Certified Adversarial Robustness via Anisotropic Randomized Smoothing

Hanbin Hong [*1] **Yuan Hong** [*1]

Abstract

Randomized smoothing has achieved great success for certified robustness against adversarial perturbations. Given any arbitrary classifier, randomized smoothing can guarantee the classifier's prediction over the perturbed input with provable robustness bound by injecting noise into the classifier. However, all of the existing methods rely on fixed i.i.d. probability distribution to generate noise for all dimensions of the data (e.g., all the pixels in an image), which ignores the heterogeneity of inputs and data dimensions. Thus, existing randomized smoothing methods cannot provide optimal protection for all the inputs. To address this limitation, we propose the first anisotropic randomized smoothing method which ensures provable robustness guarantee based on pixel-wise noise distributions. Also, we design a novel CNN-based noise generator to efficiently fine-tune the pixel-wise noise distributions for all the pixels in each input. Experimental results demonstrate that our method significantly outperforms the state-of-the-art randomized smoothing methods.

1. Introduction

Deep learning (DL) models have been proven to be vulnerable to well-crafted adversarial examples (Goodfellow et al., 2015; Carlini & Wagner, 2017). For example, adversaries can generate minor malicious perturbations either with or without access to the DL models (in white-box or blackbox settings). Once injected into the input of the DL models, it could trigger misclassification or misrecognition. These successful adversarial attacks are detrimental to DL models in real-world deployments and may cause severe consequences, e.g., car accidents in autonomous driving (Sun et al., 2020), misdiagnosis in the auto-diagnosis (Ma et al., 2021), and misrecognizing faces (Dong et al., 2019).

To protect the DL models against adversarial attacks, empirical defense methods have been proposed in the past decade. Through training more robust models by including adversarial examples in the training set (Madry et al., 2018; Shafahi et al., 2019), destroying the malicious perturbation (Xu et al., 2017; Xie et al., 2018) or regularizing the features (Xie et al., 2019a; Yang et al., 2021), these empirical methods have shown effective defenses against the adversarial attacks. However, given any new powerful defense method, stronger attacks (Athalye et al., 2018; Croce & Hein, 2020; Xie et al., 2019b) will be designed to break the defenses. None of the empirical defenses can fully ensure the robustness of DL models all the time. Recently, certified robustness methods (Wong & Kolter, 2018; Cohen et al., 2019; Lecuyer et al., 2019) were proposed to provide provable guarantees on the robustness of the DL models. They aim to certify whether potential adversarial perturbations can result in misclassification or not. Once certified, it guarantees that any perturbation cannot fool the classifier if it is within a boundary. Typically, this boundary is given by an ℓ_p-norm ball, e.g., ℓ_1, ℓ_2, or ℓ_∞.

The randomized smoothing (RS) methods (Lecuyer et al., 2019; Teng et al., 2020; Cohen et al., 2019) provide certified robustness on any arbitrary classifiers (compared to traditional certified methods on specific classifiers, e.g, ReLU based neural networks). By injecting the noises to the input, RS turns any arbitrary classifier into a smoothed classifier, then the robustness of the smoothed classifier can be guaranteed if the perturbation is within a theoretical bound in ℓ_p-norm, i.e., *certified radius*. For example, (Cohen et al., 2019) derives a tight ℓ_2 certified radius for Gaussian noise. However, existing RS theories (Cohen et al., 2019; Teng et al., 2020; Yang et al., 2020; Zhang et al., 2020) can only derive the certified radii for fixed i.i.d. noises, e.g., Gaussian noise (Cohen et al., 2019) or Laplace noise (Teng et al., 2020), which applies identical distribution to different pixels and inputs. Thus, existing methods ignore the heterogeneity of the inputs and data dimensions, and cannot provide optimal protection for all the inputs.

To pursue optimal protection for every input, we propose the first randomized smoothing theory for anisotropic noise (to our best knowledge), which applies different distributions to generate noise for different data dimensions (e.g., image pixels). In this paper, we consider Gaussian noise as a use case to introduce anisotropic randomized smoothing (*other noise can also achieve it with similar theories* as discussed in Section 7). We also propose a Noise Generator to generate the pixel-wise noise distributions for all the pixels in each input. Specifically, a tight certified radius

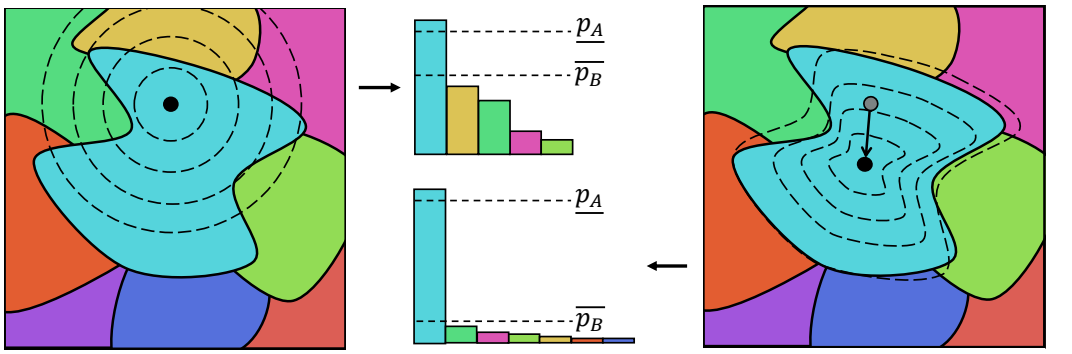

Figure 1. Evaluation of the smoothed classifier at an input x. The decision regions of the base classifier f are represented in different colors. The dashed lines are the level sets of the noise distribution adding to the input. The left figure illustrates the randomized smoothing with isotropic Gaussian noise $\mathcal{N}(0, \sigma^2\mathbf{I})$ in (Cohen et al., 2019) whereas the right figure illustrates the randomized smoothing with anisotropic Gaussian noise $\mathcal{N}(\mu, \mathbf{\Sigma})$. The middle figure shows the prediction probabilities of the input over the noises.

is derived in our theory when all the pixels are smoothed by Gaussian noise with different means and variances. The Noise Generator uses a convolution neural network (CNN) to efficiently fine-tune the noise mean and variance for each pixel in randomized smoothing.

Compared to the traditional RS methods (Cohen et al., 2019; Teng et al., 2020; Zhang et al., 2020), our certified defense provides the following new significant benefits:

- **Higher Certified Accuracy**. We train the Noise Generator to generate the optimal means for the noises to be added to the input. The noise with proper means can move the input representation to the center of its class, e.g., some input located near the decision boundary can be adjusted to the class representation center by adding the noise mean towards the center (see Figure 1 for illustration). This improves the certified accuracy for as high as 32.9% on CIFAR10 and 20.6% on ImageNet.
- **Larger Certified Radii**. We train the Noise Generator to also generate the optimal variance for the noises to be added to the input. Different from the isotropic Gaussian noise, we generate different variances for different pixels to keep the noisy sample within the decision boundary (see Figure 1 for illustration). Thanks to the optimal means and variances, our smoothed classifier maintains a higher prediction accuracy over the same noise than the traditional smoothed classifier, which leads to larger certified radii. When certified accuracy is fixed at 20%, the certified radius can be improved from 1.10 to 2.96 on CIFAR10 and from 1.92 to 3.73 on ImageNet.
- **Enhanced Robustness against Pre-Perturbing Attack**. We also study a new problem in randomized smoothing: what happens if the input is perturbed before certification with noise? Indeed, if the input is maliciously perturbed before injecting the noise for certification, the smoothed classifier's prediction could be guaranteed to be consistently wrong (certifying the class label of the perturbed input). We show that our method is more robust than the traditional RS methods against such adversarial attacks.

2. Related Work

In this paper, all the defense methods are proposed against the evasion attacks to machine learning models. They aim to make the model correctly predict results on perturbed inputs. Typically, there are two types of defense methods: empirical defenses and certified defenses. The empirical defenses empirically protect the models while the certified defenses ensure the robustness of the models with provable guarantees.

Empirical Defenses. In the past decade, empirical defenses have been proposed to protect the machine learning models in different ways, e.g., training more robust models by including adversarial examples in the training data (Madry et al., 2018; Shafahi et al., 2019; Tramer et al., 2018; Wong et al., 2019), pre-processing the inputs to destroy the malicious patterns in the perturbation (Liu et al., 2019; Xu et al., 2017; Xie et al., 2019a; Samangouei et al., 2018), regularizing the features in the model to eliminate the effects of perturbations (Xie et al., 2018; Yang et al., 2021) or detecting the adversarial examples before fed into the model (Lu et al., 2017; Metzen et al., 2017; Lee et al., 2018). Although empirical evidence has shown that these methods can efficiently defend against adversarial attacks, none of them can guarantee model robustness against adversarial attacks.

Certified Defenses. The certified defenses were proposed

to guarantee robustness against adversarial perturbations. In general, the robustness can be guaranteed if the perturbations are within a boundary, e.g., a ℓ_1 , ℓ_2 or ℓ_∞ ball of radius R. The existing certified defenses can be roughly divided into two categories: exact certified defenses and conservative certified defenses. The exact certified defenses usually leverage satisfiability modulo theories (Katz et al., 2017; Carlini et al., 2017; Ehlers, 2017; Huang et al., 2017b) or mixed-integer linear programming (Cheng et al., 2017; Lomuscio & Maganti, 2017; Fischetti & Jo, 2018; Bunel et al., 2018) to guarantee whether there exists a perturbation within radius R or not. The conservative certified defenses provide conservative guarantee on the robustness by global/local Lipschitz constant methods (Gouk et al., 2021; Tsuzuku et al., 2018; Anil et al., 2019; Cissé et al., 2017; Hein & Andriushchenko, 2017), optimization methods (Wong & Kolter, 2018; Wong et al., 2018; Raghunathan et al., 2018; Dvijotham et al., 2018) or layer-by-layer certifying (Mirman et al., 2018; Singh et al., 2018; Gowal et al., 2018; Weng et al., 2018; Zhang et al., 2018a). In certain circumstances, it cannot provide the guarantee even when the malicious perturbation exists. However, the exact certified defenses cannot be scaled to large-size networks, and the conservative certified defenses usually assume specific types of networks, e.g., ReLU based networks. None of these schemes can provide certified robustness to any arbitrary classifiers until the randomized smoothing was proposed.

Randomized Smoothing. The randomized smoothing was first studied by Lecuyer et al. (Lecuyer et al., 2019), where a loose theoretical bound for the perturbation is derived using Differential Privacy methods (Dwork, 2006; 2008). The first tight guarantee was proposed by Cohen et al. (Cohen et al., 2019), in which, any arbitrary classifier can be turned into a smoothed classifier by adding Gaussian noise to the data. The smoothed classifier's prediction can be guaranteed to be consistent within a certified radius in ℓ_2-norm, which is tightly derived. Following the track of randomized smoothing, a series of methods have been proposed to guarantee the robustness against different ℓ_p perturbations with different noise distributions, e.g., Teng et al. (Teng et al., 2020) derives the certified radius for ℓ_1 perturbations with Laplace noise, and Lee et al. (Lee et al., 2019) derives the certified radius against ℓ_0 perturbations with uniform noise. Some methods propose unified theories to guarantee the robustness against a diverse set of ℓ_p perturbations with different noises. For example, Zhang et al. (Zhang et al., 2020) propose a framework from the optimization perspective to certify the robustness against ℓ_1, ℓ_2 and ℓ_∞ perturbations with special noise distributions. Yang et al. (Yang et al., 2020) propose two different methods, e.g., level set method and differential methods, that can derive the upper bound and the lower bound of the certified radius in different norms for a wide range of distributions. However, all the existing randomized smoothing methods add noise drawn from a fixed distribution, e.g., Gaussian or Laplace, to all the inputs and all dimensions of each input (e.g., all the pixels on an image). This ignores the heterogeneity of inputs and even the pixels. Thus, they cannot provide the optimal protection for every input and pixel.

Therefore, we establish the first randomized smoothing method based on anisotropic noise.

3. Isotropic Randomized Smoothing

We first review the randomized smoothing with isotropic Gaussian noise (Cohen et al., 2019).

We study the classification from $\mathbb{R}^d$ to classes $\mathcal{Y}$. Given an arbitrary base classifier f, randomized smoothing is a method that can turn the base classifier into a "smoothed" classifier g by injecting noise into the input. The smoothed classifier predicts the top-1 class w.r.t. to the input x over the noise. The randomized smoothing in Cohen et al. (Cohen et al., 2019) is formally defined as:

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x+\epsilon) = c), \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \quad (1)$$

The injected noise ϵ follows an independent isotropic Gaussian noise $\mathcal{N}(0, \sigma^2\mathbf{I})$. Also, the mean of the Gaussian noise is set to be 0. Thus, the randomized smoothing proposed by (Cohen et al., 2019) adds noise to all the dimensions of the inputs with identical variance and zero-mean.

Based on the smoothed classifier g, the top-1 class is denoted as $c_A \in \mathcal{Y}$, the second probable class is denoted as $c_B \in \mathcal{Y}$, and the corresponding lower bound and upper bound of the class probabilities are denoted as $\underline{p_A}$ and $\overline{p_B}$. Cohen et al. (Cohen et al., 2019) derives the first tight bound of the certified radius with the isotropic Gaussian noise in Theorem 3.1.

Theorem 3.1 (Randomized Smoothing with Isotropic Gaussian Noise (Cohen et al., 2019)). *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. Define g as in Eq. (1). For a specific $x \in \mathbb{R}^d$, there exist $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0, 1]$ such that:*

$$\mathbb{P}(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\epsilon) = c) \quad (2)$$

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

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \quad (3)$$

where δ denotes the perturbation.

Proof. See detailed proof in (Cohen et al., 2019). □

Theorem 3.1 guarantees that the smoothed classifier will consistently predict the most probable class when the perturbation is within the radius defined in Eq. (3) if p_A and p_B satisfy the condition (2) in Theorem 3.1.

While certifying any arbitrary classifier, isotropic randomized smoothing applies an identical distribution to generate the noise for all the dimensions, which may limit the defense performance on all the pixels of different inputs. Thus, it is desirable to extend the randomized smoothing to add heterogeneous noise for different pixels (anisotropic).

However, there are two challenges on extending the isotropic RS to anisotropic RS. First, anisotropic RS needs more complicated theories on deriving the certified radius since the noise follows different distributions on different dimensions. Second, instead of simply adding noise with the same distribution to all the pixels, we need to fine-tune the noise distribution for each pixel in the anisotropic RS. To address these challenges, we propose a novel theory on anisotropic RS in Section 4 and design a novel mechanism for finding the optimal noise distribution for all the pixels in Section 5.

4. Anisotropic Randomized Smoothing

In this section, we propose the first anisotropic randomized smoothing (ARS) theory with the tight certified radius. We take the Gaussian noise as an example to illustrate how to extend the isotropic randomized smoothing to anisotropic randomized smoothing, while other noise can be also readily extended for ARS using similar procedures. Specifically, we also theoretically derive the certified radius for anisotropic Laplace noise in Section 7 as an extension.

For the Gaussian noise, we first extend the smoothed classifier in Eq. (4), and then provide a tight guarantee on its robustness with the anisotropic noise in Theorem 4.1.

$$g'(x) = \arg\max_{c\in\mathcal{Y}} \mathbb{P}(f(x+\epsilon)=c),\ \epsilon\sim\mathcal{N}(\mu,\mathbf{\Sigma}) \tag{4}$$

where the mean of the Gaussian noise is defined as $\mu = [\mu_1, \mu_2, ..., \mu_d]$, and the variance of the Gaussian noise is defined as $\mathbf{\Sigma} = diag(\sigma_1^2, \sigma_2^2, ..., \sigma_d^2)$.

Theorem 4.1 (Randomized Smoothing with Anisotropic Gaussian Noise). *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\epsilon \sim \mathcal{N}(\mu, \mathbf{\Sigma})$. Let g' be defined as in Eq. (4). Suppose that for a specific $x \in \mathbb{R}^d$, there exist $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ such that:*

$$\mathbb{P}(f(x+\epsilon)=c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c\neq c_A} \mathbb{P}(f(x+\epsilon)=c) \tag{5}$$

Then $g'(x+\delta) = c_A$ for all $||\delta||_2 < R$, where

$$R = \frac{1}{2}\min\{\sigma_i\}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{6}$$

where σ_i denotes the variance on i-th dimension of the input, δ denotes the perturbation.

Proof. See detailed proof in Appendix A. □

Indeed, Theorem 4.1 can be considered as a generalized form of Theorem 3.1 since when $\mathbf{\Sigma} = diag(\sigma^2, \sigma^2, ..., \sigma^2)$ and $\mu = \mathbf{0}$, we have $\min\{\sigma_i\} = \sigma$. Thus, our theorem returns the same certified radius as Theorem 3.1 in the same setting.

In Theorem 4.1, we observe that the certified radius only depends on the minimum variance over all the dimensions. Thus, any larger variance in other dimensions would not affect the certified radius. We will show that this provides benefits on defending against the attacks to the randomized smoothing in Section 6. When we certify the perturbed inputs, the large noise will also smooth the perturbation so that the adversarial effects can be reduced.

We also observe that the certified radius does not depend on the mean of the Gaussian noise. However, a proper mean of the noise may affect the smoothed classifier's prediction on the clean input, and further improve the certified radius since it affects the p_A and p_B. We will show how to design a mechanism to find a proper mean (besides the variances) for the noise to improve the certified robustness.

We also present the binary case of Theorem 4.1 as below:

Theorem 4.2 (Binary Case). *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\epsilon \sim \mathcal{N}(\mu, \mathbf{\Sigma})$. Let g' be defined as in Eq. (4). Suppose that for a specific $x \in \mathbb{R}^d$, there exist $c_A \in \mathcal{Y}$ and $\underline{p_A}$ such that:*

$$\mathbb{P}(f(x+\epsilon)=c_A) \geq \underline{p_A} \geq \frac{1}{2} \tag{7}$$

Then $g'(x+\delta) = c_A$ for all $||\delta||_2 < R$, where

$$R = \min\{\sigma_i\}\Phi^{-1}(\underline{p_A}) \tag{8}$$

Proof. See detailed proof in Appendix B. □

5. Noise Generator

Our proposed ARS theory could certify the defenses for randomized smoothing based on applying different means and different variances to generate noise for different pixels. However, how to design a new mechanism to fine-tune variance and mean for the noise distributions is a challenging problem. Note that the optimal variances and means can be different from input to input and from pixel to pixel. Therefore, we leverage the CNNs to design a Noise Generator for learning the mapping from the input to the optimal variances and means, and generating the input-dependent variances

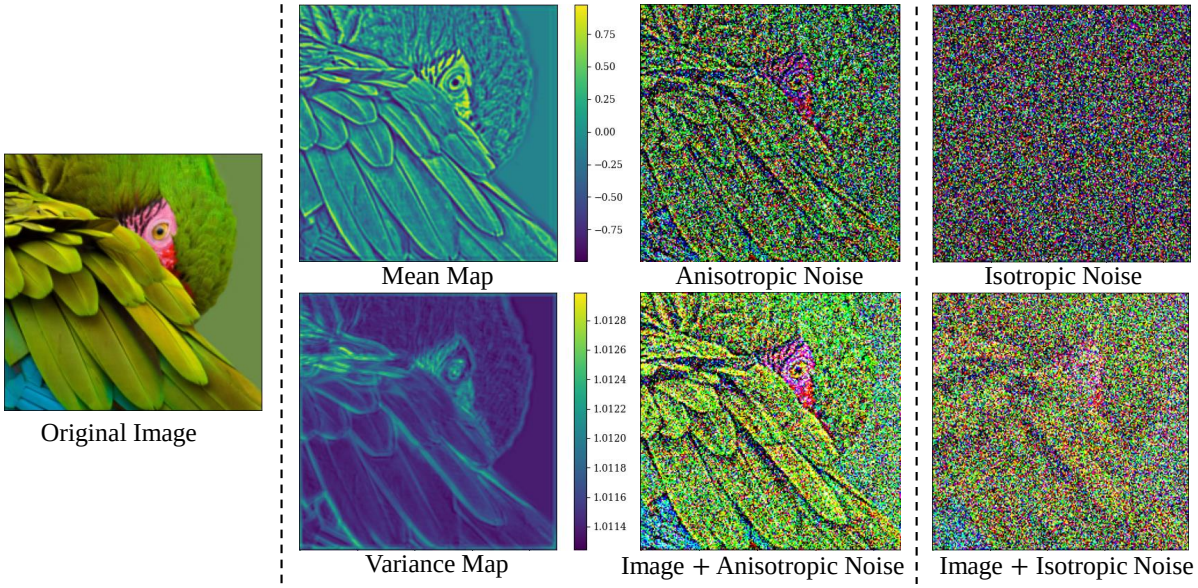

Figure 2. An example of anisotropic and isotropic noise. **Left**: The original image. **Middle**: The pixel-wise means and variances for anisotropic Gaussian distribution generated by our Noise Generator, and the noise sample. **Right**: The noise sample generated with isotropic Gaussian distribution of $\sigma = 1.0$.

and means for the certification (see Figure 2 for an example of comparing the anisotropic and isotropic noises).

Specifically, in the training and the certification, the Noise Generator takes the image as input and returns a variance map as well as a mean map for the randomized smoothing. Then, the base classifier will take the noisy images as the input for training or classification. The framework is summarized in Figure 3.

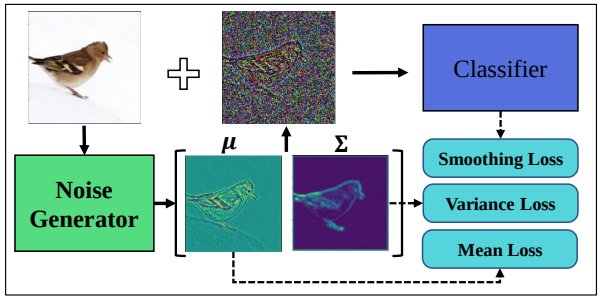

Figure 3. **Framework**. The noise generated by Noise Generator will be added to the image for smoothed classifier training and classification. We train the Noise Generator and the classifier simultaneously with three losses.

Architecture. The Noise Generator learns the mapping from the image to the variance and mean maps, which is similar to the function of the neural networks in image transformation. Therefore, inspired by the image super-resolution work (Zhang et al., 2018b), we also use the "dense blocks" (Huang et al., 2017a) as the main architecture. It consists of 4 convolution layers followed by leaky-ReLU (Xu et al., 2015). In addition, some special designs are integrated into the Noise Generator to fit our tasks (See Figure 4). First, the output of the dense block is separated into two branches to generate the mean map and variance map. Second, a sigmoid layer is inserted before the mean and variance maps to constrain the mean and variance values. Otherwise, the training may not converge due to some extreme values in the noise. Note that, our Noise Generator is a small network (5-layer deep), so it can be plugged before any classifier for customizing the noise distribution without consuming too much computing resources (See Section 7 for detailed discussion on running time).

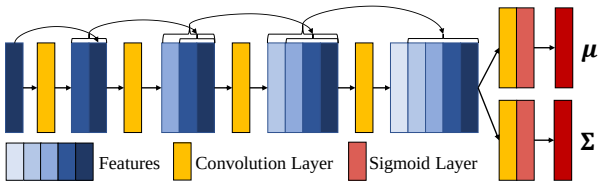

Figure 4. **Architecture of Noise Generator**.

Loss Functions. We train the Noise Generator and the base classifier simultaneously. In the training, our goal is to generate the optimal mean and variance maps. Specifically, by fine-tuning the mean and variance, we aim to train the classifier to predict the noisy image as accurately as possible (large p_A). Thus, we use the smoothing loss (Eq. 9) for

training both the Noise Generator and the classifier.

$$\mathcal{L}_s = -\sum_{k=1}^{N} y_k \; log[\hat{y}_k(x+\epsilon, \theta_f, \theta_g)] \tag{9}$$

where $y_k = 1$ if the class k is the correct label of input x, otherwise $y_k = 0$. $\hat{y}_k$ denotes the prediction of the base classifier f on the input x perturbed by noise ϵ. θ_f and θ_g denote the model parameters of classifier f and Noise Generator, respectively.

Also, since the certified radius only depends on the minimum variance, we only have constraints on the minimum values in the variance map. Large variances can improve the certified radius while they will degrade the prediction accuracy since large noises may greatly distort the image. It has been widely known that the variance tunes the trade-off between the accuracy and the certified radius (Cohen et al., 2019; Yang et al., 2020). In our case, it is the minimum variance that tunes the trade-off. Therefore, similar to recent works (Cohen et al., 2019), we constrain the minimum variance to a certain level by the variance loss (Eq. 10).

$$\mathcal{L}_v = \left| \frac{\min\{\sigma_i(x, \theta_g)\} - \sigma_0}{\sigma_0} \right| \tag{10}$$

where the $\sigma_i(x)$ is the variance for dimension i of the input x. σ_0 denotes the variance target that the minimum variance is trained to achieve. By minimizing the variance loss, we aim to train the Noise Generator to generate a variance map with the minimum value $\min\{\sigma\}_i = \sigma_0$.

We also constrain the mean map generation with the mean loss (Eq. (11)) by considering this: although the mean of the noise will not affect the certified radius and can help align the input to the representation center of its class, an extremely large value of mean can distort the image which would harm the prediction accuracy. Thus, the mean map should be as small as possible.

$$\mathcal{L}_m = ||\mu(x, \theta_g)||_2 \tag{11}$$

The training process is to minimize the total loss in Eq. (12)

$$\min_{\theta_f, \theta_g} \quad \mathbb{E}_{x\sim\mathbb{D}, \epsilon\sim\mathcal{N}(\mu,\mathbf{\Sigma})}[\alpha\mathcal{L}_s + \beta\mathcal{L}_v + \gamma\mathcal{L}_m] \tag{12}$$

where α, β, and γ are the weights of the three loss functions, and $\mathbb{D}$ denotes the dataset.

Practical Algorithms. We follow Cohen et al. (Cohen et al., 2019) to use the Monte Carlo algorithm for evaluating $g(x)$ and compute the certified robustness. Different from Cohen et al. (Cohen et al., 2019), our noise distributions are generated by Noise Generator for each input. Our algorithms for certification and prediction in binary case are presented in Algorithm 1 and 2 in Appendix C, respectively.

Specifically, in the certification (Algorithm 1), the mean and variance for the noise are generated by Noise Generator. Then, we select the top-1 class $\hat{c_A}$ by the $ClassifySamples$ function, in which the base classifier outputs the predictions on the noisy input sampled from the noise distribution. Once the top-1 class is determined, classification will be run on more samples and the $LowerConfBound$ function will output the lower bound of the probability $\underline{p_A}$ computed by the Binomial test. If $\underline{p_A} > \frac{1}{2}$, we output the prediction class and the certified radius. Otherwise, it outputs ABSTAIN. In the prediction (Algorithm 2), we also generate the noise distribution and then compute the prediction counts over the noisy inputs. If the Binomial test succeeds, then it outputs the prediction class. Otherwise, it returns ABSTAIN.

6. Experiments

We evaluate the performance of certified robustness in this section. In addition, we evaluate the enhanced robustness of the randomized smoothing with anisotropic noises.

Metrics. Following (Cohen et al., 2019), we use the *approximate certified test set accuracy* to measure the certified robustness, which is defined as the fraction of the test set that is certified to be consistently correct within the certified radius R. See Eq. (13) for the formal definition.

$$Acc(R) = \frac{\sum_{j=1}^{N} \mathbf{1}_{[g'(x^j+\delta)=y^j]}}{N} \quad \text{for all } ||\delta||_2 \leq R \tag{13}$$

where x^j and y^j denote the j-th sample and its label in the test set. N denotes the number of images in the test set.

Experimental Settings. We evaluate our method on the CIFAR10 (Krizhevsky et al., 2009) and ImageNet datasets (Russakovsky et al., 2015). We use the original size of the images in CIFAR10, i.e., $3 \times 32 \times 32$, while for ImageNet, we resize the images to $3 \times 224 \times 224$. In the training, we train the base classifier and the Noise Generator with all the training set in CIFAR10 and ImageNet. We use the ResNet110 and ResNet50 (He et al., 2016) as the base classifier for CIFAR10 and ImageNet, respectively. The training loss is computed over 5 samples from the noise distribution for each image. For the certification, following (Cohen et al., 2019), we evaluate the certified accuracy on the entire test set in CIFAR10 while randomly sample 500 samples in the test set of ImageNet. In the certification, we also follow (Cohen et al., 2019) to set $\alpha = 0.001$ and numbers of Monte Carlo samples $n_0 = 100$ and $n = 100,000$.

Experimental Environment. All the experiments were

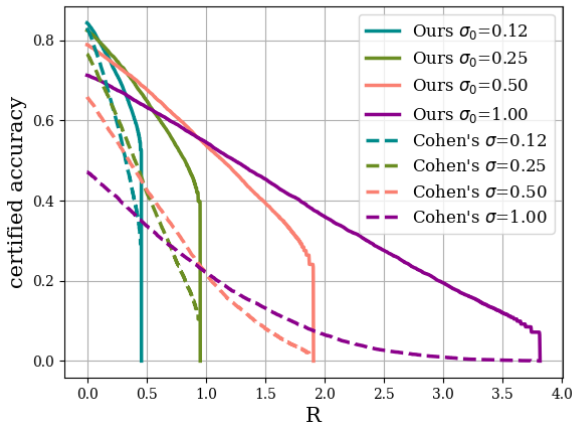

Figure 5. **Certified accuracy comparison on CIFAR10** .

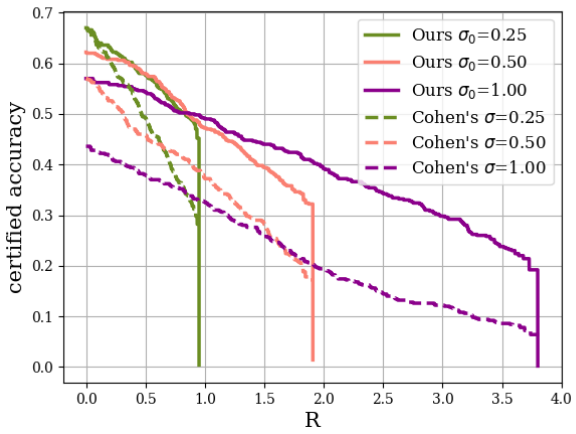

Figure 6. **Certified accuracy comparison on ImageNet** .

performed on the NSF Chameleon Cluster (Keahey et al., 2020) with Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz, 128G RAM, and Tesla V100 SXM2 32GB.

6.1. Certified Accuracy

We evaluate the certified accuracy on both CIFAR10 and ImageNet, and compare our ARS with the isotropic RS baseline (Cohen et al., 2019). In (Cohen et al., 2019), we present the certified accuracy computed with Gaussian noise. Following the setting in (Cohen et al., 2019), the noise variance is set as $\sigma = 0.12, 0.25, 0.5$, and 1.0 in CIFAR10 and $\sigma = 0.25, 0.5, 1.0$ in ImageNet. In our method, we also set the variance target σ_0 to be consistent with (Cohen et al., 2019).

We show the certified accuracy for different certified radii on CIFAR10 and ImageNet datasets in Fig. 5 and Fig. 6, respectively. We can observe that our certified accuracy is higher than the baseline's in the case of all the certified radii. In particular, when the variance is large, we observe a significant improvement in the certified accuracy using our method. This might be because our mean map in anisotropic Gaussian noise can bring the data representation to the center of the correct class and further improve the prediction probability on the noisy inputs (tuned by noise variance).

6.2. Best Performance Comparison

We compare our anisotropic randomized smoothing with the state-of-the-art randomized smoothing methods against ℓ_2 perturbations. Specifically, (Cohen et al., 2019) derives the first tight certified radius for Gaussian noise, which is shown to have the best performance of certified robustness over a wide range of distributions (Yang et al., 2020). Also, (Zhang et al., 2020) proposes an optimization-based randomized smoothing method with a special-designed distribution that outperforms the Gaussian noise. (Alfarra et al., 2020) proposes to optimize the variance of the noise distribution for each input to provide a data-dependent randomized smoothing. We compare our method with these state-of-the-art methods in Table 1 and 2.

Both on the CIFAR10 and the ImageNet, our method significantly improves the certified accuracy. For instance, we observe the best improvement of 29% at $R = 1.5$ on CIFAR10, and 9% at $R = 2.5$ on ImageNet. Different from the isotropic methods, when the certified radius is large, our method can still provide certified protection for the images.

6.3. Enhanced Robustness against Pre-Perturbation

We also study a new interesting problem for randomized smoothing methods which has not been discussed in existing works (Cohen et al., 2019; Zhang et al., 2020; Alfarra et al., 2020). In general, the randomized smoothing is applied to certify the clean images such that the prediction can be guaranteed to be correct if the clean image is perturbed within the certified radius. If the adversary crafts an adversarial example for the certification, the guaranteed prediction could be consistently wrong within the certified radius (consistent with the class label for the adversarial example), which contrarily enhances the attack performance instead of protecting the inputs. Therefore, it is important to build a "shield" for the randomized smoothing methods. Different from isotropic randomized smoothing methods, e.g., (Cohen et al., 2019), our generated mean map can re-calibrate the data point towards its correct label, which makes it harder to be attacked. In addition, our anisotropic randomized smoothing can leverage larger and more complicated noises to mitigate malicious perturbations.

We evaluate the performance of our method on defending against the strong white-box attacks, e.g., PGD attack (Madry et al., 2018) (pre-perturbing). Specifically, the image is perturbed by the PGD attack with max ℓ_∞ perturbation $16/255$ and 10 iterations before the certification, then we compute the certified accuracy on the clean image and

Radius	0.0	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50
Cohen's	83%	61%	43%	32%	22%	17%	14%	9%	7%	4%	3%	2%	1%	0	0
Zhang's	–	61%	46%	37%	25%	19%	16%	14%	11%	9%	–	–	–	–	–
Alfarra's	82%	68%	53%	44%	32%	21%	14%	8%	4%	1%	–	–	–	–	–
Ours	**84%**	**75%**	**68%**	**61%**	**55%**	**50%**	**45%**	**41%**	**36%**	**32%**	**28%**	**23%**	**19%**	**16%**	**12%**

Table 1. Certified Accuracy (%) on CIFAR10.

Radius	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
Cohen's	**67%**	49%	37%	28%	19%	15%	12%	9%
Zhang's	–	50%	39%	31%	21%	17%	13%	10%
Alfarra's	62%	**59%**	48%	43%	31%	25%	22%	19%
Ours	**67%**	58%	**49%**	**44%**	**39%**	**34%**	**30%**	**24%**

Table 2. Certified Accuracy (%) on ImageNet.

Radius	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
Cohen's	44%	38%	33%	26%	19%	15%	12%	9%
PGD	30%	24%	19%	14%	10%	7%	6%	5%
loss (%)	-31%	**-37%**	-42%	-46%	-47%	-53%	-50%	-44%
Ours	57%	54%	49%	44%	39%	34%	30%	24%
PGD	40%	33%	30%	27%	23%	20%	17%	14%
loss (%)	**-30%**	-39%	**-39%**	**-39%**	**-41%**	**-41%**	**-43%**	**-42%**

Table 3. Certified Accuracy (%) before and after the PGD attack (pre-perturbing the inputs before certification).

the adversarial image and present the loss of the certified accuracy. The variance σ and the target variance σ_0 are both set to 1.0. Other experimental settings are the same as Section 6.1.

The experimental results are shown in Table 3. Although the certified accuracy for both our method and (Cohen et al., 2019) are degraded by the pre-perturbing PGD attacks, our anisotropic randomized smoothing is still more robust (less degradation) than (Cohen et al., 2019) against such attacks in almost all the cases.

7. Discussions

7.1. Generalization to Other Noise Distributions against Different ℓ_p Perturbations

In this paper, we take the anisotropic Gaussian noise as an example for deriving the certified radii and designing the Noise Generator. In fact, anisotropic randomized smoothing and Noise Generator are general methods that can be used for different distributions against different perturbations. Here we show an extension of our theory to anisotropic Laplace noise against ℓ_1 perturbations in Theorem 7.1.

Theorem 7.1 (Randomized Smoothing with Anisotropic Laplace Noise). *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\epsilon \sim \mathcal{L}(\mu, \mathbf{\Lambda})$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_d)$. Let g'' be defined as $g''(x) = \arg\max_{c\in\mathcal{Y}} \mathbb{P}(f(x+\epsilon) = c)$. Suppose that for a specific $x \in \mathbb{R}^d$, there exist $c_A \in \mathcal{Y}$ and $\underline{p_A}$, $\overline{p_B} \in [0, 1]$ such that:*

$$\mathbb{P}(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c\neq c_A} \mathbb{P}(f(x+\epsilon) = c) \quad (14)$$

Then $g''(x+\delta) = c_A$ for all $||\delta||_1 < R$, where

$$R = \max\{\frac{1}{2}\min\{\lambda_i\}\log(\underline{p_A}/\overline{p_B}), \\ -\min\{\lambda_i\}\log(1-\underline{p_A}+\overline{p_B})\} \quad (15)$$

where λ_i is the variance on i-th dimension of the input.

Proof. See detailed proof in Appendix D. □

Similarly, for other noise distributions, e.g., Exponential, Uniform, or Pareto distributions (Yang et al., 2020), we can derive the certified radius with anisotropic noise using the same method. In addition, once the certified radius is derived, our Noise Generator can be used for generating the parameters for any anisotropic noise distribution.

7.2. Runtime

Our anisotropic randomized smoothing relies on the Noise Generator to provide optimal protection, which may need extra runtime for generating the mean and variance than traditional RS methods. However, the extra runtime resulting from the Noise Generator is actually negligible. Our model (including Noise Generator) can be trained offline and tested online as traditional RS methods. We evaluate the online certification runtime for our method and (Cohen et al., 2019) on ImageNet with four Tesla V100 GPUs and $2,000$ batch size, the average runtimes over 500 samples are 27.43s and 27.09s per sample for our method and Cohen et al.'s method, respectively. Thus, the Noise Generator will not affect the overall runtime of the certification much.

8. Conclusion

We study a new direction of randomized smoothing: the anisotropy of the distributions. Facilitated by the proposed Noise Generator, our anisotropic randomized smoothing significantly improves the certified robustness, which has been extensively evaluated on CIFA10 and ImageNet datasets. The anisotropic randomized smoothing and the Noise Generator can be adapted to various distributions for further improving the certified robustness in the future.

References

Alfarra, M., Bibi, A., Torr, P. H., and Ghanem, B. Data dependent randomized smoothing. *arXiv preprint arXiv:2012.04351*, 2020.

Anil, C., Lucas, J., and Grosse, R. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pp. 291–301. PMLR, 2019.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Bunel, R. R., Turkaslan, I., Torr, P. H., Kohli, P., and Mudigonda, P. K. A unified view of piecewise linear neural network verification. In *NeurIPS*, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

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

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

Cissé, M., Bojanowski, P., Grave, E., Dauphin, Y. N., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 854–863. PMLR, 2017.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

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

Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., and Zhu, J. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7714–7722, 2019.

Dvijotham, K., Stanforth, R., Gowal, S., Mann, T. A., and Kohli, P. A dual approach to scalable verification of deep networks. In *UAI*, volume 1, pp. 3, 2018.

Dwork, C. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pp. 1–12. Springer, 2006.

Dwork, C. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pp. 1–19. Springer, 2008.

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

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

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. J. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T. A., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018. URL http://arxiv.org/abs/1810.12715.

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

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 2266–2276, 2017.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017a.

Huang, X., Kwiatkowska, M., Wang, S., and Wu, M. Safety verification of deep neural networks. In *International conference on computer aided verification*, pp. 3–29. Springer, 2017b.

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

Keahey, K., Anderson, J., Zhen, Z., Riteau, P., Ruth, P., Stanzione, D., Cevik, M., Colleran, J., Gunawi, H. S., Hammock, C., Mambretti, J., Barnes, A., Halbach, F., Rocha, A., and Stubbs, J. Lessons learned from the chameleon testbed. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*. USENIX Association, July 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

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

Lee, G., Yuan, Y., Chang, S., and Jaakkola, T. S. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *NeurIPS*, pp. 4911–4922, 2019.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Liu, Z., Liu, Q., Liu, T., Xu, N., Lin, X., Wang, Y., and Wen, W. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 860–868. IEEE, 2019.

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

Lu, J., Issaranon, T., and Forsyth, D. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 446–454, 2017.

Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J., and Lu, F. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332, 2021.

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.

Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

Mirman, M., Gehr, T., and Vechev, M. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pp. 3578–3586. PMLR, 2018.

Neyman, J. and Pearson, E. S. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 3358–3369, 2019.

Singh, G., Gehr, T., Mirman, M., Püschel, M., and Vechev, M. Fast and effective robustness certification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 10825–10836, 2018.

Sun, J., Cao, Y., Chen, Q. A., and Mao, Z. M. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pp. 877–894, 2020.

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

Tramer, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. *stat*, 1050:22, 2018.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *NeurIPS*, 2018.

Weng, T., Zhang, H., Chen, H., Song, Z., Hsieh, C., Daniel, L., Boning, D. S., and Dhillon, I. S. Towards fast computation of certified robustness for relu networks. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*,

volume 80 of *Proceedings of Machine Learning Research*, pp. 5273–5282. PMLR, 2018.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, 2018.

Wong, E., Schmidt, F. R., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. *arXiv preprint arXiv:1805.12514*, 2018.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2019.

Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., and He, K. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019a.

Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., and Yuille, A. L. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, 2019b.

Xu, B., Wang, N., Chen, T., and Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I., and Li, J. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pp. 10693–10705. PMLR, 2020.

Yang, S., Guo, T., Wang, Y., and Xu, C. Adversarial robustness through disentangled representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3145–3153, 2021.

Zhang, D., Ye, M., Gong, C., Zhu, Z., and Liu, Q. Black-box certification with randomized smoothing: A functional optimization based framework. 2020.

Zhang, H., Weng, T., Chen, P., Hsieh, C., and Daniel, L. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 4944–4953, 2018a.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2472–2481, 2018b.

A. Proofs for Theorem 4.1

We prove the Theorem 4.1 in this section. Similar to Cohen et al. (Cohen et al., 2019), Theorem 4.1 is based on Neyman-Pearson lemma (Neyman & Pearson, 1933). Therefore, we will review the Neyman-Pearson lemma and then derive the certified radius for anisotropic Gaussian noise.

Lemma A.1 (**Neyman-Pearson** (Neyman & Pearson, 1933))**.** *Let X and Y be random variables in $\mathbb{R}^d$ with probability density functions (PDF) f_X and f_Y. Let $h : \mathbb{R}^d \to \{0, 1\}$ be a random or deterministic function. Then:*

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

(2) If $S = \{z \in \mathbb{R}^d : \frac{f_Y(z)}{f_X(z)} \geq t\}$ for some $t > 0$ 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. See the detailed proof in Cohen et al. (Cohen et al., 2019) □

Then, we prove the special case of Lemma A.1 when the random variables follows independent anisotropic Gaussian distribution.

Lemma A.2. *Let $X \sim \mathcal{N}(x + \mu, \mathbf{\Sigma})$ and $Y \sim \mathcal{N}(x + \mu + \delta, \mathbf{\Sigma})$, where $\delta = [\delta_1, \delta_2, ..., \delta_d]$, $\mu = [\mu_1, \mu_2, ..., \mu_d]$ and $\mathbf{\Sigma} = diag(\sigma_1^2, \sigma_2^2, ..., \sigma_d^2)$. Let $h : \mathbb{R}^d \to \{0, 1\}$ be any deterministic or random function. Then:*

(1) If $S = \{z \in \mathbb{R}^d : \sum_{i=1}^d \frac{\delta_i}{\sigma_i^2} z_i \leq \beta\}$ for some β and $\mathbb{P}(h(X) = 1) \geq \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) = 1) \geq \mathbb{P}(Y \in S)$

(2) If $S = \{z \in \mathbb{R}^d : \sum_{i=1}^d \frac{\delta_i}{\sigma_i^2} z_i \geq \beta\}$ for some β 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. Let $X \sim \mathcal{N}(x + \mu, \mathbf{\Sigma})$ and $Y \sim \mathcal{N}(x + \mu + \delta, \mathbf{\Sigma})$. We have the probability density functions f_X and f_Y as:

$$f_X(z) = k \exp\left(-\sum_{i=1}^d \frac{1}{2\sigma_i^2}[z_i - (x_i + \mu_i)]^2\right)$$

$$f_Y(z) = k \exp\left(-\sum_{i=1}^d \frac{1}{2\sigma_i^2}[z_i - (x_i + \mu_i + \delta_i)]^2\right)$$

where k is a constant. The ratio of the PDF is:

$$\begin{aligned}
\frac{f_Y(z)}{f_X(z)} &= \frac{\exp\left(-\sum_{i=1}^d \frac{1}{2\sigma_i^2}[z_i - (x_i + \mu_i + \delta_i)]^2\right)}{\exp\left(-\sum_{i=1}^d \frac{1}{2\sigma_i^2}[z_i - (x_i + \mu_i)]^2\right)} \\
&= \exp\left(\sum_{i=1}^d \frac{1}{2\sigma_i^2}[2z_i\delta_i - 2(x_i + \mu_i)\delta_i - \delta_i^2]\right) \\
&= \exp\left(\sum_{i=1}^d \frac{z_i\delta_i}{\sigma_i^2} - \sum_{i=1}^d \frac{1}{2\sigma_i^2}[2(x_i + \mu_i)\delta_i + \delta_i^2)]\right) \\
&= \exp\left(\sum_{i=1}^d \frac{\delta_i}{\sigma_i^2} z_i - c\right)
\end{aligned} \tag{16}$$

where $c = \sum_{i=1}^{d} \frac{1}{2\sigma_i^2}[2(x_i + \mu_i)\delta_i + \delta_i^2)]$, which is constant w.r.t. z. Let $\beta = \log t + c$, we have:

$$\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2} z_i \leq \beta \Longleftrightarrow \exp\,(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2} z_i - c) \leq t$$
$$\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2} z_i \geq \beta \Longleftrightarrow \exp\,(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2} z_i - c) \geq t$$

Therefore, for any β, there is some $t > 0$ for which:

$$\{z : \sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2} z_i \leq \beta\} = \{z : \frac{f_Y(z)}{f_X(z)} \leq t\} \quad \text{and} \quad \{z : \sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2} z_i \geq \beta\} = \{z : \frac{f_Y(z)}{f_X(z)} \geq t\} \tag{17}$$

This completes the proof. □

Then, we prove the Theorem 4.1.

Theorem 4.1 (Randomized Smoothing with Anisotropic Gaussian Noise). *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\epsilon \sim \mathcal{N}(\mu, \mathbf{\Sigma})$. Let g' be defined as in Eq. (4). Suppose that for a specific $x \in \mathbb{R}^d$, there exist $c_A \in \mathcal{Y}$ and $\underline{p_A}$, $\overline{p_B} \in [0, 1]$ such that:*

$$\mathbb{P}(f(x + \epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x + \epsilon) = c) \tag{5}$$

Then $g'(x + \delta) = c_A$ for all $||\delta||_2 < R$, where

$$R = \frac{1}{2} \min\{\sigma_i\}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{6}$$

where σ_i denotes the variance on i-th dimension of the input, δ denotes the perturbation.

Proof. To prove $g'(x + \delta) = c_A$, it is equivalent to prove that

$$\mathbb{P}(f(x + \delta + \epsilon) = c_A) > \mathbb{P}(f(x + \delta + \epsilon) = c_B) \tag{18}$$

where c_B denotes the second probable class.

For brevity, define the random variables

$$\begin{aligned} X &:= x + \epsilon = \mathcal{N}(x + \mu, \mathbf{\Sigma}) \\ Y &:= x + \delta + \epsilon = \mathcal{N}(x + \mu + \delta, \mathbf{\Sigma}) \end{aligned} \tag{19}$$

Then, proving Eq. (18) is equivalent to prove:

$$\mathbb{P}(f(Y) = c_A) > \mathbb{P}(f(Y) = c_B) \tag{20}$$

From Eq. (5) we have:

$$\mathbb{P}(f(X) = c_A) \geq \underline{p_A} \quad \text{and} \quad \mathbb{P}(f(X) = c_B) \leq \overline{p_B} \tag{21}$$

Then, we will show how to use Lemma A.2 to prove Eq. (20) from Eq. (21).

First, we define the half-spaces:

$$
\begin{aligned}
A &:= \{z : \sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}(z_i - \mu_i - x_i) \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(\underline{p_A})\} \\
B &:= \{z : \sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}(z_i - \mu_i - x_i) \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(1 - \underline{p_B})\}
\end{aligned} \tag{22}
$$

where Φ^{-1} is the inverse of the standard Gaussian CDF.

Then, we will have:

$$
\begin{aligned}
\mathbb{P}(X \in A) &= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}(X_i - \mu_i - x_i) \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(\underline{p_A})) \\
&= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}\mathcal{N}(0, \sigma_i^2) \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(\underline{p_A})) \\
&= \mathbb{P}(\sum_{i=1}^{d} \mathcal{N}(0, \frac{\delta_i^2}{\sigma_i^2}) \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(\underline{p_A})) \\
&= \mathbb{P}(\sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\, \mathcal{N}(0,1) \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(\underline{p_A})) \\
&= \mathbb{P}(\mathcal{N}(0,1) \leq \Phi^{-1}(\underline{p_A})) \\
&= \underline{p_A}
\end{aligned} \tag{23}
$$

$$
\begin{aligned}
\mathbb{P}(X \in B) &= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}(X_i - \mu_i - x_i) \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(1 - \overline{p_B})) \\
&= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}\mathcal{N}(0, \sigma_i^2) \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(1 - \overline{p_B})) \\
&= \mathbb{P}(\sum_{i=1}^{d} \mathcal{N}(0, \frac{\delta_i^2}{\sigma_i^2}) \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(1 - \overline{p_B})) \\
&= \mathbb{P}(\sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\, \mathcal{N}(0,1) \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \Phi^{-1}(1 - \overline{p_B})) \\
&= \mathbb{P}(\mathcal{N}(0,1) \geq \Phi^{-1}(1 - \overline{p_B})) \\
&= \overline{p_B}
\end{aligned} \tag{24}
$$

Now we have $\mathbb{P}(X \in A) = \underline{p_A}$ (Eq. (23)), so by Eq. (21) we have $\mathbb{P}(f(X) = c_A) \geq \mathbb{P}(X \in A)$. Using Neyman-Pearson Lemma with $h(z) := \mathbf{1}[f(z) = c_A]$, we have:

$$
\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(Y \in A) \tag{25}
$$

Similarly, by Eq. (21), Eq. (24) and Neyman-Pearson Lemma, we also have:

$$
\mathbb{P}(f(Y) = c_B) \leq \mathbb{P}(Y \in B) \tag{26}
$$

Finally, to prove that $\mathbb{P}(f(Y)=c_A) \geq \mathbb{P}(f(Y)=c_B)$, we will need to prove:

$$\mathbb{P}(f(Y)=c_A) \geq \mathbb{P}(Y \in A) \geq \mathbb{P}(Y \in B) \geq \mathbb{P}(f(Y)=c_B) \tag{27}$$

Recall that $Y \sim \mathcal{N}(x+\mu+\delta, \mathbf{\Sigma})$, and $A := \{z : \sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}(z_i - \mu_i - x_i) \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(\underline{p_A})\}$, we can have:

$$\begin{aligned}
\mathbb{P}(Y \in A) &= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}(Y_i - \mu_i - x_i) \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(\underline{p_A})) \\
&= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}\mathcal{N}(\delta_i, \sigma_i^2) \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(\underline{p_A})) \\
&= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}\mathcal{N}(0, \sigma_i^2) + \sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2} \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(\underline{p_A})) \\
&= \mathbb{P}(\sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\mathcal{N}(0,1) + \sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2} \leq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(\underline{p_A})) \\
&= \mathbb{P}(\mathcal{N}(0,1) \leq \Phi^{-1}(\underline{p_A}) - \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}}) \\
&= \Phi(\Phi^{-1}(\underline{p_A}) - \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}})
\end{aligned} \tag{28}$$

Similarly, with $B := \{z : \sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}(z_i - \mu_i - x_i) \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(1-\underline{p_B})\}$, we have:

$$\begin{aligned}
\mathbb{P}(Y \in B) &= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}(Y_i - \mu_i - x_i) \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(1-\overline{p_B})) \\
&= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}\mathcal{N}(\delta_i, \sigma_i^2) \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(1-\overline{p_B})) \\
&= \mathbb{P}(\sum_{i=1}^{d} \frac{\delta_i}{\sigma_i^2}\mathcal{N}(0, \sigma_i^2) + \sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2} \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(1-\overline{p_B})) \\
&= \mathbb{P}(\sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\mathcal{N}(0,1) + \sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2} \geq \sqrt{\sum_{i-1}^{d} \frac{\delta_i^2}{\sigma_i^2}}\Phi^{-1}(1-\overline{p_B})) \\
&= \mathbb{P}(\mathcal{N}(0,1) \geq \Phi^{-1}(1-\overline{p_B}) - \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}}) \\
&= \mathbb{P}(\mathcal{N}(0,1) \leq \Phi^{-1}(\overline{p_B}) + \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}}) \\
&= \Phi(\Phi^{-1}(\overline{p_B}) + \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}})
\end{aligned} \tag{29}$$

Therefore, to ensure $\mathbb{P}(Y \in A) \geq \mathbb{P}(Y \in B)$, we need:

$$\Phi(\Phi^{-1}(\underline{p_A}) - \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}}) \geq \Phi(\Phi^{-1}(\overline{p_B}) + \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}}) \quad \Longleftrightarrow \quad \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \leq \frac{1}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{30}$$

Let σ_m denote the minimum σ_i, we have

$$\sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}} \leq \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_m^2}} = \frac{||\delta||_2}{\sigma_m} \tag{31}$$

Therefore, if $\frac{||\delta||_2}{\sigma_m} \leq \frac{1}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$, it can ensure Eq. (30).

To conclude, $g'(x+\delta) = c_A$ is ensured if:

$$||\delta||_2 \leq \frac{1}{2} \min\{\sigma_i\}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{32}$$

Figure 7 also illustrates the relationship between the certified radius and Eq. (30).

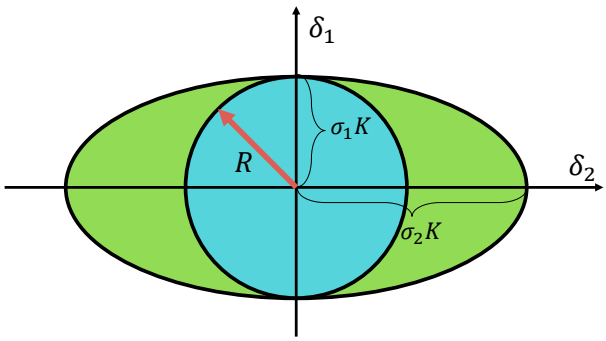

Figure 7. **Illustration**. Considering a δ space with two dimensions, Eq. (30) construct an ellipse with semi-minor axes $\sigma_1 K$ and semi-major axes $\sigma_2 K$, where $K = \frac{1}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$ and $\sigma_1 < \sigma_2$. Within the ellipse, the smoothed classifier's prediction is guaranteed to be c_A. To find the certified radius R in ℓ_2 norm, it is equivalent to find the circle of radius such that the radius is the semi-minor axes of the ellipse since within such circle, the smoothed prediction is constantly to be c_A. Therefore, in high-dimensional case, our certified radius is $\min\{\sigma_i\}K$.

□

B. Proofs for Theorem 4.2

Follow the proof in Appendix A, we can prove the binary case of Theorem 4.1.

Theorem 4.2 (Binary Case). *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\epsilon \sim \mathcal{N}(\mu, \mathbf{\Sigma})$. Let g' be defined as in Eq. (4). Suppose that for a specific $x \in \mathbb{R}^d$, there exist $c_A \in \mathcal{Y}$ and $\underline{p_A}$ such that:*

$$\mathbb{P}(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \frac{1}{2} \tag{7}$$

Then $g'(x+\delta) = c_A$ for all $||\delta||_2 < R$, where

$$R = \min\{\sigma_i\}\Phi^{-1}(\underline{p_A}) \tag{8}$$

Proof. In the binary case, if $\mathbb{P}(f(x+\delta+\epsilon)=c_A) > 1/2$, we can ensure that $g'(x+\delta)=c_A$, which is also equivalent to prove:

$$\mathbb{P}(f(Y)=c_A) > \frac{1}{2} \tag{33}$$

Define the half-space as in Eq. (22), we also have $\mathbb{P}(X \in A) = \underline{p_A}$ by Eq. (23). Using Neyman-Pearson Lemma, we have:

$$\mathbb{P}(f(Y)=c_A) \geq \mathbb{P}(Y \in A) \tag{34}$$

To guarantee that $\mathbb{P}(f(Y)=c_A) > \frac{1}{2}$, we need $\mathbb{P}(Y \in A) > \frac{1}{2}$.

By Eq. (28), we have:

$$\mathbb{P}(Y \in A) = \Phi(\Phi^{-1}(\underline{p_A}) - \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}}) \tag{35}$$

Therefore, to ensure $\mathbb{P}(f(Y)=c_A) > \frac{1}{2}$, we need:

$$\Phi(\Phi^{-1}(\underline{p_A}) - \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}}) > \frac{1}{2} \quad \Longleftrightarrow \quad \sqrt{\sum_{i=1}^{d} \frac{\delta_i^2}{\sigma_i^2}} < \Phi^{-1}(\underline{p_A}) \tag{36}$$

Similarly, it is obvious to have:

$$||\delta||_2 < \min\{\sigma_i\}\Phi^{-1}(\underline{p_A}) \tag{37}$$

This completes the proof. □

C. Algorithms

Algorithm 1 Anisotropic Randomized Smoothing Prediction

Given: Base Classifier f, Noise Generator g_n, Input image x, Monte Carlo Sampling Number n, confidence $1-\alpha$
$\mu, \mathbf{\Sigma} \leftarrow g_n(x)$
$counts \leftarrow ClassifySamples(f, x, \mu, \mathbf{\Sigma}, n)$
$\hat{c}_A, \hat{c}_B \leftarrow$ **top two indexes in** $counts$
$n_A, n_B \leftarrow counts[\hat{c}_A], counts[\hat{c}_B]$
if $BinomPValue(n_A, n_A+n_B, 0.5) \leq \alpha$ **then**
  return prediction $\hat{c}_A$
else
  return ABSTAIN
end if

D. Proof for Theorem 7.1

Similar to the proof for Theorem 4.1, we first prove the special case of Lemma A.1 when the random variables follows independent anisotropic Laplace distribution.

Lemma D.1. *Let $X \sim \mathcal{L}(x+\mu, \mathbf{\Lambda})$ and $Y \sim \mathcal{L}(x+\mu+\delta, \mathbf{\Lambda})$, where $\delta = [\delta_1, \delta_2, ..., \delta_d]$, $\mu = [\mu_1, \mu_2, ..., \mu_d]$ and $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_d)$. Let $h : \mathbb{R}^d \to 0, 1$ be any deterministic or random function. Then:*

(1) If $S = \{z \in \mathbb{R}^d : \sum_{i=1}^{d} \frac{1}{\lambda_i}(|z_i - \delta_i| - |z_i|) \geq \beta\}$ for some β and $\mathbb{P}(h(X)=1) \geq \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y)=1) \geq \mathbb{P}(Y \in S)$

Algorithm 2 Anisotropic Randomized Smoothing Certification

Given: Base Classifier f, Noise Generator g_n, Input image x, Monte Carlo Sampling Number n_0 and n, confidence $1-\alpha$
$\mu, \mathbf{\Sigma} \leftarrow g_n(x)$
$counts\_select \leftarrow ClassifySamples(f, x, \mu, \mathbf{\Sigma}, n_0)$
$\hat{c}_A \leftarrow$ **top index in** $counts\_select$
$counts \leftarrow ClassifySamples(f, x, \mu, \mathbf{\Sigma}, n)$
$\underline{p_A} \leftarrow LowerConfBound(counts[\hat{c}_A], n, 1-\alpha)$
if $\underline{p_A} > \frac{1}{2}$ **then**
  return prediction $\hat{c}_A$ and radius $\min\{\sigma_i\}\Phi^{-1}(\underline{p_A})$
else
  return ABSTAIN
end if

(2) If $S = \{z \in \mathbb{R}^d : \sum_{i=1}^d \frac{1}{\lambda_i}(|z_i - \delta_i| - |z_i|) \le \beta\}$ *for some* β *and* $\mathbb{P}(h(X)=1) \le \mathbb{P}(X \in S)$*, then* $\mathbb{P}(h(Y)=1) \le \mathbb{P}(Y \in S)$

Proof. Let $X \sim \mathcal{L}(x+\mu, \mathbf{\Lambda})$ and $Y \sim \mathcal{L}(x+\mu+\delta, \mathbf{\Lambda})$. We have the probability density functions f_X and f_Y as:

$$f_X(z) = k \exp\left(-\sum_{i=1}^d \frac{1}{\lambda_i}|z_i - (x_i + \mu_i)|\right)$$

$$f_Y(z) = k \exp\left(-\sum_{i=1}^d \frac{1}{\lambda_i}|z_i - (x_i + \mu_i + \delta_i)|\right)$$

where k is a constant. The ratio of the PDF is:

$$\begin{aligned}\frac{f_Y(z)}{f_X(z)} &= \frac{\exp\left(-\sum_{i=1}^d \frac{1}{\lambda_i}|z_i - (x_i+\mu_i)|\right)}{\exp\left(-\sum_{i=1}^d \frac{1}{\lambda_i}|z_i - (x_i+\mu_i+\delta_i)|\right)} \\ &= \frac{\exp\left(-\sum_{i=1}^d \frac{1}{\lambda_i}|z_i|\right)}{\exp\left(-\sum_{i=1}^d \frac{1}{\lambda_i}|z_i - \delta_i|\right)} \\ &= \exp\left(-\sum_{i=1}^d \frac{1}{\lambda_i}(|z_i - \delta_i| - |z_i|)\right)\end{aligned} \tag{38}$$

Let $\beta = -\log t$, we have:

$$\begin{aligned}\sum_{i=1}^d \frac{1}{\lambda_i}(|z_i - \delta_i| - |z_i|) \ge \beta &\iff \frac{f_Y(z)}{f_X(z)} \le t \\ \sum_{i=1}^d \frac{1}{\lambda_i}(|z_i - \delta_i| - |z_i|) \le \beta &\iff \frac{f_Y(z)}{f_X(z)} \ge t\end{aligned} \tag{39}$$

This completes the proof. □

Theorem 7.1 (Randomized Smoothing with Anisotropic Laplace Noise). *Let* $f : \mathbb{R}^d \to \mathcal{Y}$ *be any deterministic or random function, and let* $\epsilon \sim \mathcal{L}(\mu, \mathbf{\Lambda})$*, where* $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_d)$*. Let* g'' *be defined as* $g''(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x+\epsilon) = c)$*. Suppose that for a specific* $x \in \mathbb{R}^d$*, there exist* $c_A \in \mathcal{Y}$ *and* $\underline{p_A}, \overline{p_B} \in [0,1]$ *such that:*

$$\mathbb{P}(f(x+\epsilon) = c_A) \ge \underline{p_A} \ge \overline{p_B} \ge \max_{c \ne c_A} \mathbb{P}(f(x+\epsilon) = c) \tag{14}$$

Then $g''(x+\delta) = c_A$ for all $||\delta||_1 < R$, where

$$R = \max\{\frac{1}{2}\min\{\lambda_i\}\log(\underline{p_A}/\overline{p_B}), \\ -\min\{\lambda_i\}\log(1-\underline{p_A}+\overline{p_B})\} \tag{15}$$

where λ_i is the variance on i-th dimension of the input.

Proof. Denote $T(x) = \sum_{i=1}^{d} \frac{(|x_i-\delta_i|-|x_i|)}{\lambda_i}$. Use Triangle Inequality for each term in the summation, we can derive a bound for $T(x)$:

$$-\sum_{i=1}^{d}\frac{|\delta_i|}{\lambda_i} \le T(x) \le \sum_{i=1}^{d}\frac{|\delta_i|}{\lambda_i} \tag{40}$$

Define two sets:

$$A := \{z : T(z) \ge \beta_1\} \\ B := \{z : T(z) \le \beta_2\} \tag{41}$$

where the β_1 and β_2 are selected to suffice:

$$\mathbb{P}(X \in A) = \underline{p_A} \\ \mathbb{P}(X \in B) = \overline{p_B} \tag{42}$$

Applying Lemma D.1 to Eq. (41), we have:

$$\mathbb{P}(f(Y) = c_A) \ge \mathbb{P}(Y \in A) \\ \mathbb{P}(f(Y) = c_B) \le \mathbb{P}(Y \in B) \tag{43}$$

To ensure $\mathbb{P}(f(Y) = c_A) \ge \mathbb{P}(f(Y) = c_B)$, we need:

$$\mathbb{P}(Y \in A) \ge \mathbb{P}(Y \in B) \tag{44}$$

For $\mathbb{P}(Y \in A)$, we have:

$$\begin{aligned}
\mathbb{P}(Y \in A) &= \iint ... \int_A k^d \exp(-\sum_{i=1}^{d}\frac{|x_i-\delta_i|}{\lambda_i}) dx_1 dx_2 ... dx_d \\
&= \iint ... \int_A k^d \exp(-\sum_{i=1}^{d}\frac{|x_i|}{\lambda_i}) \exp(-T(x)) dx_1 dx_2 ... dx_d \\
&\ge \iint ... \int_A k^d \exp(-\sum_{i=1}^{d}\frac{|x_i|}{\lambda_i}) \exp(-\sum_{i=1}^{d}\frac{|\delta_i|}{\lambda_i}) dx_1 dx_2 ... dx_d \\
&= \exp(-\sum_{i=1}^{d}\frac{|\delta_i|}{\lambda_i})\ \underline{p_A}
\end{aligned} \tag{45}$$

The inequality in the middle is derived by Eq. (40). Simlilarly, for $\mathbb{P}(Y \in B)$, we have:

$$\begin{aligned}
\mathbb{P}(Y \in B) &= \iint ... \int_b k^d \exp(-\sum_{i=1}^{d} \frac{|x_i - \delta_i|}{\lambda_i}) dx_1 dx_2 ... dx_d \\
&= \iint ... \int_B k^d \exp(-\sum_{i=1}^{d} \frac{|x_i|}{\lambda_i}) \exp(-T(x)) dx_1 dx_2 ... dx_d \\
&\leq \iint ... \int_B k^d \exp(-\sum_{i=1}^{d} \frac{|x_i|}{\lambda_i}) \exp(\sum_{i=1}^{d} \frac{|\delta_i|}{\lambda_i}) dx_1 dx_2 ... dx_d \\
&= \exp(\sum_{i=1}^{d} \frac{|\delta_i|}{\lambda_i}) \, \overline{p_B}
\end{aligned} \tag{46}$$

Therefore, the robustness is guaranteed if $\exp(-\sum_{i=1}^{d} \frac{|\delta_i|}{\lambda_i}) \, \underline{p_A} \geq \exp(\sum_{i=1}^{d} \frac{|\delta_i|}{\lambda_i}) \, \overline{p_B}$, which is equivalent to:

$$\sum_{i=1}^{d} \frac{|\delta_i|}{\lambda_i} \leq \frac{1}{2} \log(\underline{p_A} / \overline{p_B}) \tag{47}$$

Since

$$\sum_{i=1}^{d} \frac{|\delta_i|}{\lambda_i} \leq \sum_{i=1}^{d} \frac{|\delta_i|}{\min\{\lambda_i\}} \tag{48}$$

if $||\delta||_1 \leq \frac{1}{2} \min\{\lambda_i\} \log(\underline{p_A} / \overline{p_B})$, the inequality (47) is also sufficed.

Also, if we apply the complement set of A to Eq. (45), we have:

$$\mathbb{P}(f(Y) = c_A) \geq 1 - \exp(\sum_{i=1}^{d} \frac{|\delta_i|}{\lambda_i})(1 - \underline{p_A}) \tag{49}$$

By Eq. (46) and Eq. (44), we have:

$$\sum_{i=1}^{d} \frac{|\delta_i|}{\lambda_i} \leq -\log(1 - \underline{p_A} + \overline{p_B}) \tag{50}$$

Similarly, by Eq. (48), to guarantee the robustness, we need:

$$||\delta||_1 \leq -\min\{\lambda_i\} \log(1 - \underline{p_A} + \overline{p_B}) \tag{51}$$

In conclusion, to guarantee the robustness, we need:

$$||\delta||_1 \leq \max\{\frac{1}{2} \min\{\lambda_i\} \log(\underline{p_A} / \overline{p_B}), -\min\{\lambda_i\} \log(1 - \underline{p_A} + \overline{p_B})\} \tag{52}$$

This completes the proof. □