Certified Adversarial Robustness via Randomized Smoothing

Jeremy Cohen 1  Elan Rosenfeld 1  J. Zico Kolter 1 2

Abstract

Recent work has shown that any classifier which classifies well under Gaussian noise can be leveraged to create a new classifier that is provably robust to adversarial perturbations in $\ell_2$ norm. However, existing guarantees for such classifiers are suboptimal. In this work we provide the first tight analysis of this “randomized smoothing” technique. We then demonstrate that this extremely simple method outperforms by a wide margin all other provably $\ell_2$-robust classifiers proposed in the literature. Furthermore, we train an ImageNet classifier with e.g. a provable top-1 accuracy of 49% under adversarial perturbations with $\ell_2$ norm less than 0.5 (=127/255). No other provable adversarial defense has been shown to be feasible on ImageNet. While randomized smoothing with Gaussian noise only confers robustness in $\ell_2$ norm, the empirical success of the approach suggests that provable methods based on randomization at test time are a promising direction for future research into adversarially robust classification.

Code and trained models are available at http://github.com/locuslab/smoothing.

1. Introduction

Modern image classifiers achieve high accuracy on i.i.d. test sets, but are not robust to small, adversarially-chosen perturbations of their inputs (Szegedy et al., 2014; Biggio et al., 2013). Given an image $x$ correctly classified by, say, a neural network, an adversary can usually engineer an adversarial perturbation $\delta$ so small that $x + \delta$ looks just like $x$ to the human eye, yet the network classifies $x + \delta$ as a different, incorrect class. Many works have proposed heuristic methods for training classifiers intended to be robust to adversarial perturbations. However, most of these heuristics were subsequently shown to fail against suitably powerful adversaries (Carlini & Wagner, 2017; Athalye et al., 2018; Uesato et al., 2018). In response, a line of work on certifi-

able robustness has studied classifiers whose prediction at any point $x$ can be verified to be constant within some set around $x$ (Wong & Kolter, 2018; Raghunathan et al., 2018a, e.g.). In most of these works, the robust classifier takes the form of a neural network. Unfortunately, no existing algorithms for certifying the robustness of neural networks can scale to networks that are large and expressive enough to solve modern, challenging problems like ImageNet.

One workaround is to look for robust classifiers that are not neural networks. Recently, two papers (Lecuyer et al., 2019; Li et al., 2018) showed that an operation we call randomized smoothing\footnote{We adopt this term because it has been used to describe a similar technique in a different context (Duchi et al., 2012).} can transform an arbitrary classifier $f$ (in practice, a neural network) into a new “smoothed” classifier $g$ that is certifiably robust in $\ell_2$ norm. Let $f$ be an arbitrary base classifier which maps inputs $\mathbb{R}^d$ to classes $\mathcal{Y}$. For any input $x$, the smoothed classifier’s prediction $g(x)$ is defined to be the class $c$ whose decision region $\{x' \in \mathbb{R}^d : f(x') = c\}$ has the highest probability under the distribution $\mathcal{N}(x, \sigma I)$ (Figure 1).

If the base classifier $f$ is likely to return $x$’s correct class when fed noisy corruptions of $x$, then the smoothed classifier...
g will be correct at \( x \). But the smoothed classifier \( g \) will also possess a desirable property that the base classifier may lack: for any any input \( x \), the user of the classifier can verify that \( g \)'s prediction is stable within an \( \ell_2 \) ball around \( x \), simply by estimating the probability of each class’s decision region under the distribution \( \mathcal{N}(x, \sigma^2 I) \). The intuition behind this bound is that if \( \| \delta \| \) is small, the probability measure of each decision region cannot differ too much between the Gaussian distributions \( \mathcal{N}(x, \sigma^2 I) \) and \( \mathcal{N}(x + \delta, \sigma^2 I) \).

In this paper, we improve substantially on the previous analyses of randomized smoothing (Lecuyer et al., 2019; Li et al., 2018). The bounds provided in both previous analyses are unnecessarily loose, in the sense that the smoothed classifier \( g \) is provably always more robust than the bound indicates. In contrast, our bound is tight when the base classifier is linear.

Randomized smoothing has several drawbacks. It is not possible to exactly compute the distribution of a neural network’s output when the network’s input is corrupted by random Gaussian noise. Therefore, if \( f \) is a neural network it is not possible to exactly evaluate the smoothed classifier \( g \) or to compute the radius in which \( g \) is robust. Instead, we present Monte Carlo algorithms for both tasks that are successful as an adversarial defense because it reduces the unsolved problem of adversarially robust classification to the (comparatively) solved problem of standard supervised learning under noise.

To illustrate these points, we construct an ImageNet classifier that achieves e.g. 49% provable top-1 accuracy under adversarial perturbations with \( \ell_2 \) norm less than 127/255 (Table 1). No other certifiably robust classifiers have been demonstrated to be feasible on ImageNet. On datasets of smaller scale like CIFAR-10 and SVHN where competing approaches are viable, we show that randomized smoothing outperforms all competitors by a large margin.

2. Related Work

Many works have proposed classifiers intended to be robust to adversarial perturbations. These approaches can be broadly divided into empirical defenses, which empirically seem robust to known adversarial attacks, and certified defenses, which are provably robust to certain kinds of adversarial perturbations.

**Empirical defenses** The most successful empirical defense to date is adversarial training (Kurakin et al., 2017; Madry et al., 2018), in which a neural network is trained to optimize the worst-case loss over balls around the training data. Unfortunately, it is typically impossible to tell whether a prediction by an empirically robust classifier is truly robust to adversarial perturbations; the most that can be said is that a specific attack was unable to find any. In fact, most heuristic defenses proposed in the literature were later “broken” by stronger adversaries (Carlini & Wagner, 2017; Athalye et al., 2018; Uesato et al., 2018; Athalye & Carlini, 2018). Aiming to escape this cat-and-mouse game, a growing body of work has focused on defenses with formal guarantees.

**Certified defenses** A classifier is said to be certifiably robust if for any input \( x \), the user of the classifier can easily obtain a guarantee that the classifier’s prediction is constant within some set around \( x \), often an \( \ell_2 \) or \( \ell_\infty \) ball. In most

| \( \ell_2 \) RADIUS | BEST \( \sigma \) | CERTIFIED ACCURACY (%) |
|---------------------|-----------------|------------------------|
| 0.5                 | 0.25            | 49                     |
| 1.0                 | 0.50            | 37                     |
| 2.0                 | 0.50            | 19                     |
| 3.0                 | 1.00            | 12                     |

Table 1. Each row is the approximate certified top-1 accuracy of our best ImageNet classifier at some radius. For each radius, we show the best hyperparameter \( \sigma \) and the certified accuracy of the corresponding smoothed classifier. To give a sense of scale, a perturbation with \( \ell_2 \) radius 1.0 could change one pixel by 255, ten pixels by 80, 100 pixels by 25, or 1000 pixels by 8. Random guessing on ImageNet would attain 0.1% accuracy.
work in this area, the certifiably robust classifier is a neural network. Some works propose algorithms for certifying the robustness of generically trained networks, while others propose both a robust training method and a complementary certification mechanism. The latter approach has been found more effective (Wong & Kolter, 2018; Raghunathan et al., 2018a).

Certification methods are either exact (a.k.a “complete”) or conservative (a.k.a. “sound but incomplete”). In the context of $\ell_p$ norm-bounded perturbations, exact methods take a classifier $g$ and an input $x$, and return the perturbation $\delta$ with minimal norm such that $g(x) \neq g(x + \delta)$. In contrast, conservative methods return a (potentially loose) lower bound on the norm of all class changing perturbations. Exact methods are usually based on Satisfiability Modulo Theories (Katz et al., 2017; Carlini et al., 2017; Ehlers, 2017; Huang et al., 2017) or mixed integer linear programming (Cheng et al., 2017; Lomuscio & Maganti, 2017; Dutta et al., 2017; Fischetti & Jo, 2018; Bunel et al., 2018). Unfortunately, no exact methods have been shown to scale beyond moderately-sized (100,000 activations) networks (Tjeng et al., 2019), and networks of that size can only be verified when they are trained in a manner that impairs their expressivity.

Conservative certification is more scalable. Some conservative methods bound the global Lipschitz constant of the neural network (Gouk et al., 2018; Tsuzuku et al., 2018; Anil et al., 2019; Cisse et al., 2017), but these approaches tend to be very loose on expressive networks. Others measure the local smoothness of the network in the vicinity of a particular input $x$. In theory, one could obtain a robustness guarantee via an upper bound on the local Lipschitz constant of the network (Hein & Andriushchenko, 2017), but computing this quantity is intractable for general neural networks. Instead, a panoply of practical solutions have been proposed in the literature (Wong & Kolter, 2018; Wang et al., 2018a;b; Raghunathan et al., 2018a;b; Wong et al., 2018; Dvijotham et al., 2018b;a; Croce et al., 2018; Gehr et al., 2018; Mirman et al., 2018; Singh et al., 2018; Gowal et al., 2018; Weng et al., 2018a; Zhang et al., 2018). Two themes stand out. Some approaches cast verification as an optimization problem and import tools such as relaxation and duality from the optimization literature to provide conservative guarantees (Wong & Kolter, 2018; Wong et al., 2018; Raghunathan et al., 2018a;b; Dvijotham et al., 2018b;a). Others step through the network layer by layer, maintaining at each layer an outer approximation of the set of activations reachable by a perturbed input (Mirman et al., 2018; Singh et al., 2018; Gowal et al., 2018; Weng et al., 2018a; Zhang et al., 2018). None of these local certification methods have been shown to be feasible on networks that are large and expressive enough to solve modern machine learning problems like the ImageNet visual recognition task. Also, all either assume specific network architectures (e.g. ReLU activations or a layered feedforward structure) or require extensive customization for new network architectures.

**Related work involving noise** Prior works have proposed using a network’s robustness to Gaussian noise as a proxy for its robustness to adversarial perturbations (Weng et al., 2018b; Ford et al., 2019), and have suggested that Gaussian data augmentation could supplement or replace adversarial training (Zantedeschi et al., 2017; Kannan et al., 2018). Smilkov et al. (2017) observed that averaging a classifier’s input gradients over Gaussian corruptions of an image yields very interpretable saliency maps. The robustness of neural networks to random noise has been analyzed both theoretically (Fawzi et al., 2016; Franceschi et al., 2018) and empirically (Dodge & Karam, 2017). Finally, Webb et al. (2019) proposed a statistical technique for estimating the noise robustness of a classifier more efficiently than naive Monte Carlo simulation; we did not use this technique since it appears to lack formal high-probability guarantees. While these works hypothesized relationships between a neural network’s robustness to random noise and the network’s robustness to adversarial perturbations, randomized smoothing instead uses a classifier’s robustness to random noise to create a new classifier robust to adversarial perturbations.

**Randomized smoothing** In contrast to other certified defenses, a randomized smoothing classifier is not itself a neural network, though it leverages a neural network’s ability to recognize images. Several works (Liu et al., 2018; Cao & Gong, 2017) proposed randomized smoothing as a heuristic defense, but did not prove any guarantees. The first work to prove that randomized smoothing yields a certifiably robust classifier was Lecuyer et al. (2019), which derived a robustness guarantee using inequalities borrowed from the differential privacy literature. Subsequently, Li et al. (2018) gave a stronger robustness guarantee using tools from information theory. Both of these robustness guarantees are unnecessarily loose.

**Our contribution** This paper’s main contribution is a tight robustness guarantee for randomized smoothing. We then conduct the first experiments comparing randomized smoothing to other certifiably robust classifiers for $\ell_2$ norm that have been proposed in the literature (Wong et al., 2018; Tsuzuku et al., 2018; Zhang et al., 2018), and find that randomized smoothing outperforms all of them by a wide margin. Recently, we used randomized smoothing to construct an ImageNet classifier that is provably robust...
with high accuracy under $\ell_2$ bounded attacks of non-trivial norm (Table 1). Code and trained models are available at http://github.com/locuslab/smoothing.

3. Randomized smoothing

Consider a classification problem from $\mathbb{R}^d$ to classes $\mathcal{Y}$. As discussed above, randomized smoothing is a method for constructing a new, “smoothed” classifier $g$ from an arbitrary base classifier $f$. When queried at $x$, the smoothed classifier $g$ returns whichever class the base classifier $f$ is most likely to return when $x$ is perturbed by isotropic Gaussian noise:

$$g(x) = \arg \max_{c \in \mathcal{Y}} \mathbb{P}(f(x + \epsilon) = c)$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$

An equivalent definition is that $g(x)$ returns the class $c$ whose decision region $\{x' \in \mathbb{R}^d : f(x') = c\}$ has largest measure under the distribution $\mathcal{N}(x, \sigma^2 I)$.

The noise level $\sigma^2$ is a hyperparameter of the smoothed classifier $g$ which controls a robustness/accuracy tradeoff. It does not change with the input $x$.

We will first present our robustness guarantee for the smoothed classifier $g$. Then, since it is not possible to exactly evaluate the prediction of $g$ at $x$ or to certify the robustness of $g$ around $x$, we will give Monte Carlo algorithms for both tasks that succeed with arbitrarily high probability.

3.1. Robustness guarantee

Suppose that when the base classifier $f$ classifies $\mathcal{N}(x, \sigma^2 I)$, the class $c_A$ is returned with probability $p_A$, and the “runner-up” class $c_B$ is returned with probability $p_B$. Our main result is that smoothed classifier $g$ is robust around $x$ within the radius $R = \frac{\sigma}{2} (\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$, where $\Phi^{-1}$ is the inverse of the standard Gaussian CDF. This result also holds if we replace $p_A$ with a lower bound $\underline{p}_A$ and we replace $p_B$ with an upper bound $\overline{p}_B$.

**Theorem 1 (main bound).** Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Let $g$ be defined as in (1). Suppose $\underline{p}_A, \overline{p}_B \in [0, 1]$ are such that:

$$\mathbb{P}(f(x + \epsilon) = c_A) \geq \underline{p}_A \geq \overline{p}_B \geq \max_{c \neq c_A} \mathbb{P}(f(x + \epsilon) = c)$$

Then $g(x) = g(x + \delta)$ whenever $\|\delta\|_2 < R$, where

$$R = \frac{\sigma}{2} (\Phi^{-1}(p_A) - \Phi^{-1}(\overline{p}_B))$$

Moreover, the bound (3) is tight in the sense that any bound that depends only on $\underline{p}_A$ and $\overline{p}_B$ and makes no additional assumptions on $f$ cannot possibly certify a larger radius. We formalize this claim as follows:

**Theorem 2.** For any perturbation $\delta$ with $\|\delta\|_2 > R$, there exists a (linear) classifier $f$ consistent with (2) for which $g(x) \neq g(x + \delta)$.

We now make several observations about Theorem 1:

- Theorem 1 assumes nothing about $f$ (except measurability). This is crucial since it is unclear which well-behavedness assumptions, if any, are satisfied by modern deep architectures.

The certified radius $R$ is large when: (1) the noise level $\sigma$ is high, (2) the probability of the top class $c_A$ is high, and (3) the probability of each other class is low.

- The certified radius $R$ goes to infinity as $\underline{p}_A \to 1$ and $\overline{p}_B \to 0$. This should sound reasonable: the Gaussian distribution is supported on all of $\mathbb{R}^d$, so the only way that $f(x + \epsilon) = c_A$ with probability 1 is if $f(x) = c_A$ almost everywhere.

The complete proofs of Theorems 1 and 2 are in Appendix A. We now sketch the proof of the special case of Theorem 1 when there are only two classes. In this case, $p_B = 1 - p_A$, and therefore we may take $R = \sigma \Phi^{-1}(p_A)$.

**Proof sketch.** Let $c_A = g(x)$ and consider a fixed perturbation $\delta \in \mathbb{R}^d$. To guarantee that $g(x + \delta)$ is also $c_A$, we need to show that $f$ classifies the translated Gaussian $\mathcal{N}(x + \delta, \sigma^2 I)$ as $c_A$ with probability $> \frac{1}{2}$.

However, all we know about $f$ is that $f$ classifies $\mathcal{N}(x, \sigma^2 I)$ as $c_A$ with probability $\geq \underline{p}_A$. This raises the question: out of all possible base classifiers $f$ which classify $\mathcal{N}(x, \sigma^2 I)$ as $c_A$ with probability $\geq \underline{p}_A$, which one $f^*$ classifies $\mathcal{N}(x + \delta, \sigma^2 I)$ as $c_A$ with the smallest probability? One can show using an argument similar to the Neyman-Pearson lemma from statistics (Neyman & Pearson, 1933) that this “worst-case” $f^*$ is a linear classifier (Figure 3):

$$f^*(x') = \begin{cases} c_A & \text{if } \sigma^T (x' - x) \leq \sigma \|\delta\| \Phi^{-1}(p_A) \\ c_B & \text{otherwise} \end{cases}$$

This “worst-case” $f^*$ classifies $\mathcal{N}(x + \delta, \sigma^2 I)$ as $c_A$ with probability

$$\Phi \left( \Phi^{-1}(p_A) - \frac{\|\delta\|}{\sigma} \right) > \frac{1}{2}$$

which rearranges to $\|\delta\| < \sigma \Phi^{-1}(p_A)$.

Theorem 2 is a simple consequence: for any $\delta$ with $\|\delta\| > R$, there exists a base classifier $f^*$, consistent with (2) of the form (4), yet if $f^*$ is the base classifier, $g(x) \neq g(x + \delta)$.
Certified Adversarial Robustness via Randomized Smoothing

Figure 3. Illustration of the “worst-case” base classifier. Consider the set on which \( f \) returns \( c_A \). All we know is that this set has probability \( \geq p_A \) under the distribution \( \mathcal{N}(x, \sigma^2 I) \). Out of all such sets — two of which are depicted above — which one has the smallest probability under the distribution \( \mathcal{N}(x + \delta, \sigma^2 I) \)? We prove that this extremal set is the set depicted on the right — the half-space with normal vector \( \delta \) which has probability \( p_A \) under the distribution \( \mathcal{N}(x, \sigma^2 I) \).

Linear base classifier A binary linear classifier \( f(x) = \text{sign}(w^T x + b) \) is already certifiably robust: given an input \( x \), the distance to the decision boundary is \( |w^T x + b|/\|w\| \), and no perturbation \( \delta \) with norm less than this distance can possibly change \( f \)’s prediction. In Appendix B we show that if \( f \) is binary linear, then \( g(x) = f(x) \) everywhere. Moreover, we show that our bound (3) will certify the true robust radius \( |w^T x + b|/\|w\| \), rather than a smaller, overconservative radius. In a sense, linear classifiers are a “fixed point” of randomized smoothing: smoothing a binary linear base classifier yields the same classifier with the same robustness guarantee.

Comparison to prior bounds Both Lecuyer et al. (2019) and Li et al. (2018) prove a bound for the same setting as Theorem 1 but with a different expression than (3) for the certified radius \( R \). Both bounds are loose, and both come with a tuning parameter that needs to be optimized. In Appendix F, we derive the other two bounds using this paper’s notation. Here we simply state them.

The bound from Lecuyer et al. (2019) is

\[
R = \sup_{0 < \beta \leq \frac{1}{2}} \log \frac{p_A}{\bar{p}_B} \left( \frac{\sigma \beta}{2 \log \left( \frac{1.25(1+\exp(\beta))}{p_A-\exp(2\beta)\bar{p}_B} \right)} \right)
\]

The bound from Li et al. (2018) is

\[
R = \sup_{\alpha > 0} \sigma \left( \frac{-2 \log \left( 1 - p_A - \frac{p_A^\alpha}{\bar{p}_B} + 2 \left( \frac{1}{2} \left( p_A^{1-\alpha} + \bar{p}_B^{1-\alpha} \right) \right)^{\frac{1}{\alpha}} \right)}{\alpha} \right)
\]

All three bounds are plotted in Figure 4 (left). Observe that

3.2. Certification in practice

To compute the certified radius of \( g \) around a point \( x \) using Theorem 1, we would ideally know \( p_A = \mathbb{P}(f(x+\epsilon) = c_A) \) and \( \bar{p}_B = \max_{c \neq c_A} \mathbb{P}(f(x+\epsilon) = c) \). However, if \( f \) is a deep neural network, it is not possible to exactly compute the distribution of \( f(x+\epsilon) \), a discrete distribution over the classes \( Y \). Therefore, we instead use Monte Carlo sampling to estimate some \( \tilde{p}_A \) and \( \tilde{p}_B \) for which \( \tilde{p}_A \leq p_A \) and \( \tilde{p}_B \geq \bar{p}_B \) with arbitrarily high probability \( 1 - \alpha \) over the samples. Suppose for simplicity that we already knew \( c_A \) and needed to obtain \( p_A \). We could collect \( n \) samples of \( f(x+\epsilon) \) and use any Bernoulli confidence interval to obtain a lower bound on \( p_A \) that holds with probability at least \( 1 - \alpha \) over the \( n \) samples. In this work, following Lecuyer et al. (2019), we use the Clopper-Pearson Bernoulli confidence interval, which inverts the binomial CDF (Clopper & Pearson, 1934).

However, estimating \( p_A \) and \( \bar{p}_B \) while simultaneously identifying the top class \( c_A \) is a little bit tricky, statistically speaking. We propose a simple two-step procedure. First, use \( n_0 \) samples from \( f(x+\epsilon) \) to take a guess \( \tilde{c}_A \) at the identity of the top class \( c_A \). In practice we observed that \( f(x+\epsilon) \) tends to put most of its weight on the top class, so \( n_0 \) can be set very small. Second, use \( n \) samples from \( f(x+\epsilon) \) to obtain some \( \tilde{p}_A \) and \( \tilde{p}_B \) for which \( \tilde{p}_A \leq p_A \) and \( \tilde{p}_B \geq \bar{p}_B \) with high probability. We observed that it is much more typical for the mass of \( f(x+\epsilon) \) not allocated to \( c_A \) to be allocated entirely to one runner-up class than to be allocated uniformly over all remaining classes. Therefore, the quantity \( 1 - p_A \) is a reasonably tight upper bound on \( \bar{p}_B \). Hence, we simply set \( \tilde{p}_B = 1 - \tilde{p}_A \), so our bound becomes \( R = \sigma \Phi^{-1}(\tilde{p}_A) \).

The full certification procedure is described in pseudocode as CERTIFY. If \( p_A < \frac{1}{2} \), we abstain from making a cer-
Pseudocode for certification and prediction

# certify the robustness of $g$ around an input $x$

function CERTIFY($f$, $\sigma^2$, $x$, $n_0$, $n$, $\alpha$)
    counts ← SAMPLE UNDER NOISE($f$, $x$, $n_0$, $\sigma^2$)
    $\hat{c}_A$ ← arg max$_k$ counts[$k$]
    counts ← SAMPLE UNDER NOISE($f$, $x$, $n$, $\sigma^2$)
    $\bar{p}$ ← counts[$\hat{c}_A$]/SUM(counts)
    $\bar{p}_A$ ← LOWER CONFIDENCE BOUND($\bar{p}$, $n$, $1 - \alpha$)
    if $\bar{p}_A > \frac{1}{2}$ return prediction $\hat{c}_A$ and radius $\sigma \Phi^{-1}(\bar{p}_A)$
    else return ABSTAIN

# evaluate $g$ at $x$

function PREDICT($f$, $\sigma^2$, $x$, $n$, $\alpha$)
    counts ← SAMPLE UNDER NOISE($f$, $x$, $n$, $\sigma^2$)
    $\hat{c}_A$, $\hat{c}_B$ ← top two indices in counts
    $n_A$, $n_B$ ← counts[$\hat{c}_A$], counts[$\hat{c}_B$]
    if BINOMIAL VALUE($n_A$, $n_A + n_B$, $0.5$) $\leq \alpha$ return $\hat{c}_A$
    else return ABSTAIN

# helper function: draw num samples from $f(x + \epsilon)$

function SAMPLE UNDER NOISE($f$, $x$, num, $\sigma^2$)
    Initialize counts ← ZEROS[$K$]
    for $i = 1$ to num do
        Sample noise: $\epsilon_i \sim N(0, \sigma^2 I)$.
        Classify: $c_i \leftarrow f(x + \epsilon_i)$
        Increment: counts[$c_i$]++
    return counts

tification; this can occur especially if $\hat{c}_A \neq g(x)$, i.e. if we misidentify the top class using the first $n_0$ samples of $f(x + \epsilon)$. On the other hand, if $p_A > \frac{1}{2}$, then $p_A \geq p_A \geq \bar{p}_B \geq \bar{p}_B$ with probability at least $1 - \alpha$ over the Monte Carlo samples, so both the class $\hat{c}_A$ and the certified radius $R$ returned by CERTIFY will be simultaneously correct with probability at least $1 - \alpha$ over the samples.

Given a test set $\{(x_1, c_1) \ldots (x_m, c_m)\}$, it is not possible to exactly compute the certified accuracy of $g$ at radius $r$, but by running CERTIFY on each example, we can construct a confidence interval for this quantity (Appendix C).

Certifying large radii requires many samples Recall from Theorem 1 that $R$ approaches $\infty$ as $p_A$ approaches 1. Unfortunately, it turns out that $p_A$ approaches 1 so slowly with $n$ that $R$ also approaches $\infty$ very slowly with $n$. Consider the most favorable situation: $f(x) = c_A$ everywhere. This means that $g$ is robust at radius $\infty$. But how rapidly does our robustness radius $R$ increase with the number of samples $n$? After observing $n$ samples of $f(x + \epsilon)$ which all equal $c_A$, the tightest possible (to our knowledge) lower bound would say that with probability at least $1 - \alpha$, $p_A \geq \alpha^{1/n}$. Plugging $p_A = \alpha^{1/n}$ and $\bar{p}_B = 1 - p_A$ into (3) yields an expression for the certified radius as a function of $n$: $R = \sigma \Phi^{-1}(\alpha^{1/n})$. This function is plotted in Figure 4 (right) for $\alpha = 0.001$, $\sigma = 1$. Observe that certifying a radius of $4\sigma$ with 99.9% confidence would require $\approx 10^{5}$ samples.

It is unclear whether there is a more sample-efficient way to estimate a high-probability lower bound on $p_A$ than our approach of treating $1[f(x + \epsilon) = c_A]$ as a Bernoulli($p_A$) random variable. If $f$ is deterministic, the problem amounts to computing a lower bound on the Gaussian measure of the decision region of class $c_A$. Specialized algorithms have been developed for estimating the Gaussian measure of convex sets (Cousins & Vempala, 2015), but the decision regions of a neural network are not in general convex.

3.3. Prediction in practice

When $f$ is a neural network, it is not possible to exactly evaluate the smoothed classifier $g$. A practical implementation would draw $n$ samples of $f(x + \epsilon)$ and return whichever class appeared most often. There is always some probability that this procedure will fail to correctly return $g(x)$, but the failure rate can be controlled by abstaining from making a prediction when it’s a close call. Though CERTIFY is a valid prediction procedure, the procedure described in pseudocode as PREDICT is less liable to abstain if the top two classes of $f(x + \epsilon)$ have similar mass. It is based on the hypothesis test given in Hung & Fithian (2017) for identifying the top category of a multinomial distribution. Let $\alpha$ be the target failure rate of the prediction procedure. Suppose that out of the $n$ samples of $f(x + \epsilon)$, the top class $\hat{c}_A$ occurred $n_A$ times and the runner-up class $\hat{c}_B$ occurred $n_B$ times. We then return $c_A$ if the two-sided Binomial hypothesis test that $n_A$ is drawn from Binomial($\frac{1}{2}$, $n_A + n_B$) returns a p-value less than $\alpha$; otherwise we abstain. The probability over the samples that this procedure will return a class $\hat{c}_A \neq g(x)$ is no greater than $\alpha$.

Adversarial vulnerability Even if the true smoothed classifier $g$ is robust at radius $R$, PREDICT will be vulnerable in a certain sense to adversarial perturbations with $\ell_2$ norm less than $R$. By engineering a perturbation $\delta$ for which $f(x + \delta + \epsilon)$ puts mass just over $\frac{1}{2}$ on class $c_A$ and mass just under $\frac{1}{2}$ on class $c_B$, an adversary can force the abstention rate of PREDICT to be arbitrarily high. (The failure rate will still remain bounded by $\alpha$.)

3.4. Training with noise

Theorem 1 holds regardless of how the base classifier $f$ is trained. Nevertheless, in order to make $p_A$ high enough that $g$ is robust at a reasonable radius, it is necessary to train the base classifier with Gaussian data augmentation at variance $\sigma^2$, as reported in Lecuyer et al. (2019). We did not experiment with stability training (Zheng et al., 2016), as
was proposed in Li et al. (2018). We found that training the base classifier with Gaussian data augmentation at variance other than $\sigma^2$ was not as effective (Appendix E.2). Finally, we note that training the base classifier with Gaussian data augmentation has the same computational cost as standard training and is much computationally cheaper than PGD adversarial training.

4. Experiments

In adversarially robust classification, one metric of interest is the certified test set accuracy at radius $r$, defined as the fraction of the test set which $g$ classifies correctly with a prediction that is certifiably robust within an $\ell_2$ ball of radius $r$. However, if $g$ is a randomized smoothing classifier, computing this quantity exactly is not possible, so we instead report the approximate certified test set accuracy, defined as the fraction of the test set which CERTIFY classifies correctly (without abstaining) and certifies robust at radius $r$. Appendix C shows how to convert the approximate certified accuracy into a high-probability lower bound on the true certified accuracy, but Appendix E.1 demonstrates that when $\alpha$ is small, the difference between these two quantities is negligible. Therefore, in our experiments we omit the step for simplicity.

We primarily ran experiments on ImageNet (Deng et al., 2009), though we compare against baselines on CIFAR-10 (Krizhevsky, 2009) and SVHN (Netzer et al., 2011). Our base classifier was a ResNet-50 (He et al., 2016) on ImageNet and a 110-layer residual network on CIFAR-10 and SVHN.

In all experiments, unless otherwise stated, we ran CERTIFY with $\alpha = 0.001$, so there was at most a 0.1% chance that CERTIFY returned a radius in which $g$ was not truly robust. Unless otherwise stated, when running CERTIFY we used $n_0 = 100$ Monte Carlo samples for selection and $n = 100,000$ samples for estimation. With this setting of $n$, on an NVIDIA RTX 2080 Ti, running CERTIFY took 16 seconds for each CIFAR-10 example and 150 seconds for each ImageNet example.

In the figures below that plot certified accuracy as a function of radius $r$, the certified accuracy always decreases gradually with $r$ until reaching some $r$ where it plummets all at once to zero. This drop occurs because for each noise level $\sigma$ and number of samples $n$, there is a hard upper limit to the radius we can certify with high probability, achieved when all $n$ samples are classified by $f$ as the same class.

Comparison to baselines We compared randomized smoothing to three recent algorithms for certifiable $\ell_2$-robust classification: (1) the Lipschitz approach from Tsuzuku et al. (2018), (2) the duality approach from Wong et al. (2018), and (3) the approach from Weng et al. (2018a); Zhang et al. (2018). We did not compare against Dvijotham et al. (2018b) since the authors did not release code and the method is very similar to Wong et al. (2018). We also did not compare against Croce et al. (2018) since that paper reports that their certified radii are smaller than those of Wong et al. (2018).

In all cases we compare against pretrained networks provided by the authors, and we preprocess the data the same way that the authors did. For this reason, the numbers in these plots are not directly comparable to each other or to the numbers elsewhere in this paper. The 110-layer residual network we used as the base classifier for randomized smoothing has much higher capacity that the networks of competing methods, since our goal was to illustrate that randomized smoothing scales to large networks.

- Figure 5a compares randomized smoothing against Tsuzuku et al. (2018) on the SVHN dataset.
- Figure 5b compares randomized smoothing against Wong et al. (2018) on the CIFAR-10 dataset. We compare against three settings of the hyperparameter of Wong et al. (2018).
- Figure 5c compares randomized smoothing against Zhang et al. (2018) on the CIFAR-10 dataset. This
The bound in Theorem 1 is tight when $f$ is linear. Since deep neural networks are not linear, we empirically assessed the tightness of our bound by subjecting an ImageNet randomized smoothing classifier ($\sigma = 0.25$) to a projected gradient descent-style adversarial attack. For each example, we ran CERTIFY with

standard accuracy.

Tables for ImageNet and CIFAR-10 are in Appendix D.

**Comparison** The comparison is not exactly fair, since Zhang et al. (2018) propose only a verification algorithm and not a training algorithm.

Observe that randomized smoothing outperforms all competing approaches in certified accuracy at all radii. In particular, we outperform the closest competitor — Wong et al. (2019) on CIFAR-10 — by a margin of 15% accuracy. Full experimental details are in Appendix G.

**Effect of $\sigma$** Figure 7 plots the certified accuracy of randomized smoothing on ImageNet for varying noise levels $\sigma$. We see that $\sigma$ controls a robustness/accuracy trade-off. When $\sigma$ is low, small radii can be certified with high accuracy, but large radii cannot be certified at all. When $\sigma$ is high, larger radii can be certified, but smaller radii are certified at a lower accuracy. This observation echoes the finding in Tsipras et al. (2019) that adversarially trained networks with higher robust accuracy tend to have lower standard accuracy.

**Prior randomized smoothing bounds** Figure 6a plots the certified accuracy obtained using our bound alongside the certified accuracy obtained using the bounds of Lecuyer et al. (2019) and Li et al. (2018). This figure is for ImageNet with $\sigma = 0.25$.

**Sensitivity to CERTIFY parameters** Figure 6b plots the certified accuracy across different numbers of Monte Carlo samples $n$. Observe that the maximum $\ell_2$ radius we can certify is roughly logarithmic in $n$. Figure 6c plots the certified accuracy across varying settings of the confidence parameter $\alpha$. Observe that the certified accuracy is not very sensitive to $\alpha$. Both figures are for ImageNet with $\sigma = 0.25$.

**Prediction** It is computationally expensive to certify the robustness of $g$ around a point $x$ using CERTIFY, since $n$ must be very large. However, it is far cheaper to evaluate $g$ at $x$ using PREDICT, since $n$ can be small. For example, when we ran PREDICT on ImageNet ($\sigma = 0.25$) using $n=100$, making each prediction only took 0.15 seconds, and we attained a top-1 test accuracy of 65% (Appendix D).

As discussed earlier, an adversary can force PREDICT to abstain with arbitrarily high probability. However, It is relatively rare for PREDICT to abstain on the actual data distribution. On ImageNet ($\sigma = 0.25$), PREDICT with failure rate $\alpha = 0.001$ abstained 12% of the time when $n = 100$, 4% of the time when $n = 1000$, and 1% of the time when $n = 10,000$.

**Empirical tightness of bound** The bound in Theorem 1 is tight when $f$ is linear. Since deep neural networks are not linear, we empirically assessed the tightness of our bound by subjecting an ImageNet randomized smoothing classifier ($\sigma = 0.25$) to a projected gradient descent-style adversarial attack. For each example, we ran CERTIFY with

Figure 6. Randomized smoothing on ImageNet with $\sigma = 0.25$.

Figure 7. Approximate certified accuracy of randomized smoothing on ImageNet with three different noise levels $\sigma$. 

(a) Our bound vs. prior bounds. (b) Vary number of samples $n$. (c) Vary confidence $1 - \alpha$. 

Figure 6a plots the certified accuracy of randomized smoothing on ImageNet with $\sigma = 0$. Observe that the maximum $\ell_2$ radius we can certify is roughly logarithmic in $n$. Figure 6c plots the certified accuracy across varying settings of the confidence parameter $\alpha$. Observe that the certified accuracy is not very sensitive to $\alpha$. Both figures are for ImageNet with $\sigma = 0.25$.

**Prior randomized smoothing bounds** Figure 6a plots the certified accuracy obtained using our bound alongside the certified accuracy obtained using the bounds of Lecuyer et al. (2019) and Li et al. (2018). This figure is for ImageNet with $\sigma = 0.25$.

**Sensitivity to CERTIFY parameters** Figure 6b plots the certified accuracy across different numbers of Monte Carlo samples $n$. Observe that the maximum $\ell_2$ radius we can certify is roughly logarithmic in $n$. Figure 6c plots the certified accuracy across varying settings of the confidence parameter $\alpha$. Observe that the certified accuracy is not very sensitive to $\alpha$. Both figures are for ImageNet with $\sigma = 0.25$.

**Prediction** It is computationally expensive to certify the robustness of $g$ around a point $x$ using CERTIFY, since $n$ must be very large. However, it is far cheaper to evaluate $g$ at $x$ using PREDICT, since $n$ can be small. For example, when we ran PREDICT on ImageNet ($\sigma = 0.25$) using $n=100$, making each prediction only took 0.15 seconds, and we attained a top-1 test accuracy of 65% (Appendix D).

As discussed earlier, an adversary can force PREDICT to abstain with arbitrarily high probability. However, It is relatively rare for PREDICT to abstain on the actual data distribution. On ImageNet ($\sigma = 0.25$), PREDICT with failure rate $\alpha = 0.001$ abstained 12% of the time when $n = 100$, 4% of the time when $n = 1000$, and 1% of the time when $n = 10,000$. 

Figure 7 plots the certified accuracy of randomized smoothing on ImageNet for varying noise levels $\sigma$. We see that $\sigma$ controls a robustness/accuracy trade-off. When $\sigma$ is low, small radii can be certified with high accuracy, but large radii cannot be certified at all. When $\sigma$ is high, larger radii can be certified, but smaller radii are certified at a lower accuracy. This observation echoes the finding in Tsipras et al. (2019) that adversarially trained networks with higher robust accuracy tend to have lower standard accuracy.

Tables for ImageNet and CIFAR-10 are in Appendix D.

**Prior randomized smoothing bounds** Figure 6a plots the certified accuracy obtained using our bound alongside the certified accuracy obtained using the bounds of Lecuyer et al. (2019) and Li et al. (2018). This figure is for ImageNet with $\sigma = 0.25$.

**Sensitivity to CERTIFY parameters** Figure 6b plots the certified accuracy across different numbers of Monte Carlo samples $n$. Observe that the maximum $\ell_2$ radius we can certify is roughly logarithmic in $n$. Figure 6c plots the certified accuracy across varying settings of the confidence parameter $\alpha$. Observe that the certified accuracy is not very sensitive to $\alpha$. Both figures are for ImageNet with $\sigma = 0.25$.

**Prediction** It is computationally expensive to certify the robustness of $g$ around a point $x$ using CERTIFY, since $n$ must be very large. However, it is far cheaper to evaluate $g$ at $x$ using PREDICT, since $n$ can be small. For example, when we ran PREDICT on ImageNet ($\sigma = 0.25$) using $n=100$, making each prediction only took 0.15 seconds, and we attained a top-1 test accuracy of 65% (Appendix D).

As discussed earlier, an adversary can force PREDICT to abstain with arbitrarily high probability. However, It is relatively rare for PREDICT to abstain on the actual data distribution. On ImageNet ($\sigma = 0.25$), PREDICT with failure rate $\alpha = 0.001$ abstained 12% of the time when $n = 100$, 4% of the time when $n = 1000$, and 1% of the time when $n = 10,000$.

**Empirical tightness of bound** The bound in Theorem 1 is tight when $f$ is linear. Since deep neural networks are not linear, we empirically assessed the tightness of our bound by subjecting an ImageNet randomized smoothing classifier ($\sigma = 0.25$) to a projected gradient descent-style adversarial attack. For each example, we ran CERTIFY with
\( \alpha = 0.01 \), and, if the example was correctly classified and certified robust at radius \( R \), we tried finding an adversarial example for \( g \) within radius \( 1.5R \) and within radius \( 2R \). We succeeded 17% of the time at radius \( 1.5R \) and 53% of the time at radius \( 2R \). Note that since \( p_A \) is a conservative lower bound on \( p_A \), the true radius in which \( g \) is robust is always larger than \( R \). See Appendix G.3 for more details on the attack.

5. Discussion

Noise can be larger in high dimension Since our expression (3) for the certified radius does not depend explicitly on the data dimension \( d \), one might worry that randomized smoothing is less effective in high dimension — certifying a fixed \( \ell_2 \) radius is “less impressive” for, say, a 224 \( \times \) 224 image than for a 112 \( \times \) 112 image. However, it turns out that in higher dimension, the hyperparameter \( \sigma \) can be increased without degrading \( g \)'s accuracy or the \( p_A \)'s, so overall the certified radii are larger. For example, it is easy to show that certifying ImageNet at resolution 224 \( \times \) 224 under Gaussian noise with standard deviation \( 2\sigma \) is no more difficult than certifying ImageNet at resolution 112 \( \times \) 112 under Gaussian noise with standard deviation \( \sigma \). To see this, consider an image at resolution 224 \( \times \) 224 corrupted by Gaussian noise at standard deviation \( 2\sigma \). Now average together every 2x2 group of four pixels. The result is an image at resolution 112 \( \times \) 112 corrupted by Gaussian noise at standard deviation \( \sigma \). This holds because the average of four independent copies of \( \mathcal{N}(0, 4\sigma^2) \) is distributed as \( \mathcal{N}(0, \sigma^2) \). In effect, any high-resolution image corrupted by large noise can be mapped via average pooling to a low-resolution image corrupted by small noise. As a result, in high dimension one can add larger noise while still retaining enough information to identify the image’s class.

This effect is made intuitively clear in Figure 8, which shows an image at high and low resolution corrupted by Gaussian noise with the same variance. The class (“hummingbird”) is easy to discern from the high-resolution noisy image, but not from the low-resolution noisy image.

Certified radii are small compared to \( \sigma \) Unfortunately, randomized smoothing requires \( f \) to classify well under Gaussian perturbations with large norm in order for \( g \) to be robust to adversarial perturbations with comparably small norm. For example, Figure 2 shows an image corrupted by Gaussian noise at \( \sigma = 0.5 \). On ImageNet a smoothed classifier with \( \sigma = 0.5 \) attains approximately 55% certified top-1 accuracy at \( \ell_2 \) radius 0.25 (=63/255), and approximately 42% certified top-1 accuracy at \( \ell_2 \) radius 0.5 (=127/255). For comparison, ImageNet has dimension \( d = 224 \times 224 \times 3 \), so the expected norm of a draw from \( \mathcal{N}(0, \sigma^2I_d) \) is \( \sqrt{2\Gamma(\frac{d+1}{2})/\Gamma(\frac{d}{2})}\sigma \approx 97 \) (Chandrasekaran et al., 2012), which is many times larger than the radii at which \( g \) is provably robust at high accuracy.

Other norms Randomized smoothing with Gaussian noise confers robustness in \( \ell_2 \) norm. It is unclear whether randomized smoothing with other noise distributions would confer robustness in other norms. The uniform and Laplace distributions do not lead to tight robustness guarantees for the \( \ell_\infty \) and \( \ell_1 \) norm balls, respectively, as one might hope.

6. Conclusion

Randomized smoothing is a method for obtaining an adversarially robust classifier \( g \) from any base classifier \( f \) that is accurate under noisy inputs. Unlike previous randomized smoothing bounds, our bound in Theorem 1 is tight and has a simple geometric intuition. We conducted the first experiments comparing randomized smoothing \( \ell_2 \)-certifiably robust classifiers and found that randomized smoothing gives higher certified accuracy at all radii than all competing methods. Moreover, randomized smoothing is the only certifiably robust classifier that has been shown feasible on ImageNet.

Regardless of whether or not imperceptible adversarial perturbations truly pose a unique real-world security threat (Gilmer et al., 2018a), they do present an interesting scientific question: under what conditions is it possible to learn a high-accuracy classifier that is invariant to imperceptible perturbations of its input? Recent theoretical work has taken steps towards addressing this question (Mahlojifar et al., 2019; Gilmer et al., 2018b; Fawzi et al., 2018; Tsipras et al., 2019; Schmidt et al., 2018; Shafahi et al., 2019; Dohma-
Certified Adversarial Robustness via Randomized Smoothing

tob, 2018; Bubeck et al., 2018b). Of particular relevance, Bubeck et al. (2018a); Degwekar & Vaikuntanathan (2019) show that for certain classification problems, there exists a robust classifier that is not efficiently computable (like randomized smoothing) but there does not exist any robust classifier that is efficiently computable. We hope that by establishing a new provably correct baseline for adversarially robust classification in \( \ell_2 \) norm, randomized smoothing can help future work disentangle what is truly impossible from what has merely eluded the field so far.

Our strong empirical results suggest that provable methods based on randomization at test time may be the most promising future direction for research into adversarially robust classification. Most empirical approaches have been “broken,” and all other provable approaches depend strongly on minor technical aspects of the network architecture, such as the choice of activation function or the presence of skip connections. This property seems philosophically at odds with deep learning, in which any given task can usually be solved by a variety of possible network architectures. In contrast, randomized smoothing makes no assumptions about the network’s architecture, and only relies on the base classifier’s ability to classify well under noise, thereby reducing adversarially robust classification to classical supervised learning.

7. Acknowledgements

We thank MathOverflow user mateusz-kwasnicki for helping us prove Lemma 4 in the appendix, Aaditya Ramdas for pointing us toward the work of Hung & Fithian (2017), and Siva Balakrishnan for helpful discussions regarding the confidence interval in Appendix C. We thank Tolani Olarinre, Adarsh Prasad, Ben Cousins, and Ramon Van Handel for useful conversations. Finally, we are very grateful to Vaishnav Nagarajan, Arun Sai Suggala, Shaojie Bai, Mikhail Khodak, Han Zhao, and Zachary Lipton for reviewing drafts of this work. Jeremy Cohen is supported by a grant from the Bosch Center for AI.

References

Anil, C., Lucas, J., and Grosse, R. B. Sorting out lipschitz function approximation. arXiv preprint arXiv:1811.03381, 2019.

Athalye, A. and Carlini, N. On the robustness of the cvpr 2018 white-box adversarial example defenses. The Bright and Dark Sides of Computer Vision: Challenges and Opportunities for Privacy and Security, 2018.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Proceedings of the 35th International Conference on Machine Learning, 2018.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., nndu, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. Joint European Conference on Machine Learning and Knowledge Discovery in Database, 2013.

Blanchard, G. Lecture Notes, 2007. URL http://www.math.uni-potsdam.de/~blanchard/lectures/lect_2.pdf.

Bubeck, S., Lee, Y. T., Price, E., and Razenshteyn, I. Adversarial examples from cryptographic pseudo-random generators, 2018a.

Bubeck, S., Price, E., and Razenshteyn, I. Adversarial examples from computational constraints. arXiv preprint arXiv:1805.10204, 2018b.

Bunel, R. R., Turkaslan, I., Torr, P., Kohli, P., and Mudigonda, P. K. A unified view of piecewise linear neural network verification. In Advances in Neural Information Processing Systems 31. 2018.

Cao, X. and Gong, N. Z. Mitigating evasion attacks to deep neural networks via region-based classification. 33rd Annual Computer Security Applications Conference, 2017.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017.

Carlini, N., Katz, G., Barrett, C., and Dill, D. L. Provably minimally-distorted adversarial examples. arXiv preprint arXiv:1709.10207, 2017.

Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. The convex geometry of linear inverse problems. Found. Comput. Math., 2012.

Cheng, C.-H., Nhenenberg, G., and Ruess, H. Maximum resilience of artificial neural networks. International Symposium on Automated Technology for Verification and Analysis, 2017.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In Proceedings of the 34th International Conference on Machine Learning, 2017.

Clopper, C. J. and Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika, 26(4):pp. 404–413, 1934. ISSN 00063444.
Cousins, B. and Vempala, S. Bypassing kls: Gaussian cooling and an o(n^3) volume algorithm. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, 2015.

Croce, F., Andriushchenko, M., and Hein, M. Provable robustness of relu networks via maximization of linear regions. *arXiv preprint arXiv:1810.07481*, 2018.

Degwekar, A. and Vaikuntanathan, V. Computational limitations in robust classification and win-win results. *arXiv preprint arXiv:1902.01086*, 2019.

Dutta, S., Jha, S., Sanakaranarayanan, S., and Tiwari, A. Output range analysis for deep neural networks. *arXiv preprint arXiv:1709.09130*, 2017.

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

Dvijotham, K., Stanforth, R., Gowal, S., Mann, T., and Kohli, P. A dual approach to scalable verification of deep networks. *Proceedings of the Thirty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018b.

Ehlers, R. Formal verification of piece-wise linear feed-forward neural networks. In *Automated Technology for Verification and Analysis*, 2017.

Fawzi, A., Moosavi-Dezfooli, S.-M., and Frossard, P. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems 29*, 2016.

Fawzi, A., Fawzi, H., and Fawzi, O. Adversarial vulnerability for any classifier. In *Advances in Neural Information Processing Systems 31*, 2018.

Fischetti, M. and Jo, J. Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3):296–309, July 2018.

Ford, N., Gilmer, J., and Cubuk, E. D. Adversarial examples are a natural consequence of test error in noise, 2019. URL https://openreview.net/forum?id=S1xoy3CcYX.

Franceschi, J.-Y., Fawzi, A., and Fawzi, O. Robustness of classifiers to uniform \(L_p\) and gaussian noise. In *21st International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2018.

Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. T. A12: safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings*, 21-23 May 2018, San Francisco, California, USA, pp. 3–18, 2018.

Gilmer, J., Adams, R. P., Goodfellow, I., Andersen, D., and Dahl, G. E. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018a.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., and Goodfellow, I. The relationship between high-dimensional geometry and adversarial examples. *arXiv preprint arXiv:1801.02774*, 2018b.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems 30*, 2017.

Huang, X., Kwiatkowska, M., Wang, S., and Wu, M. Safety verification of deep neural networks. *Computer Aided Verification*, 2017.

Hung, K. and Fithian, W. Rank verification for exponential families. *arXiv preprint arXiv:1610.03944*, 2017.

Kannan, H., Kurakin, A., and Goodfellow, I. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.
Certified Adversarial Robustness via Randomized Smoothing

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. Lecture Notes in Computer Science, pp. 97117, 2017. ISSN 1611-3349.

Kolter, J. Z. and Madry, A. Adversarial robustness: Theory and practice. https://adversarial-ml-tutorial.org/adversarial_examples/, 2018.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. 2017. URL https://arxiv.org/abs/1611.01236.

Lee, L., Chen, C., Wang, W., and Carin, L. Second-order adversarial attack and certifiable robustness. arXiv preprint arXiv:1809.03113, 2018.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In The European Conference on Computer Vision (ECCV), September 2018.

Lomuscio, A. and Maganti, L. An approach to reachability analysis for feed-forward relu neural networks, 2017.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations, 2018.

Malmoujifar, S., Diochnos, D. I., and Mahmoody, M. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. Conference on Artificial Intelligence (AAAI), 2019.

Mirman, M., Gehr, T., and Vechev, M. Differentiable abstract interpretation for provably robust neural networks. In Proceedings of the 35th International Conference on Machine Learning, 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, 2011.

Neyman, J. and Pearson, E. S. 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, 231:289–337, 1933.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. In International Conference on Learning Representations, 2018a.

Raghunathan, A., Steinhardt, J., and Liang, P. Semidefinite relaxations for certifying robustness to adversarial examples. In Advances in Neural Information Processing Systems 31, 2018b.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In Advances in Neural Information Processing Systems 31, 2018.

Shafahi, A., Huang, W. R., Studer, C., Feizi, S., and Goldstein, T. Are adversarial examples inevitable? In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=r1lWUoA9FQ.

Singh, G., Gehr, T., Mirman, M., Püschel, M., and Vechev, M. Fast and effective robustness certification. In Advances in Neural Information Processing Systems 31, 2018.

Smilkov, D., Thorat, N., Kim, B., Vargas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825, 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In International Conference on Learning Representations, 2014.

Tjeng, V., Xiao, K. Y., and Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=HyGIdiRqtm.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

Tsuetsu, Y., Sato, I., and Sugiyama, M. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In Advances in Neural Information Processing Systems 31, 2018.

Uesato, J., O’Donoghue, B., Kohli, P., and van den Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In Proceedings of the 35th International Conference on Machine Learning, 2018.
Certified Adversarial Robustness via Randomized Smoothing

Wang, S., Chen, Y., Abdou, A., and Jana, S. Mixtrain: Scalable training of formally robust neural networks. *arXiv preprint arXiv:1811.02625*, 2018a.

Wang, S., Pei, K., Whitehouse, J., Yang, J., and Jana, S. Efficient formal safety analysis of neural networks. In *Advances in Neural Information Processing Systems 31*. 2018b.

Webb, S., Rainforth, T., Teh, Y. W., and Kumar, M. P. Statistical verification of neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1xcx3C5FX.

Weng, L., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Daniel, L., Boning, D., and Dhillon, I. Towards fast computation of certified robustness for ReLU networks. In *Proceedings of the 35th International Conference on Machine Learning*, 2018a.

Weng, T.-W., Zhang, H., Chen, P.-Y., Yi, J., Su, D., Gao, Y., Hsieh, C.-J., and Daniel, L. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*, 2018b.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems 31*, 2018.

Zantedeschi, V., Nicolae, M.-I., and Rawat, A. Efficient defenses against adversarial attacks. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security - AISec 17*, 2017.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems 31*. 2018.

Zheng, S., Song, Y., Leung, T., and Goodfellow, I. J. Improving the robustness of deep neural networks via stability training. In *Computer Vision and Pattern Recognition*, 2016.
A. Full Proof of Theorem 1

Here we provide the complete proof for Theorem 1. We fist prove the following lemma, which is essentially a restatement of the Neyman-Pearson lemma (Neyman & Pearson, 1933) from statistical hypothesis testing.

**Lemma 3 (Neyman-Pearson).** Let $X$ and $Y$ be random variables in $\mathbb{R}^d$ with densities $\mu_X$ and $\mu_Y$. Let $h: \mathbb{R}^d \to \{0, 1\}$ be a random or deterministic function. Then:

1. If $S = \{ z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \leq t \}$ for some $t > 0$ and $P(h(X) = 1) \geq P(X \in S)$, then $P(h(Y) = 1) \geq P(Y \in S)$.

2. If $S = \{ z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \geq t \}$ for some $t > 0$ and $P(h(X) = 1) \leq P(X \in S)$, then $P(h(Y) = 1) \leq P(Y \in S)$.

**Proof.** Without loss of generality, we assume that $h$ is random and write $h(1|x)$ for the probability that $h(x) = 1$.

First we prove part 1. We denote the complement of $S$ as $S^c$.

$$
P(h(Y) = 1) - P(Y \in S) = \int_{\mathbb{R}^d} h(1|z) \mu_Y(z) dz - \int_S \mu_Y(z) dz$$

$$= \left[ \int_{S^c} h(1|z) \mu_Y(z) dz + \int_S h(1|z) \mu_Y(z) dz \right] - \left( \int_{S^c} h(1|z) \mu_Y(z) dz + \int_S h(0|z) \mu_Y(z) dz \right)$$

$$= \int_{S^c} h(1|z) \mu_Y(z) dz - \int_S h(0|z) \mu_Y(z) dz$$

$$\geq t \left[ \int_{S^c} h(1|z) \mu_X(z) dz - \int_S h(0|z) \mu_X(z) dz \right]$$

$$= t \left[ \int_{S^c} h(1|z) \mu_X(z) dz + \int_S h(1|z) \mu_X(z) dz - \int_S h(1|z) \mu_X(z) dz - \int_S h(0|z) \mu_X(z) dz \right]$$

$$= t \left[ \int_{\mathbb{R}^d} h(1|z) \mu_X(z) dz - \int_S \mu_X(z) dz \right]$$

$$\geq 0$$

The inequality in the middle is due to the fact that $\mu_Y(z) \leq t \mu_X(z) \forall z \in S$ and $\mu_Y(z) \geq t \mu_X(z) \forall z \in S^c$. The inequality at the end is because both terms in the product are non-negative by assumption.

The proof for part 2 is virtually identical, except both “$\geq$” become “$\leq$”. □

**Remark: connection to statistical hypothesis testing.** Part 2 of Lemma 3 is known in the field of statistical hypothesis testing as the Neyman-Pearson Lemma (Neyman & Pearson, 1933). The hypothesis testing problem is this: we are given samples (or, in this case, one sample) that comes from one of two distributions over $\mathbb{R}^d$: either the null distribution $X$ or the alternative distribution $Y$. We would like to identify which distribution the sample came from. It is worse to say “$Y$” when the true answer is “$X$” than to say “$X$” when the true answer is “$Y$.” Therefore we seek a (potentially randomized) procedure $h: \mathbb{R}^d \to \{0, 1\}$ which returns “$Y$” when the sample really came from $X$ with probability no greater than some failure rate $\alpha$. In particular, out of all such rules $h$, we would like the uniformly most powerful one $h^*$, i.e. the rule which is most likely to correctly say “$Y$” when the sample really came from $Y$. Neyman & Pearson (1933) showed that $h^*$ is the rule which returns “$Y$” deterministically on the set $S^* = \{ z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \geq t \}$ for whichever $t$ makes $P(X \in S^*) = \alpha$. In other words, to state this in a form that looks like Part 2 of Lemma 3: if $h$ is a different rule with $P(h(X) = 1) \leq \alpha$, then $h^*$ is more powerful than $h$, i.e. $P(h(Y) = 1) \leq P(Y \in S^*)$.

Now we state the special case of Lemma 3 for when $X$ and $Y$ are isotropic Gaussians.

**Lemma 4.** Let $X \sim \mathcal{N}(x, \sigma^2 I)$ and $Y \sim \mathcal{N}(x + \delta, \sigma^2 I)$. Let $h: \mathbb{R}^d \to \{0, 1\}$ be any deterministic or random function. Then:

1. If $S = \{ z \in \mathbb{R}^d : \delta^T z \leq \beta \}$ for some $\beta$ and $P(h(X) = 1) \geq P(X \in S)$, then $P(h(Y) = 1) \geq P(Y \in S)$.
2. If \( S = \{ z \in \mathbb{R}^d : \delta^T z \geq \beta \} \) for some \( \beta \) and \( \mathbb{P}(h(X) = 1) \leq \mathbb{P}(X \in S) \), then \( \mathbb{P}(h(Y) = 1) \leq \mathbb{P}(Y \in S) \)

**Proof.** This lemma is the special case of Lemma 3 when \( X \) and \( Y \) are isotropic Gaussians with means \( x \) and \( x + \delta \). By Lemma 3 it suffices to simply show that for any \( \beta \), there is some \( t > 0 \) for which:

\[
\{ z : \delta^T z \leq \beta \} = \left\{ z : \frac{\mu_Y(z)}{\mu_X(z)} \leq t \right\} \quad \text{and} \quad \{ z : \delta^T z \geq \beta \} = \left\{ z : \frac{\mu_Y(z)}{\mu_X(z)} \geq t \right\}
\]

(7)

The likelihood ratio for this choice of \( X \) and \( Y \) turns out to be:

\[
\frac{\mu_Y(z)}{\mu_X(z)} = \frac{\exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{d} (z_i - (x_i + \delta_i))^2 \right)}{\exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{d} (z_i - x_i)^2 \right)} = \exp \left( \frac{1}{2\sigma^2} \sum_{i=1}^{d} 2z_i\delta_i - \delta_i^2 - 2x_i\delta_i \right) = \exp(a\delta^T z + b)
\]

where \( a > 0 \) and \( b \) are constants w.r.t \( z \), specifically \( a = \frac{1}{\sigma^2} \) and \( b = -\frac{1}{2\sigma^2} (2\delta^T x + \|\delta\|^2) \).

Therefore, given any \( \beta \) we may take \( t = \exp(a\beta + b) \), noticing that

\[
\delta^T z \leq \beta \iff \exp(a\delta^T z + b) \leq t
\]

\[
\delta^T z \geq \beta \iff \exp(a\delta^T z + b) \geq t
\]

Finally, we prove Theorem 1 and Theorem 2.

**Theorem 1 (main bound).** Let \( f : \mathbb{R}^d \to \mathcal{Y} \) be any deterministic or random function and let \( \epsilon \sim \mathcal{N}(0, \sigma^2 I_d) \). Define \( g(x) = \arg \max_{c \in \mathcal{Y}} \mathbb{P}(f(x + \epsilon) = c) \). Suppose for a specific \( x \in \mathbb{R}^d \), there exist \( c_A \in \mathcal{Y} \) and \( p_A, p_B \in [0, 1] \) such that:

\[
\mathbb{P}(f(x + \epsilon) = c_A) \geq p_A \geq p_B \geq \max_{c \neq c_A} \mathbb{P}(f(x + \epsilon) = c)
\]

(8)

Then \( g(x) = g(x + \delta) \) whenever \( \|\delta\|_2 < \frac{\sigma}{\sqrt{2}} (\Phi^{-1}(p_A) - \Phi^{-1}(p_B)) \)

**Proof.** We will derive conditions on \( \delta \) which ensure that \( g(x) = g(x + \delta) \). It follows from the definition of \( g \) and (8) that \( g(x) = c_A \). To show that \( g(x + \delta) = c_A \), we need to show that

\[
\mathbb{P}(f(x + \delta + \epsilon) = c_A) > \max_{c_B \in \mathcal{Y} \setminus \{c_A\}} \mathbb{P}(f(x + \delta + \epsilon) = c_B)
\]

We will prove that \( \mathbb{P}(f(x + \delta + \epsilon) = c_A) > \mathbb{P}(f(x + \delta + \epsilon) = c_B) \) for any class \( c_B \neq c_A \). Fix one such class \( c_B \) without loss of generality.

For brevity, define the random variables

\[
X := x + \epsilon = \mathcal{N}(x, \sigma^2 I)
\]

\[
Y := x + \delta + \epsilon = \mathcal{N}(x + \delta, \sigma^2 I)
\]

In this notation, we know from (8) that

\[
\mathbb{P}(f(X) = c_A) \geq p_A \quad \text{and} \quad \mathbb{P}(f(X) = c_B) \leq p_B
\]

(9)
Figure 9. Illustration of the proof of Theorem 1. The solid line concentric circles are the density level sets of $X := x + \epsilon$; the dashed line concentric circles are the level sets of $Y := x + \delta + \epsilon$. The set $A$ is in blue and the set $B$ is in red. The figure on the left depicts a situation where $P(Y \in A) > P(Y \in B)$, and hence $g(x + \delta)$ may equal $c_A$. The figure on the right depicts a situation where $P(Y \in A) < P(Y \in B)$ and hence $g(x + \delta) \neq c_A$.

and our goal is to show that
\[
P(f(Y) = c_A) \geq P(f(Y) = c_B) \quad (10)
\]

Define the half-spaces:
\[
A := \{ z : \delta^T(z - x) \leq \sigma \| \delta \| \Phi^{-1}(p_A) \} \\
B := \{ z : \delta^T(z - x) \geq \sigma \| \delta \| \Phi^{-1}(1 - p_B) \}
\]

Algebra (deferred to the end) shows that $P(X \in A) = p_A$. Therefore, by (9) we know that $P(f(X) = c_A) \geq P(X \in A)$. Hence we may apply Lemma 4 with $h(z) := 1[f(z) = c_A]$ to conclude:
\[
P(f(Y) = c_A) \geq P(Y \in A) \quad (11)
\]

Similarly, algebra shows that $P(X \in B) = p_B$. Therefore, by (9) we know that $P(f(X) = c_B) \leq P(X \in B)$. Hence we may apply Lemma 4 with $h(z) := 1[f(z) = c_B]$ to conclude:
\[
P(f(Y) = c_B) \leq P(Y \in B) \quad (12)
\]

To guarantee (10), we see from (11, 12) that it suffices to show that $P(Y \in A) > P(Y \in B)$ as this step completes the chain of inequalities
\[
P(f(Y) = c_A) \geq P(Y \in A) > P(Y \in B) \geq P(f(Y) = c_B) \quad (13)
\]

Algebra shows that:
\[
P(Y \in A) = \Phi \left( \Phi^{-1}(p_A) - \frac{\| \delta \|}{\sigma} \right) \\
P(Y \in B) = \Phi \left( \Phi^{-1}(p_B) + \frac{\| \delta \|}{\sigma} \right)
\]

Simple algebra shows that $P(Y \in A) > P(Y \in B)$ if and only if:
\[
\| \delta \| < \frac{\sigma}{2} (\Phi^{-1}(p_A) - \Phi^{-1}(p_B)) \quad (14)
\]

which recovers the theorem statement.
We now restate and prove Theorem 2, which shows that the bound in Theorem 1 is tight. We adopt the same notation from Theorem 1.

**Theorem 2.** For any perturbation \( \delta \in \mathbb{R}^d \) with \( \|\delta\| > R \), there exists some \( f \) consistent with (8) for which \( g(x) \neq g(x + \delta) \).

**Proof.** We re-use notation from the preceding proof.

Pick any class \( c_B \) arbitrarily. Define \( A \) and \( B \) as above, and consider the function

\[
    f^*(x) := \begin{cases} 
        c_A & \text{if } x \in A \\
        c_B & \text{if } x \in B \\
        \text{other classes} & \text{otherwise}
    \end{cases}
\]

The function \( f^* \) satisfies (8). Much like (14), it is easy to show that \( \|\delta\| > R \) if and only if \( \mathbb{P}(Y \in A) < \mathbb{P}(Y \in B) \). Therefore, since \( \|\delta\| > R \), we have \( \mathbb{P}(Y \in A) < \mathbb{P}(Y \in B) \). This means that \( g(x + \delta) \neq c_A \). Therefore, \( g(x + \delta) \neq g(x) \). \( \square \)

A.0.1. DEFERRED ALGEBRA

**Proposition.** \( \mathbb{P}(X \in A) = p_A \)

**Proof.** Recall that \( X \sim \mathcal{N}(x, \sigma^2 I) \) and \( A = \{ z : \delta^T (z - x) \leq \sigma \|\delta\| \Phi^{-1}(p_A) \} \).

\[
    \mathbb{P}(X \in A) = \mathbb{P}(\delta^T (X - x) \leq \sigma \|\delta\| \Phi^{-1}(p_A)) \\
    = \mathbb{P}(\delta^T \mathcal{N}(0, \sigma^2 I) \leq \sigma \|\delta\| \Phi^{-1}(p_A)) \\
    = \mathbb{P}(\sigma \|\delta\| Z \leq \sigma \|\delta\| \Phi^{-1}(p_A)) \\
    = \Phi(\Phi^{-1}(p_A)) \\
    = p_A
\]

\( \square \)

**Proposition.** \( \mathbb{P}(X \in B) = \overline{p_B} \)

**Proof.** Recall that \( X \sim \mathcal{N}(x, \sigma^2 I) \) and \( B = \{ z : \delta^T (z - x) \leq \sigma \|\delta\| \Phi^{-1}(1 - \overline{p_B}) \} \).

\[
    \mathbb{P}(X \in A) = \mathbb{P}(\delta^T (X - x) \geq \sigma \|\delta\| \Phi^{-1}(1 - \overline{p_B})) \\
    = \mathbb{P}(\delta^T \mathcal{N}(0, \sigma^2 I) \geq \sigma \|\delta\| \Phi^{-1}(1 - \overline{p_B})) \\
    = \mathbb{P}(\sigma \|\delta\| Z \geq \sigma \|\delta\| \Phi^{-1}(1 - \overline{p_B})) \\
    = \mathbb{P}(Z \geq \Phi^{-1}(1 - \overline{p_B})) \\
    = 1 - \Phi(\Phi^{-1}(1 - \overline{p_B})) \\
    = \overline{p_B}
\]

\( \square \)

**Proposition.** \( \mathbb{P}(Y \in A) = \Phi \left( \Phi^{-1}(p_A) - \frac{\|\delta\|}{\sigma} \right) \)
Proof. Recall that $y \sim \mathcal{N}(x + \delta, \sigma^2 I)$ and $A = \{z : \delta^T(z - x) \leq \sigma \|\delta\| \Phi^{-1}(p_A)\}$.

$$
\mathbb{P}(Y \in A) = \mathbb{P}(\delta^T(Y - x) \leq \sigma \|\delta\| \Phi^{-1}(p_A)) \\
= \mathbb{P}(\delta^T \mathcal{N}(0, \sigma^2 I) + \|\delta\|^2 \leq \sigma \|\delta\| \Phi^{-1}(p_A)) \\
= \mathbb{P}(\sigma \|\delta\| Z \leq \sigma \|\delta\| \Phi^{-1}(p_A) - \|\delta\|^2) \\
= \mathbb{P} \left( Z \leq \Phi^{-1}(p_A) - \frac{\|\delta\|^2}{\sigma} \right) \\
= \Phi \left( \Phi^{-1}(p_A) - \frac{\|\delta\|^2}{\sigma} \right)
$$

\[ \square \]

Proposition. $\mathbb{P}(Y \in B) = \Phi \left( \Phi^{-1}(p_B) + \frac{\|\delta\|}{\sigma} \right)$

Proof. Recall that $y \sim \mathcal{N}(x + \delta, \sigma^2 I)$ and $B = \{z : \delta^T(z - x) \geq \sigma \|\delta\| \Phi^{-1}(1 - p_B)\}$.

$$
\mathbb{P}(Y \in B) = \mathbb{P}(\delta^T(Y - x) \geq \sigma \|\delta\| \Phi^{-1}(1 - p_B)) \\
= \mathbb{P}(\delta^T \mathcal{N}(0, \sigma^2 I) + \|\delta\|^2 \geq \sigma \|\delta\| \Phi^{-1}(1 - p_B)) \\
= \mathbb{P}(\sigma \|\delta\| Z + \|\delta\|^2 \geq \sigma \|\delta\| \Phi^{-1}(1 - p_B)) \\
= \mathbb{P} \left( Z \geq \Phi^{-1}(1 - p_B) - \frac{\|\delta\|^2}{\sigma} \right) \\
= \mathbb{P} \left( Z \leq \Phi^{-1}(p_B) + \frac{\|\delta\|}{\sigma} \right) \\
= \Phi \left( \Phi^{-1}(p_B) + \frac{\|\delta\|}{\sigma} \right)
$$

\[ \square \]

B. Other Theorems

Proposition. If $f$ is a binary linear classifier $f(x) = \text{sign}(w^T x + b)$, then the smoothed classifier $g$ is equal to $f$ everywhere.

Proof. By the definition of $g$,

$$
g(x) = 1 \left[ \mathbb{P}_x(f(x + \epsilon) = 1) \geq \frac{1}{2} \right] \\
= 1 \left[ \mathbb{P}_x(\text{sign}(w^T(x + \epsilon) + b) = 1) \geq \frac{1}{2} \right] \\
= 1 \left[ \mathbb{P}_x(w^T x + w^T \epsilon + b \geq 0) \geq \frac{1}{2} \right] \\
= 1 \left[ \mathbb{P}(\sigma \|w\| Z \geq -w^T x - b) \geq \frac{1}{2} \right] \\
= 1 \left[ \mathbb{P} \left( Z \leq \frac{w^T x + b}{\sigma \|w\|} \right) \geq \frac{1}{2} \right] \\
= 1 \left[ \frac{w^T x + b}{\sigma \|w\|} \geq 0 \right] \\
= \text{sign}(w^T x + b) \\
= f(x)
$$

\[ \square \]
**Proposition.** If \( f \) is a binary linear classifier, then our main bound in Theorem 1 will return the certified radius

\[
R = \frac{w^T x + b}{\|w\|}.
\]

**Proof.** In binary classification, \( p_A = 1 - p_B \), so our main bound is \( R = \sigma \Phi^{-1}(p_A) \).

If \( g(x) = 1 \), (i.e. \( w^T x + b \geq 0 \)) then

\[
p_A = \mathbb{P}(f(x + \epsilon) = 1) = \mathbb{P}(w^T (x + \epsilon) + b \geq 0) = \mathbb{P}(\sigma \|w\| Z \geq -w^T x - b) = \Phi\left( \frac{w^T x + b}{\sigma \|w\|} \right).
\]

Similarly, if \( g(x) = 0 \), then \( p_A = \Phi\left( \frac{-w^T x - b}{\sigma \|w\|} \right) \).

Either way, \( p_A = \Phi\left( \frac{|w^T x + b|}{\sigma \|w\|} \right) \).

Therefore, the bound in Theorem 1 returns a radius of

\[
R = \sigma \Phi^{-1}(p_A) = \frac{|w^T x + b|}{\|w\|}.
\]
C. Estimating the certified test-set accuracy

In this appendix, we show how to convert the “approximate certified test accuracy” considered in the main paper into a lower bound on the true certified test accuracy that holds with high probability over the randomness in CERTIFY.

Consider a classifier \( g \), a test set \( S = \{ (x_1, c_1) \ldots (x_m, c_m) \} \), and a radius \( r \). For each example \( i \in [m] \), let \( z_i \) indicate whether \( g \)'s prediction at \( x_i \) is both correct and robust at radius \( r \), i.e.

\[
 z_i = 1[g(x_i + \delta) = c_i \ \forall \|\delta\|_2 \leq r]
\]

The quantity of interest, the certified accuracy \( \rho > 0 \), is defined as \( \frac{1}{m} \sum_{i=1}^{m} z_i \). If \( g \) is a randomized smoothing classifier, we cannot compute this quantity exactly, but we can estimate a lower bound that holds with arbitrarily high probability over the randomness in CERTIFY. In particular, suppose that we run CERTIFY with failure rate \( \alpha \) on each example \( x_i \) in the test set.

Let the Bernoulli random variable \( Y_i \) denote the event that on example \( i \), CERTIFY returns the correct label \( c_A = c_i \) and a certified radius \( R \) which is greater than \( r \). Let \( Y = \sum_{i=1}^{m} Y_i \). In the main paper, we referred to \( Y/m \) as the “approximate certified accuracy.” It is “approximate” because \( Y_i = 1 \) does not mean that \( z_i = 1 \). Rather, we know the following: if \( z_i = 0 \), then \( P(Y_i = 1) \leq \alpha \). We now show how to exploit this fact to construct a one-sided confidence interval for the unobserved quantity \( \frac{1}{m} \sum_{i=1}^{m} z_i \) using the observed quantities \( Y \) and \( m \).

**Theorem 5.** For any \( \rho > 0 \), with probability at least \( 1 - \rho \) over the randomness in CERTIFY,

\[
 \frac{1}{m} \sum_{i=1}^{m} z_i \geq 1 - \alpha \left( \frac{Y}{m} - \alpha - \sqrt{\frac{2\alpha(1-\alpha)\log(1/\rho)}{m}} - \frac{\log(1/\rho)}{3m} \right)
\]

**Proof.** Let \( m_{\text{good}} = \sum_{i:z_i=1} z_i \) and \( m_{\text{bad}} = \sum_{i:z_i=0} (1 - z_i) \) be the number of test examples on which \( z_i = 1 \) or \( z_i = 0 \), respectively. We model \( Y_i \sim \text{Bernoulli}(p_i) \), where \( p_i \) is in general unknown. Let \( Y_{\text{good}} = \sum_{i:z_i=1} Y_i \) and \( Y_{\text{bad}} = \sum_{i:z_i=0} Y_i \).

The quantity of interest, the certified accuracy \( \frac{1}{m} \sum_{i=1}^{m} z_i \), is equal to \( m_{\text{good}}/m \). However, we only observe \( Y = Y_{\text{good}} + Y_{\text{bad}} \).

Note that if \( z_i = 0 \), then \( p_i \leq \alpha \), so we have \( E[Y_i] = p_i \leq \alpha \) and assuming \( \alpha \leq \frac{1}{2} \), we have \( \text{Var}(Y_i) = p_i(1-p_i) \leq \alpha(1-\alpha) \).

Since \( Y_{\text{bad}} \) is a sum of \( m_{\text{bad}} \) independent random variables each bounded between zero and one, with \( E[Y_{\text{bad}}] \leq m_{\text{bad}} \alpha \) and \( \text{Var}(Y_{\text{bad}}) \leq m_{\text{bad}} \alpha(1-\alpha) \), Bernstein’s inequality (Blanchard, 2007) guarantees that with probability at least \( 1 - \rho \) over the randomness in CERTIFY,

\[
 Y_{\text{bad}} \leq m_{\text{bad}} + \sqrt{2m_{\text{bad}} \alpha(1-\alpha) \log(1/\rho)} + \frac{\log(1/\rho)}{3}
\]

From now on, we manipulate this inequality — remember that it holds with probability at least \( 1 - \rho \).

Since \( Y = Y_{\text{good}} + Y_{\text{bad}} \), may write

\[
 Y_{\text{good}} \geq Y - \alpha m_{\text{bad}} - \sqrt{2m_{\text{bad}} \alpha(1-\alpha) \log(1/\rho)} + \frac{\log(1/\rho)}{3}
\]

Since \( m_{\text{good}} \geq Y_{\text{good}} \), we may write

\[
 m_{\text{good}} \geq Y - \alpha m_{\text{bad}} - \sqrt{2m_{\text{bad}} \alpha(1-\alpha) \log(1/\rho)} - \frac{\log(1/\rho)}{3}
\]

Since \( m_{\text{good}} + m_{\text{bad}} = m \), we may write

\[
 m_{\text{good}} \geq \frac{1}{1-\alpha} \left( Y - \alpha m - \sqrt{2m_{\text{bad}} \alpha(1-\alpha) \log(1/\rho)} - \frac{\log(1/\rho)}{3} \right)
\]

Finally, in order to make this confidence interval depend only on observables, we use \( m_{\text{bad}} \leq m \) to write

\[
 m_{\text{good}} \geq \frac{1}{1-\alpha} \left( Y - \alpha m - \sqrt{2m \alpha(1-\alpha) \log(1/\rho)} - \frac{\log(1/\rho)}{3} \right)
\]

Dividing both sides of the inequality by \( m \) recovers the theorem statement.
D. ImageNet and CIFAR-10 Results

D.1. Certification

Tables 2 and 3 show the approximate certified top-1 test set accuracy of randomized smoothing on ImageNet and CIFAR-10 with various noise levels $\sigma$. By “approximate certified accuracy,” we mean that we ran CERTIFY on a subsample of the test set, and for each $r$ we report the fraction of examples on which CERTIFY (a) did not abstain, (b) returned the correct class, and (c) returned a radius $R$ greater than $r$. There is some probability (at most $\alpha$) that any example’s certification is inaccurate. We used $\alpha = 0.001$ and $n = 100000$. On both datasets we certified a subsample of 500 test points. See Appendix G for more experimental details.

| $\sigma$  | $r = 0.0$ | $r = 0.5$ | $r = 1.0$ | $r = 1.5$ | $r = 2.0$ | $r = 2.5$ | $r = 3.0$ |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0.25      | 0.67      | 0.49      | 0.00      | 0.00      | 0.00      | 0.00      | 0.00      |
| 0.50      | 0.57      | 0.46      | 0.37      | 0.29      | 0.00      | 0.00      | 0.00      |
| 1.00      | 0.44      | 0.38      | 0.33      | 0.26      | 0.19      | 0.15      | 0.12      |

Table 2. Approximate certified test accuracy on ImageNet. Each row is a setting of the hyperparameter $\sigma$, each column is an $\ell_2$ radius. The entry of the best $\sigma$ for each radius is bolded. For comparison, random guessing would attain 0.001 accuracy.

| $\sigma$  | $r = 0.0$ | $r = 0.25$ | $r = 0.5$ | $r = 0.75$ | $r = 1.0$ | $r = 1.25$ | $r = 1.5$ |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0.25      | 0.75      | 0.60      | 0.43      | 0.26      | 0.00      | 0.00      | 0.00      |
| 0.50      | 0.65      | 0.54      | 0.41      | 0.32      | 0.23      | 0.15      | 0.09      |
| 1.00      | 0.47      | 0.39      | 0.34      | 0.28      | 0.21      | 0.17      | 0.14      |

Table 3. Approximate certified test accuracy on CIFAR-10. Each row is a setting of the hyperparameter $\sigma$, each column is an $\ell_2$ radius. The entry of the best $\sigma$ for each radius is bolded. For comparison, random guessing would attain 0.1 accuracy.

D.2. Prediction

Table 4 shows the performance of PREDICT as the number of Monte Carlo samples $n$ is varied between 100 and 10,000. Suppose that for some test example $(x, c)$, PREDICT returns the label $c_A$. We say that this prediction was correct if $c_A = c$ and we say that this prediction was accurate if $c_A = g(x)$. For example, a prediction could be correct but inaccurate if $g$ is wrong at $x$, yet PREDICT accidentally returns the correct class. Ideally, we’d like PREDICT to be both correct and accurate.

With $n = 100$ Monte Carlo samples and a failure rate of $\alpha = 0.001$, PREDICT is cheap to evaluate (0.15 seconds on our hardware) yet it attains relatively high top-1 accuracy of 65% on the ImageNet test set, and only abstains 12% of the time. When we use $n = 10,000$ Monte Carlo samples, PREDICT takes longer to evaluate (15 seconds), yet only abstains 4% of the time. Interestingly, we observe from Table 4 that most of the abstentions when $n = 100$ were for examples on which $g$ was wrong, so in practice we would lose little accuracy by taking $n$ to be as small as 100.

| $n$      | CORRECT, ACCURATE | CORRECT, INACCURATE | INCORRECT, ACCURATE | INCORRECT, INACCURATE | ABSTAIN |
|----------|-------------------|---------------------|---------------------|-----------------------|---------|
| 100      | 0.65              | 0.00                | 0.23                | 0.00                  | 0.12    |
| 1000     | 0.68              | 0.00                | 0.28                | 0.00                  | 0.04    |
| 10000    | 0.69              | 0.00                | 0.30                | 0.00                  | 0.01    |

Table 4. Performance of PREDICT as $n$ is varied. The dataset was ImageNet and $\sigma = 0.25$, $\alpha = 0.001$. Each column shows the fraction of test examples which ended up in one of five categories; the prediction at $x$ is “correct” if PREDICT returned the true label, while the prediction is “accurate” if PREDICT returned $g(x)$. Computing $g(x)$ exactly is not possible, so in order to determine whether PREDICT was accurate, we took the gold standard to be the top class over $n = 100,000$ Monte Carlo samples.
E. Additional Experiments

E.1. High-probability guarantees

Appendix C details how to use CERTIFY to obtain a lower bound on the certified test accuracy at radius $r$ of a randomized smoothing classifier that holds with high probability over the randomness in CERTIFY. In the main paper, we declined to do this and simply reported the approximate certified test accuracy, defined as the fraction of test examples for which CERTIFY gives the correct prediction and certifies it at radius $r$. Of course, with some probability (guaranteed to be less than $\alpha$), each of these certifications is wrong.

However, we now demonstrate empirically that there is a negligible difference between a proper high-probability lower bound on the certified accuracy and the approximate version that we reported in the paper. We created a randomized smoothing classifier $g$ on ImageNet with a ResNet-50 base classifier and noise level $\sigma = 0.25$. We used CERTIFY with $\alpha = 0.001$ to certify a subsample of 500 examples from the ImageNet test set. From this we computed the approximate certified test accuracy at each radius $r$. Then we used the correction from Appendix C with $\rho = 0.001$ to obtain a lower bound on the certified test accuracy at $r$ that holds pointwise with probability at least $1 - \rho$ over the randomness in CERTIFY. Figure 10 plots both quantities as a function of $r$. Observe that the difference is so negligible that the lines almost overlap.

![Figure 10](image)

Figure 10. The difference between the approximate certified accuracy, and a high-probability lower bound on the certified accuracy, is negligible.

E.2. How much noise to use when training the base classifier?

In the main paper, whenever we created a randomized smoothing classifier $g$ at noise level $\sigma$, we always trained the corresponding base classifier $f$ with Gaussian data augmentation at noise level $\sigma$. In Figure 11, we show the effects of training the base classifier with a different level of Gaussian noise. Observe that $g$ has a lower certified accuracy if $f$ was trained using a different noise level. It seems to be worse to train with noise $< \sigma$ than to train with noise $> \sigma$.

![Figure 11](image)

Figure 11. Vary training noise while holding prediction noise fixed at $\sigma = 0.50$. 
F. Derivation of Prior Randomized Smoothing Bounds

In this appendix, we derive the randomized smoothing bounds of Lecuyer et al. (2019) and Li et al. (2018) using the notation of our paper.

F.1. Problem Setup

Let $X \sim \mathcal{N}(x, \sigma^2 I)$ and $Y \sim \mathcal{N}(x + \delta, \sigma^2 I)$. Let $f : \mathbb{R}^d \rightarrow \mathcal{Y}$ be the base classifier. We are told that

$$\mathbb{P}(f(X) = c_A) \geq p_A \geq \frac{p_B}{1 - \epsilon} \geq \max_{c \neq c_A} \mathbb{P}(f(X) = c)$$ (16)

Our goal is to identify conditions on $\delta$ which guarantee, for each $c_B \neq c_A$, that:

$$\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(f(Y) = c_B)$$ (17)

F.2. Lecuyer

Note: we use $\beta$ and $\gamma$ where Lecuyer et al. (2019) uses $\epsilon$ and $\delta$.

Suppose that we have some $0 < \beta \leq 1$ and $\gamma > 0$ such that

$$\sigma^2 = \frac{\|\delta\|^2}{\beta^2} - 2 \log \frac{1.25}{\gamma}$$ (18)

The “Gaussian mechanism” from differential privacy guarantees that:

$$\mathbb{P}(f(X) = c_A) \leq \exp(\beta) \mathbb{P}(f(Y) = c_A) + \gamma$$ (19)

and, symmetrically,

$$\mathbb{P}(f(Y) = c_B) \leq \exp(\beta) \mathbb{P}(f(X) = c_B) + \gamma$$ (20)

See Lecuyer et al. (2019), Lemma 2 for how to obtain this form from the standard form of the $(\beta, \gamma)$ DP definition.

Together, (19) and (20) imply that to guarantee $\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(f(Y) = c_B)$ it suffices to show that:

$$\mathbb{P}(f(X) = c_A) \geq \exp(2\beta) \mathbb{P}(f(X) = c_B) + \gamma(1 + \exp(\beta))$$ (21)

In fact, by (16) it suffices to show:

$$p_A \geq \exp(2\beta) \frac{p_B}{1 - \epsilon} + \gamma(1 + \exp(\beta))$$ (22)

Now, inverting (18), we obtain:

$$\gamma = 1.25 \exp \left( -\frac{\sigma^2 \beta^2}{2 \|\delta\|^2} \right)$$ (23)

Plugging (23) into (22), we see that to guarantee $\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(f(Y) = c_B)$ it suffices to show that:

$$p_A \geq \exp(2\beta) \frac{p_B}{1 - \epsilon} + 1.25 \exp \left( -\frac{\sigma^2 \beta^2}{2 \|\delta\|^2} \right)(1 + \exp(\beta))$$ (24)

which rearranges to:

$$\frac{p_A - \exp(2\beta) \frac{p_B}{1 - \epsilon}}{1.25(1 + \exp(\beta))} \geq \exp \left( -\frac{\sigma^2 \beta^2}{2 \|\delta\|^2} \right)$$ (25)

Since the RHS is always positive, and the denominator on the LHS is always positive, this condition can only possibly hold if the numerator on the LHS is positive. Therefore, we need to restrict $\beta$ to

$$0 < \beta \leq \min \left( 1, \frac{1}{2} \log \frac{p_A}{p_B} \right)$$ (26)
The condition (25) is equivalent to:

$$\|\delta\|_2^2 \log \frac{1.25(1 + \exp(\beta))}{p_A - \exp(2\beta)p_B} \leq \frac{\sigma^2 \beta^2}{2}$$

(27)

Since $p_A \leq 1$ and $p_B \geq 0$, the denominator in the LHS is $\leq 1$ which is in turn $\leq$ the numerator on the LHS. Therefore, the term inside the log in the LHS is greater than 1, so the log term on the LHS is greater than zero. Therefore, we may divide both sides of the inequality by the log term on the LHS to obtain:

$$\|\delta\|_2^2 \leq \frac{\sigma^2 \beta^2}{2 \log \left(\frac{1.25(1 + \exp(\beta))}{p_A - \exp(2\beta)p_B}\right)}$$

(28)

Finally, we take the square root and maximize the bound over all valid $\beta$ (26) to yield:

$$\|\delta\| \leq \sup_{0 < \beta \leq \min \left(1, \frac{1}{2} \log \frac{p_A}{p_B} \right)} \sigma \beta \sqrt{2 \log \left(\frac{1.25(1 + \exp(\beta))}{p_A - \exp(2\beta)p_B}\right)}$$

(29)

Figure 12a plots this bound at varying settings of the tuning parameter $\beta$, while figure 12c plots how the bound varies with $\beta$ for a fixed $p_A$ and $p_B$.

F.3. Li

A generalization of KL divergence, the $\alpha$-Renyi divergence is an information theoretic measure of distance between two distributions. It is parameterized by some $\alpha > 0$. The $\alpha$-Renyi divergence between two discrete distributions $P$ and $Q$ is defined as:

$$D_{\alpha}(P||Q) := \frac{1}{\alpha - 1} \log \left(\sum_{i=1}^{k} \frac{p_i^\alpha}{q_i^\alpha - 1}\right)$$

(30)

In the continuous case, this sum is replaced with an integral. The divergence is undefined when $\alpha = 1$ since a division by zero occurs, but the limit of $D_{\alpha}(P||Q)$ as $\alpha \to 1$ is the KL divergence between $P$ and $Q$.

Li et al. (2018) prove that if $P$ is a discrete distribution for which the highest probability class has probability $\geq p_A$ and all other classes have probability $\leq p_B$, then for any other discrete distribution $Q$ for which

$$D_{\alpha}(P||Q) \leq -\log \left(1 - p_A - p_B + 2 \left(\frac{1}{2} (p_A^{1-\alpha} + p_B^{1-\alpha})^{1-\alpha}\right)\right)$$

(31)

the highest-probability class in $Q$ is guaranteed to be the same as the highest-probability class in $P$.

We now apply this result to the discrete distributions $P = f(X)$ and $Q = f(Y)$. If $D_{\alpha}(f(X)||f(Y))$ satisfies (31), then it is guaranteed that $g(x) = g(x + \delta)$.

A result from information theory called the data processing inequality states that applying a function to two random variables can only decrease the $\alpha$-Renyi divergence between them. In particular,

$$D_{\alpha}(f(X)||f(Y)) \leq D_{\alpha}(X||Y)$$

(32)

There is a closed-form expression for the $\alpha$-Renyi divergence between two Gaussians:

$$D_{\alpha}(X||Y) = \frac{\alpha \|\delta\|^2}{2\sigma^2}$$

(33)

Therefore, we can guarantee that $g(x) = g(x + \delta)$ so long as

$$\frac{\alpha \|\delta\|^2}{2\sigma^2} \leq -\log \left(1 - p_A - p_B + 2 \left(\frac{1}{2} (p_A^{1-\alpha} + p_B^{1-\alpha})^{1-\alpha}\right)\right)$$

(34)
which simplifies to

\[ \| \delta \| \leq \sigma \sqrt{\frac{-2}{\alpha} \log \left( 1 - p_A - p_B + 2 \left( \frac{1}{2} (p_A^{1-\alpha} + p_B^{1-\alpha})^{1-\alpha} \right) \right)} \]  

(35)

Finally, since this result holds for any \( \alpha > 0 \), we may maximize over \( \alpha \) to obtain the largest possible certified radius:

\[ \| \delta \| \leq \sup_{\alpha > 0} \sigma \sqrt{\frac{-2}{\alpha} \log \left( 1 - p_A - p_B + 2 \left( \frac{1}{2} (p_A^{1-\alpha} + p_B^{1-\alpha})^{1-\alpha} \right) \right)} \]  

(36)

Figure 12b plots this bound at varying settings of the tuning parameter \( \alpha \), while figure 12d plots how the bound varies with \( \alpha \) for a fixed \( p_A \) and \( p_B \).

(a) The Lecuyer et al. (2019) bound over several settings of \( \beta \). The brown line is the pointwise supremum over all eligible \( \beta \), computed numerically.

(b) The Li et al. (2018) bound over several settings of \( \alpha \). The purple line is the pointwise supremum over all eligible \( \alpha \), computed numerically.

(c) Tuning the Lecuyer et al. (2019) bound wrt \( \beta \) when \( p_A = 0.8, p_B = 0.2 \)

(d) Tuning the Li et al. (2018) bound wrt \( \alpha \) when \( p_A = 0.999, p_B = 0.0001 \)
G. Experiment Details

G.1. Comparison to baselines

We compared randomized smoothing against three recent approaches for $\ell_2$-robust classification (Tsuzuku et al., 2018; Wong et al., 2018; Zhang et al., 2018). Tsuzuku et al. (2018) and Wong et al. (2018) propose both a robust training method and a complementary certification mechanism, while Zhang et al. (2018) propose a method to certify generically trained networks. In all cases we compared against networks provided by the authors. We compared against Wong et al. (2018) and Zhang et al. (2018) on CIFAR-10, and we compared against Tsuzuku et al. (2018) on SVHN.

In image classification it is common practice to preprocess a dataset by subtracting from each channel the mean over the dataset, and dividing each channel by the standard deviation over the dataset. However, we wanted to report certified radii in the original image coordinates rather than in the standardized coordinates. Therefore, throughout most of this work we first added the Gaussian noise, and then standardized the channels, before feeding the image to the base classifier. (In the practical PyTorch implementation, the first layer of the base classifier was a layer that standardized the input.) However, all of the baselines we compared against provided pre-trained networks which assumed that the dataset was first preprocessed in a specific way. Therefore, when comparing against the baselines we also preprocessed the datasets first, so that we could report certified radii that were directly comparable to the radii reported by the baseline methods.

We note that Wong et al. (2018) and Zhang et al. (2018) use very different preprocessing steps for CIFAR-10: Wong et al. (2018) subtracted the mean of each channel and divided by the standard deviation of each channel, while Zhang et al. (2018) simply subtracted 0.5 from each channel. This is why the numbers reported for the certified accuracy of randomized smoothing differ between Figure 5b and Figure 5c.

Comparison to Wong et al. (2018) Following Wong et al. (2018), the CIFAR-10 dataset was preprocessed by subtracting the mean $(0.4914, 0.4822, 0.4465)$ and dividing by the standard deviation $(0.2023, 0.1994, 0.2010)$.

While the body of the Wong et al. (2018) paper focuses on $\ell_\infty$ certified robustness, their algorithm naturally extends to $\ell_2$ certified robustness, as developed in the appendix of the paper. We obtained from the authors three residual networks trained with varying settings of their hyperparameter $\epsilon \in \{0.157, 0.628, 2.51\}$. We used code publicly released by the authors at https://github.com/locuslab/convex_adversarial/blob/master/examples/cifar_evaluate.py to compute the robustness radius of test images. The code accepts a radius and returns TRUE (robust) or FALSE (not robust); we incorporated this subroutine into a binary search procedure to find the largest radius for which the code returned TRUE.

For randomized smoothing we used $\sigma = 0.6$ and a 110-layer residual network base classifier. We ran CERTIFY with $n = 100,000$ and $\alpha = 0.001$.

For both methods, we computed the certified radius on every 10th image from the CIFAR-10 test set, for 1000 images total.

Comparison to Tsuzuku et al. (2018) Following Tsuzuku et al. (2018), the SVHN dataset was not preprocessed except that pixels were divided by 255 so as to lie within $[0, 1]$.

We compared against a pretrained network provided to us by the authors in which the hyperparameter of their method was set to $c = 0.1$. The network was a wide residual network with 16 layers and a width factor of 4. We used the authors’ code at https://github.com/ytsmiling/lmt to compute the robustness radius of test images.

For randomized smoothing we used $\sigma = 0.1$ and a 110-layer residual network base classifier. We ran CERTIFY with $n = 100,000$ and $\alpha = 0.001$.

For Tsuzuku et al. (2018), we computed the certified radius on every image in the SVHN test set; for randomized smoothing, we took every 40th image, for 651 images total.

Comparison to Zhang et al. (2018) Following Zhang et al. (2018), the CIFAR-10 dataset was preprocessed by subtracting 0.5 from each pixel.

We compared against the cifar_7_1024_vanilla network released by the authors, which is a 7-layer MLP. We used the authors’ code at https://github.com/IBM/CROWN-Robustness-Certification to compute the robustness radius of test images.
For randomized smoothing we used \( \sigma = 1.2 \) and a 110-layer residual network base classifier. We ran CERTIFY with \( n = 100,000 \) and \( \alpha = 0.001 \).

For both methods, we computed the certified radius on every 20th image from the CIFAR-10 test set, for 500 images total.

G.2. ImageNet and CIFAR-10 Experiments

Our code is available at http://github.com/locuslab/smoothing.

In order to report certified radii in the original coordinates, we first added Gaussian noise, and then standardized the data. Specifically, in our PyTorch implementation, the first layer of the base classifier was a normalization layer that performed a channel-wise standardization of its input. For CIFAR-10 we subtracted the dataset mean \((0.4914, 0.4822, 0.4465)\) and divided by the dataset standard deviation \((0.2023, 0.1994, 0.2010)\). For ImageNet we subtracted the dataset mean \((0.485, 0.456, 0.406)\) and divided by the standard deviation \((0.229, 0.224, 0.225)\).

For both ImageNet and CIFAR-10, we trained the base classifier with random horizontal flips and random crops (in addition to the Gaussian data augmentation discussed explicitly in the paper). On ImageNet we trained with synchronous SGD on four NVIDIA RTX 2080 Ti GPUs; training took approximately three days.

On ImageNet our base classifier used the ResNet-50 architecture provided in torchvision. On CIFAR-10 we used a 110-layer residual network from https://github.com/bearpaw/pytorch-classification.

On ImageNet we computed the certified radius on every 100-th image in the validation set, for 500 images total. On CIFAR-10 we computed the certified radius on every 20-th image in the test set, for 500 images total.

In Figure 7, we fixed \( \alpha = 0.001 \) and \( n = 100,000 \) while varying \( \sigma \), which was both the noise level of Gaussian data augmentation during training and the noise level of randomized smoothing prediction.

In Figure 6b we fixed \( \sigma = 0.25 \) and \( \alpha = 0.001 \) while varying the number of samples \( n \). We did not actually vary the number of samples \( n \) that we simulated: we kept this number fixed at 100,000 but varied the number that we fed the Clopper-Pearson confidence interval.

In Figure 6c, we fixed \( \sigma = 0.25 \) and \( n = 100,000 \) while varying \( \alpha \).

G.3. Adversarial Attacks

As discussed in Section 4, we subjected smoothed classifiers to a projected gradient descent-style adversarial attack. We now describe the details of this attack.

Let \( f \) be the base classifier and let \( \sigma \) be the noise level. Following Li et al. (2018), given an example \((x, c) \in \mathbb{R}^d \times \mathcal{Y}\) and a radius \( r \), we used a projected gradient descent style adversarial attack to optimize the objective:

\[
\arg \max_{\delta: \|\delta\|_2 \leq r} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} \left[ \ell(f(x + \delta + \epsilon), c) \right]
\]  

(37)

where \( \ell \) is the softmax loss function. (Breaking notation with the rest of the paper in which \( f \) returns a class, the function \( f \) here refers to the function that maps an image in \( \mathbb{R}^d \) to a vector of classwise scores.)

At each iteration of the attack, we drew \( k \) samples of noise, \( \epsilon_1 \ldots \epsilon_k \sim \mathcal{N}(0, \sigma^2 I) \), and followed the stochastic gradient

\[
g_t = \sum_{i=1}^{k} \nabla_{\delta_i} \ell(f(x + \delta_i + \epsilon_k), c).
\]

As is typical (Kolter & Madry, 2018), we used a “steepest ascent” update rule, which, for the \( \ell_2 \) norm, means that we normalized the gradient before updating the attack. The overall PGD update is:

\[
\delta_{t+1} = \text{proj}_r \left( \delta_t + \eta \frac{g_t}{\|g_t\|} \right)
\]

where the function \( \text{proj}_r \) that projects its input onto the ball \( \{ z : \|z\|_2 \leq r \} \) is given by \( \text{proj}_r(z) = \frac{rz}{\max(r, \|z\|_2)} \). We used a constant step size \( \eta \) and a fixed number \( T \) of PGD iterations.

In practice, our step size was \( \eta = 0.1 \), we used \( T = 20 \) steps of PGD, and we computed the stochastic gradient using \( k = 1000 \) Monte Carlo samples.

Unfortunately, the objective we optimize (37) is not actually the objective of interest. The real goal of an attacker is to find
some perturbation $\delta$ with $\|\delta\|_2 \leq r$ and some class $c_B$ for which

$$P_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)}(f(x + \delta + \epsilon) = c_B) \geq P_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)}(f(x + \delta + \epsilon) = c)$$

Effective adversarial attacks against randomized smoothing are outside the scope of this paper.
H. Examples of Noisy Images

We now show examples of CIFAR-10 and ImageNet images corrupted with varying levels of noise.

\[
\sigma = 0.00 \quad \sigma = 0.25 \quad \sigma = 0.50 \quad \sigma = 1.00
\]

*Figure 13. CIFAR-10 images additively corrupted by varying levels of Gaussian noise $\mathcal{N}(0, \sigma^2 I)$. Pixel values greater than 1.0 (=255) or less than 0.0 (=0) were clipped to 1.0 or 0.0.*
Certified Adversarial Robustness via Randomized Smoothing

Figure 14. ImageNet images additively corrupted by varying levels of Gaussian noise $N(0, \sigma^2 I)$. Pixel values greater than 1.0 (≈255) or less than 0.0 (≈0) were clipped to 1.0 or 0.0.