random data y_j.

As mentioned in Section I-A, the scenario approach was used recently in reachable set estimation for dynamical systems [17] and in neural network robustness certification [18]. We remark that these works are special cases of our proposed problem (7). In particular, (7) recovers the optimization of [17] in the special case that $\lambda \to \infty$, $v(\theta)$ equals the volume of the set $h(\theta)$, and $\mathcal{H}$ is the norm ball class. On the other hand, [18] is recovered in the special case that $\lambda = 0$ and $\mathcal{H}$ is the class of all half-spaces in $\mathbb{R}^{n_y}$. Consequently, (7) subsumes these prior works, handling more general classes $\mathcal{H}$ and regularizations v, and providing a unified framework for simultaneous certification *and* localization of the random output $Y = f(X)$. In Section V-B, we demonstrate the necessity for the more powerful formulation (7) by giving an example where reducing to the special case of [17] causes the robustness certification to fail.

Now, although the scenario approach has successfully eliminated the chance constraint from (6), there remain two issues to consider. First, it is not immediately clear whether the scenario optimization problem is convex. In Section IV-A, we leverage results from parametric optimization to develop conditions on our choice of Θ and h to ensure that the scenario problem (7) is convex. Second, the solution of the scenario problem (7) gives a random approximation to the solution of (6), which optimizes the bound (4) on the probabilistic robustness level. In Section IV-B we develop formal guarantees showing that the solution of (7) maintains a lower bound on the probabilistic robustness level with high probability, provided that the number of samples used is sufficiently large.

A. Conditions for Convex Optimization

In this section, we consider the effect of Θ and h on lower bounding the probabilistic robustness level of the network, and on the tractability of the resulting scenario optimization (7). A key insight is this: an ϵ-cover of the output set may in general be much larger than the output set itself. This is because regions of an ϵ-cover that do not intersect with $\mathcal{Y}$ also do not count towards the coverage proportion $1 - \epsilon$. Therefore, if the class $\mathcal{H}$ from which we choose an ϵ-cover does not have high enough complexity, then the ϵ-covers within $\mathcal{H}$ may need to be exceedingly large in order to achieve ϵ-coverage. As an example, consider covering a line segment in $\mathbb{R}^2$ first with an ℓ_2-norm ball, and then, instead, with an ellipsoid. See Figure 1. Clearly, the additional complexity of the ellipsoid allows for tighter coverage of the line segment.

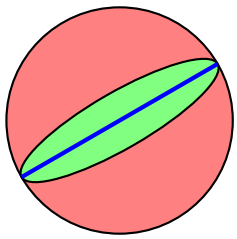

Fig. 1. Additional complexity of ellipsoid (green) compared to ℓ_2-norm ball (red) allows for tighter coverage of line segment (blue).

The problem with unnecessarily large ϵ-covers is that the feasible set in (5) includes many vectors y that may not be actual outputs in $\mathcal{Y}$. In this case, the approximate robustness level $\hat{r}(\theta)$ is small, even though the probabilistic robustness level $\bar{r}(\epsilon)$ may be high. To avoid this problem, our choice of Θ and h should ensure that the class $\mathcal{H}$ has high enough complexity. However, our choices should also yield a scenario problem (7) that is convex. Indeed, Theorem 1 gives sufficient conditions for the convexity of the scenario optimization. Before presenting these conditions, let us recall a fundamental definition for set-valued functions.

Definition 2 (Convexity of set-valued functions)**.** Consider a set-valued function $h\colon \mathbb{R}^p \to \mathcal{P}(\mathbb{R}^{n_y})$ defined on the convex set $\Theta \subseteq \mathbb{R}^p$. The function h is said to be *convex* on Θ if

$$\big(\lambda h(\theta_1) + (1-\lambda)h(\theta_2)\big) \subseteq h(\lambda\theta_1 + (1-\lambda)\theta_2)$$

for all $\theta_1, \theta_2 \in \Theta$ and all $\lambda \in [0, 1]$. The function h is said to be *concave* on Θ if

$$h(\lambda\theta_1 + (1-\lambda)\theta_2) \subseteq \big(\lambda h(\theta_1) + (1-\lambda)h(\theta_2)\big)$$

for all $\theta_1, \theta_2 \in \Theta$ and all $\lambda \in [0, 1]$. Finally, the function h on Θ is said to be *affine* if it is both convex and concave.

Remark 1. The definitions of convexity and concavity for a set-valued function appear to be opposite of those for scalar-valued and vector-valued functions. However, these definitions are consistent with those used in set-valued optimization and coincide with the traditional definition of cone-convexity. In particular, a convex cone $C \subseteq \mathbb{R}^{n_y}$ defines an order relation on $\mathcal{P}(\mathbb{R}^{n_y})$; $A, B \in \mathcal{P}(\mathbb{R}^{n_y})$ are ordered as $A \leq_C B$ if and only if $B \subseteq A + C$ [24]. Taking $C = \{0\}$ yields the familiar partial order of subset inclusion,

and Definition 2 amounts to the usual definition of cone-convexity with respect to the order $\preceq_{\{0\}}$.

Example 2 (Norm ball functions are affine)**.** Consider again the norm ball class $\mathcal{H}$ given in Example 1. It is easily verified by Definition 2 that the set-valued function h defining the class $\mathcal{H}$ is both convex and concave on $\Theta = \mathbb{R}^{n_y} \times \mathbb{R}_{++}$. Therefore, h is an affine set-valued function.

With tools for defining and proving convexity of set-valued functions now in place, we can present conditions under which the scenario optimization (7) is convex, and therefore easily solvable.

Theorem 1 (Convex scenario optimization)**.** *Consider the scenario optimization problem* (7)*. Suppose* Θ *takes the form*

$$\Theta = \{\theta \in \mathbb{R}^p : g_i(\theta) \leq 0 \text{ for all } i \in \{1, 2, \ldots, m\}\},$$

where the functions $g_i \colon \mathbb{R}^p \to \mathbb{R}$ *are convex. Furthermore, suppose* h *is a concave set-valued function that takes the form*

$$h(\theta) = \{y \in \mathbb{R}^{n_y} : h_i(y, \theta) \leq 0 \text{ for all } i \in \{1, 2, \ldots, n\}\},$$

where $h_i \colon \mathbb{R}^{n_y} \times \mathbb{R}^p \to \mathbb{R}$ *and* $h_i(y, \cdot)$ *is convex for all* $y \in \mathbb{R}^{n_y}$*. Then,* (7) *is a convex optimization problem.*

Proof. Since (7) is a maximization problem, we must show that under the assumptions on Θ and h, the objective is concave on Θ and the constraints are convex.

Let us first consider the objective $\hat{r}(\theta) - \lambda v(\theta)$, where $\hat{r}(\theta) = \inf\{a^\top y + b : y \in h(\theta)\}$. Since

1) $g(y, \theta) \coloneqq a^\top y + b$ is jointly concave on $\mathbb{R}^{n_y} \times \Theta$;
2) h is a concave set-valued function on Θ;
3) and Θ is a convex set;

Proposition 3.1 of [25] gives that $\hat{r}$ is a concave function on Θ. Since v is assumed to be convex on Θ and $\lambda \geq 0$, we conclude that the objective is concave.

Now, let us consider the constraints. The constraints $g_i(\theta) \leq 0$ are convex, so $\theta \in \Theta$ is a convex constraint. Next, the random constraint $y_j \in h(\theta)$ is equivalent to the constraint on θ that $h_i(y_j, \theta) \leq 0$ for all i. Since $h_i(y_j, \cdot)$ is a convex function, the constraint is convex. Since this holds for all $i \in \{1, 2, \ldots, n\}$ and all $j \in \{1, 2, \ldots, N\}$, we conclude that all of the constraints in (7) are convex. □

Remark 2. Theorem 1 is easily extended to include affine equality constraints in the forms taken by Θ and $h(\theta)$. Additionally, if the functions h_i in Theorem 1 are *jointly* convex, one can show that h is an affine set-valued function, and therefore $\hat{r}$ in (7) is affine (by applying Proposition 4.2 of [25]). Therefore, if v is also affine, the scenario problem (7) has an affine objective.

Theorem 1 precisely answers our earlier inquiry: the class $\mathcal{H}$ should be complex enough to contain ϵ-covers of the output set $\mathcal{Y}$ that are not unnecessarily large, but at the same time Θ should be defined by convex constraints and h should be taken as a concave set-valued function also defined by convex constraints. Note that these conditions on h are not as restrictive as they may seem. In particular, Example 2 shows for the norm ball class that h is affine (and therefore concave) and defined by convex constraints, and that this holds *for all norms on* $\mathbb{R}^{n_y}$, even though norm functions themselves are not affine. Therefore, Theorem 1 guarantees that the scenario optimization (7) using the norm ball class is a convex problem, and its objective $\hat{r}$ is affine per Remark 2. We verify this fact in the following example.

Example 3 (Scenario optimization with norm ball class)**.** Recall the norm ball class and its corresponding set-valued function defined on $\Theta = \mathbb{R}^{n_y} \times \mathbb{R}_{++}$ given by

$$h(\bar{y}, r) = \{y \in \mathbb{R}^{n_y} : \|y - \bar{y}\| \leq r\}.$$

We show that (7) using this class is convex. Indeed, the approximate robustness level is

$$\begin{aligned}\hat{r}(\bar{y}, r) &= \inf_{\|y - \bar{y}\| \leq r} a^\top y + b \\ &= b - \sup_{\|z\| \leq 1} -a^\top (rz + \bar{y}) \\ &= b + a^\top \bar{y} - r\|a\|_*,\end{aligned}$$

which is affine in the optimization variable $\theta = (\bar{y}, r)$. Hence, the scenario problem reduces to

$$\begin{aligned}\text{maximize} \quad & b + a^\top \bar{y} - r\|a\|_* - \lambda v(\bar{y}, r) \\ \text{subject to} \quad & \|y_j - \bar{y}\| \leq r \text{ for all } j \in \{1, 2, \ldots, N\}, \\ & r > 0,\end{aligned} \tag{8}$$

which is a convex problem since v is convex.

B. High-Probability Guarantees

We now turn to consider the randomness of the scenario problem's optimal value. In particular, we ask the following question: can the random scenario problem (7) be used to accurately lower bound the probabilistic robustness level and localize the random output $Y = f(X)$? In Theorem 2, we show that the answer is affirmative with high probability, provided that the problem is convex and a large enough number of samples is used.

Theorem 2 (High-probability guarantees)**.** *Let* $\epsilon, \delta \in [0, 1]$*. Assume that the scenario optimization* (7) *is convex and is attained by a solution* $\theta^* \in \mathbb{R}^p$*. If*

$$N \geq \frac{2}{\epsilon}\left(\log\frac{1}{\delta} + p\right),$$

then the following events hold with probability at least $1 - \delta$*:*

1) $h(\theta^*)$ *is an* ϵ*-cover;*
2) $\hat{r}(\theta^*) \leq \bar{r}(\epsilon)$*.*

Proof. Since the scenario problem is convex and $N \geq \frac{2}{\epsilon}\left(\log\frac{1}{\delta} + p\right)$, Theorem 1 of [22] gives that, with probability at least $1 - \delta$, we have

$$\mathbb{P}_X(f(X) \in h(\theta^*)) \geq 1 - \epsilon.$$

By Definition 1, this implies that $h(\theta^*)$ is an ϵ-cover of $\mathcal{Y}$. By Proposition 1, this further implies that $\hat{r}(\theta^*) \leq \bar{r}(\epsilon)$. □

In Theorem 2, randomness of a solution θ^* to the scenario problem (7) is taken care of by the $1-\delta$ probability bound. In particular, $h(\theta^*)$ may not actually be an ϵ-cover, albeit with probability at most δ. This added randomness is precisely the price paid for replacing the intractable chance-constrained problem (6) with the tractable scenario problem (7). However, as Theorem 2 shows, the additional randomness is not a problem, since the requirement on N scales like $\log \frac{1}{\delta}$. Therefore, we can take δ very small and still maintain a reasonable sample size N. In doing so, the scenario problem can be used in place of the chance-constrained problem to compute the maximum approximate robustness level and lower bound the probabilistic robustness level of the neural network. The resulting certification and localization hold with a probability that can be made arbitrarily close to one. For this reason, we slightly abuse terminology and call $h(\theta^*)$ in the scenario problem (7) the optimal ϵ-cover.

C. Procedural Outline

Before demonstrating our theoretical developments in Section V, we briefly recapitulate our proposed assessment method, and note the procedure's remarkable generality. The procedure amounts to three steps:

1) Choose the parameter set $\Theta \subseteq \mathbb{R}^p$ and concave set-valued function $h\colon \mathbb{R}^p \to \mathcal{P}(\mathbb{R}^{n_y})$ according to Theorem 1 with sufficiently high complexity (e.g., moderately large p).
2) Choose probability levels $\epsilon, \delta \in [0,1]$ close to zero. Independently sample $N \geq \frac{2}{\epsilon}\left(\log \frac{1}{\delta} + p\right)$ inputs x_j from the distribution $\mathbb{P}_X$ over the support $\mathcal{X}$, and then compute $y_j = f(x_j)$.
3) Choose a regularization parameter $\lambda \geq 0$ and nonnegative convex function $v\colon \mathbb{R}^p \to \mathbb{R}$. Solve the scenario optimization problem (7). Theorem 1 guarantees that the problem is convex, and Theorem 2 guarantees with probability $1-\delta$ that the solution $\hat{r}(\theta^*)$ lower bounds the probabilistic robustness level $\bar{r}(\epsilon)$ and that $h(\theta^*)$ is an ϵ-cover of $\mathcal{Y}$.

We now remark the high generality of our procedure. First, the procedure does not require knowledge of the model of the network f or its internal structures. Indeed, the only characteristics of the network that affect the above computation are the input and output dimensions, n_x and n_y. Therefore, this procedure is effectively invariant to the number and width of hidden layers, making it particularly powerful in assessing the probabilistic robustness of deep neural networks. Furthermore, the procedure makes no assumptions on the differentiability, continuity, or nonlinearity type of the network's activation functions.

Another remarkable generality of the proposed approach is that it applies to *any* input probability distribution $\mathbb{P}_X$. The support of the distribution, i.e., the input set $\mathcal{X}$, can be nonconvex, and our procedure still reduces to solving a convex optimization problem.

Finally, we remark the personalization granted to the user. Specifically, the user has the freedom to choose ϵ, δ, Θ, h, λ, and v. These choices correspond to trading off computational cost with the tightness of the high-probability guarantees and with the tightness of the resulting bound on the probabilistic robustness level. Thus, the procedure can always be tailored to the user's individual resources and desires. In particular, computational resources permitting, our data-driven approach can make the certification and localization hold with arbitrarily high probability by choosing ϵ and δ small enough. Finally, by varying λ, the user can choose the amount of importance they place on robustness certification versus on output localization. In particular, taking $\lambda = 0$ reduces to pure certification, whereas $\lambda \to \infty$ reduces to pure localization. This effect of varying λ is empirically demonstrated in Section V-A.

V. NUMERICAL EXPERIMENTS

A. Illustrative Example

Consider a $5 \times 35 \times 30 \times 2$ neural network f with ReLU activations and randomly designed weights. In our computations, we treat the weights and network structure as unknown, but assume that for $x \in \mathcal{X} \subseteq \mathbb{R}^5$ we may compute $y = f(x) \in \mathbb{R}^2$. The input X is distributed uniformly on the input set $\mathcal{X} = \{x \in \mathbb{R}^5 : \|x - \bar{x}\|_\infty \leq \epsilon_x\}$, where $\epsilon_x = 0.1$ and $\bar{x} = (1, 1, \ldots, 1) \in \mathbb{R}^5$. We consider the safe set $\mathcal{S} = \{y \in \mathbb{R}^2 : a^\top y + b \geq 0\}$, where $a \in \mathbb{R}^2$ and $b \in \mathbb{R}$ are chosen randomly for the purpose of this experiment.

We now follow our procedural outline given in Section IV-C to localize the output $Y = f(X)$ and assess its safety. We start by selecting the set $\Theta = \mathbb{R}^2 \times \mathbb{R}_{++}$ and the set-valued function $h\colon \mathbb{R}^3 \to \mathcal{P}(\mathbb{R}^2)$ defined by

$$h(\bar{y}, r) = \{y \in \mathbb{R}^2 : \|y - \bar{y}\|_Q \leq r\},$$

where $\|\cdot\|_Q$ is a norm on $\mathbb{R}^2$ defined by $\|y\|_Q = \sqrt{y^\top Q y} = \|Q^{\frac{1}{2}} y\|_2$ for a fixed symmetric positive definite matrix $Q \in \mathbb{R}^{2\times 2}$. It is easily shown that the dual norm of $\|\cdot\|_Q$ takes the form $\|y\|_{Q*} = \|y\|_{Q^{-1}} = \|Q^{-\frac{1}{2}} y\|_2$. As shown in Example 2, h is an affine set-valued function, and therefore Θ and h satisfy the conditions of Theorem 1.

The probability levels are chosen as $\epsilon = 0.1$ and $\delta = 10^{-5}$. We set $N = \left\lceil \frac{2}{\epsilon}(\log \frac{1}{\delta} + p) \right\rceil = 291$, then uniformly sample N inputs x_j from $\mathcal{X}$ and compute their corresponding random outputs y_j. We compute the (symmetric positive definite) sample covariance matrix $\Sigma \in \mathbb{R}^{2\times 2}$ of the data $\{y_j\}_{j=1}^N$ and use it to define $\|\cdot\|_Q$. Namely, we set $Q = \Sigma^{-1}$. By doing so, we take our class $\mathcal{H}$ to be the set of ellipsoids with axes scaled and oriented according to the principal components of the sampled output data.

As shown in Example 3, the scenario problem of interest takes the form

$$\begin{aligned} \text{maximize} \quad & b + a^\top \bar{y} - r\|a\|_{Q*} - \lambda v(\bar{y}, r) \\ \text{subject to} \quad & \|y_j - \bar{y}\|_Q \leq r \text{ for all } j \in \{1, 2, \ldots, N\}, \\ & r > 0, \end{aligned}$$

where the optimization variable is $\theta = (\bar{y}, r) \in \mathbb{R}^3$. We choose the regularizer to be the square of the norm ball radius, i.e., $v(\bar{y}, r) = r^2$. The optimization problem is convex as guaranteed by Theorem 1.

We solve the scenario problem first without regularization, and then with two different levels of reguarization: $\lambda_1 = 0.0001$ and $\lambda_2 = 1$. The respective solutions are denoted by θ^*, $\theta^*_{\lambda_1}$, and $\theta^*_{\lambda_2}$. Each instance takes approximately 15 seconds to solve using CVX in MATLAB on a standard laptop with a 2.9 GHz quad-core i7 processor. The resulting approximate robustness levels are $\hat{r}(\theta^*) = 42.7190$, $\hat{r}(\theta^*_{\lambda_1}) = 42.7101$, and $\hat{r}(\theta^*_{\lambda_2}) = 42.4788$. In each instance, Theorem 2 guarantees that the probabilistic robustness level $\bar{r}(0.1)$ is at least 42 with probability at least 0.99999. In other words, the random output $Y = f(X)$ has a safety level of 42 with high probability, showing that the neural network is probabilistically robust.

The optimal ϵ-covers, $h(\theta^*)$, $h(\theta^*_{\lambda_1})$, and $h(\theta^*_{\lambda_2})$, contain $Y = f(X)$ with probability at least 0.9 (disregarding $1-\delta = 0.99999 \approx 1$), and are shown in Figures 2 and 3. The set $h(\theta^*)$ is massively over-conservative due to the choice $\lambda = 0$, which corresponds to pure robustness certification. In the cases of $\lambda = \lambda_1$ and $\lambda = \lambda_2$, the optimal ϵ-covers give much tighter localizations of the output $Y = f(X)$. The approximate robustness levels with regularization are only slightly lower than the unregularized value. Yet, the most regularized ϵ-cover, $h(\theta^*_{\lambda_2})$, clearly provides much tighter approximation to $\mathcal{Y} = f(\mathcal{X})$, and still guarantees with high probability that $Y \in h(\theta^*_{\lambda_2})$. Despite the clear success of regularization in this example, it is important to remark that when the norm ball is not chosen to align with the data, the effect of regularization on the approximate robustness level can be more dramatic, and may cause the approximate robustness level to be negative even when the unregularized value is nonnegative.

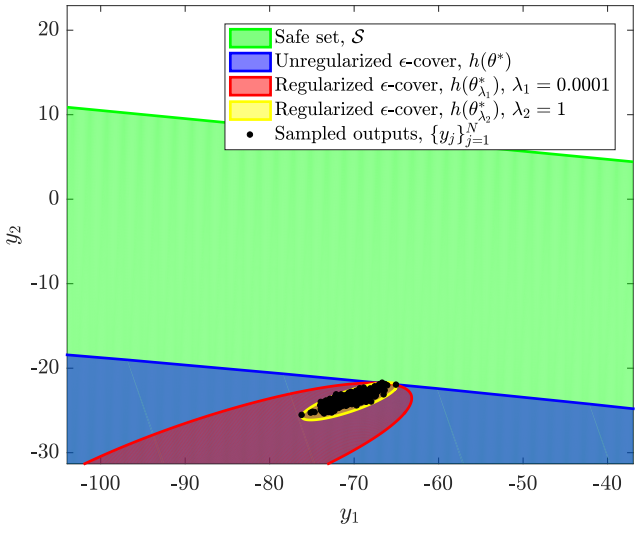

Fig. 2. Optimal ϵ-covers amongst $\|\cdot\|_Q$-norm balls. The sets $h(\theta^*)$, $h(\theta^*_{\lambda_1})$, and $h(\theta^*_{\lambda_2})$ all issue high-probability certificates of robustness for the neural network since they are contained in the safe set $\mathcal{S}$. This is even true for the unregularized $h(\theta^*)$, despite its poor localization.

B. Comparison to Output Set Estimation

In this example, we compare our proposed assessment method to an alternate approach. In the second approach, we first estimate the output set of the neural network using the scenario-based reachability analysis in [17]. We then use the resulting output set estimate to assess the robustness of the network. Recall that our proposed scenario optimization (7) generalizes the reachability analysis of [17]. In addition to localizing the network outputs, our approach directly takes the goal of robustness certification into account, whereas the estimation technique of [17] does not.

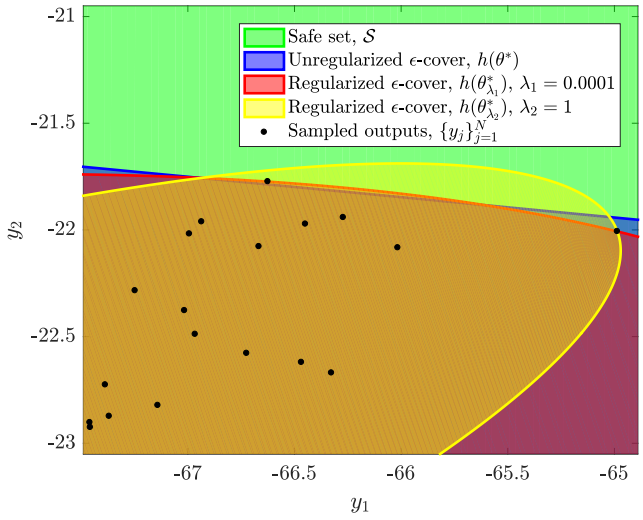

Fig. 3. Close-up view of the optimal ϵ-covers. Increased regularization makes the ϵ-cover fit the output set tighter, at the expense of losing some robustness margin. Although they are smaller, the regularized sets have extra area that lies closer to the boundary of the safe set $\mathcal{S}$ towards the top-right of the figure.

To illustrate our comparison, consider a simple ReLU neural network given by $f\colon \mathbb{R}^2 \to \mathbb{R}^2$, where $f_i(x) = \max\{0, x_i\}$ for $i \in \{1,2\}$. The input X is distributed uniformly on the input set $\mathcal{X} = \{x \in \mathbb{R}^2 : \|x - \bar{x}\|_1 \leq 1\}$, where $\bar{x} = (1,0)$. The safe set is given as $\mathcal{S} = \{y \in \mathbb{R}^2 : a^\top y + b \geq 0\}$, where $a = (0,1)$ and $b = 0.5$. It is straightforward to show that the output set is the top-half of the input set, namely, $\mathcal{Y} = \mathcal{X} \cap \{y \in \mathbb{R}^2 : y_2 \geq 0\}$. Hence, if $y \in \mathcal{Y}$ then $a^\top y + b = y_2 + b \geq b \geq 0$. Therefore, $\mathcal{Y} \subseteq \mathcal{S}$, and so the random output $Y = f(X)$ is safe with probability one. The network is deterministically robust (and therefore has nonnegative probabilistic robustness level as well).

We now perform the two assessments at hand, computing our proposed solution first. We choose the ℓ_2-norm ball class for our candidate ϵ-covers and draw sufficiently many output samples $\{y_j\}_{j=1}^N$ according to Theorem 2 with $\epsilon = 0.1$ and $\delta = 10^{-5}$. Next, we choose the regularizer $v(\bar{y}, r) = r^2$ and regularization parameter $\lambda = 0.1$, and then solve our proposed scenario problem (8) for the ℓ_2-norm ball class. The solution correctly certifies that network outputs are safe with high probability; see the blue set in Figure 4.

We now turn to the alternative method using the reachability analysis proposed in [17]. We use the same ℓ_2-norm ball class as above and solve for the minimum volume ϵ-cover using the same N sampled outputs. The estimated output set is shown in red in Figure 4. Despite being a tighter localization,

a substantial portion of the estimated output set exits the safe set, meaning this approach cannot certify the robustness of the network, even though the random output is truly safe with probability one. This comparison illustrates the fundamental difference between the problems of output set estimation and robustness certification. In particular, a good estimate of the output set of the network may not be the most informative set to use for robustness certification. This observation endorses our proposed method, which simultaneously encodes both goals of certification and localization.

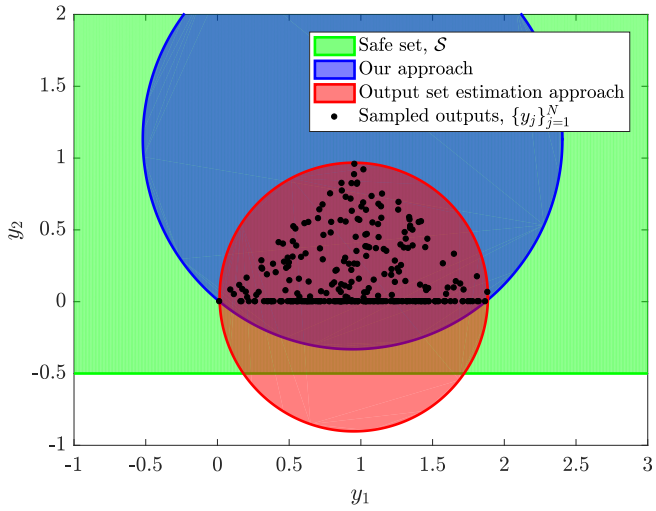

Fig. 4. The red set is the tightest ϵ-cover of the output set from the class $\mathcal{H}$, but it does not correctly certify the true robustness of the network. On the other hand, the ϵ-cover computed using our approach, shown in blue, correctly certifies the robustness of the network while maintaining reasonably tight localization of the output.

VI. CONCLUSIONS

In this paper, we propose a data-driven method for assessing the robustness of a general deep neural network to an input with random uncertainty. We introduce an intuitive notion of probabilistic robustness based on the safety level of the random output, and we relate this to the more common definition of deterministic robustness. We show that by approximating the deterministic robustness level using ϵ-covers of the output set, the probabilistic robustness level can be lower bounded while simultaneously localizing the output. We provide conditions to ensure that optimizing the lower bound amounts to a tractable convex optimization problem. The optimization's solution issues formal guarantees on the safety and localization of the random output that can be made to hold with overwhelming probability.

REFERENCES

[1] 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.

[2] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "Robustness of classifiers: from adversarial to random noise," in *Advances in Neural Information Processing Systems*, 2016, pp. 1632–1640.

[3] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[5] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 129–137.

[6] W. Xiang, H.-D. Tran, and T. T. Johnson, "Output reachable set estimation and verification for multilayer neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 11, pp. 5777–5783, 2018.

[7] 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.

[8] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, "Output range analysis for deep feedforward neural networks," in *NASA Formal Methods Symposium*. Springer, 2018, pp. 121–138.

[9] T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhillon, and L. Daniel, "Towards fast computation of certified robustness for relu networks," *arXiv preprint arXiv:1804.09699*, 2018.

[10] 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.

[11] B. G. Anderson, Z. Ma, J. Li, and S. Sojoudi, "Tightened convex relaxations for neural network robustness certification," *59th IEEE Conference on Decision and Control*, 2020, to appear, arXiv preprint arXiv:2004.00570.

[12] S. Webb, T. Rainforth, Y. W. Teh, and M. P. Kumar, "A statistical approach to assessing neural network robustness," *arXiv preprint arXiv:1811.07209*, 2018.

[13] T.-W. Weng, P.-Y. Chen, L. M. Nguyen, M. S. Squillante, I. Oseledets, and L. Daniel, "Proven: Certifying robustness of neural networks with a probabilistic approach," *arXiv preprint arXiv:1812.08329*, 2018.

[14] R. Mangal, A. V. Nori, and A. Orso, "Robustness of neural networks: a probabilistic and practical approach," in *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*. IEEE, 2019, pp. 93–96.

[15] M. Fazlyab, M. Morari, and G. J. Pappas, "Probabilistic verification and reachability analysis of neural networks via semidefinite programming," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 2726–2731.

[16] K. Dvijotham, M. Garnelo, A. Fawzi, and P. Kohli, "Verification of deep probabilistic models," *arXiv preprint arXiv:1812.02795*, 2018.

[17] A. Devonport and M. Arcak, "Estimating reachable sets with scenario optimization," *Proceedings of Machine Learning Research*, 2020.

[18] B. G. Anderson and S. Sojoudi, "Certifying neural network robustness to random input noise from samples," *preprint*, 2020. [Online]. Available: http://eecs.berkeley.edu/~sojoudi/certify_neural_net_random_2020.pdf

[19] E. Wong and J. Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," *arXiv preprint arXiv:1711.00851*, 2017.

[20] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007.

[21] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer Science & Business Media, 2012.

[22] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149–157, 2009.

[23] J. Luedtke and S. Ahmed, "A sample approximation approach for optimization with probabilistic constraints," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 674–699, 2008.

[24] A. H. Hamel, F. Heyde, A. Löhne, B. Rudloff, and C. Schrage, "Set optimization and applications–the state of the art," *Springer Proc. Math. Stat*, vol. 151, 2015.

[25] A. V. Fiacco and J. Kyparisis, "Convexity and concavity properties of the optimal value function in parametric nonlinear programming," *Journal of optimization theory and applications*, vol. 48, no. 1, pp. 95–126, 1986.