Improved, Deterministic Smoothing for ℓ_1 Certified Robustness

Alexander Levine [1] **Soheil Feizi** [1]

Abstract

Randomized smoothing is a general technique for computing sample-dependent robustness guarantees against adversarial attacks for deep classifiers. Prior works on randomized smoothing against ℓ_1 adversarial attacks use additive smoothing noise and provide probabilistic robustness guarantees. In this work, we propose a non-additive and deterministic smoothing method, **D**eterministic **S**moothing with **S**plitting **N**oise (**DSSN**). To develop DSSN, we first develop SSN, a randomized method which involves generating each noisy smoothing sample by first randomly splitting the input space and then returning a representation of the center of the subdivision occupied by the input sample. In contrast to uniform additive smoothing, the SSN certification does not require the random noise components used to be independent. Thus, smoothing can be done effectively in just one dimension and can therefore be efficiently derandomized for quantized data (e.g., images). To the best of our knowledge, this is the first work to provide deterministic "randomized smoothing" for a norm-based adversarial threat model while allowing for an arbitrary classifier (i.e., a deep model) to be used as a base classifier and without requiring an exponential number of smoothing samples. On CIFAR-10 and ImageNet datasets, we provide substantially larger ℓ_1 robustness certificates compared to prior works, establishing a new state-of-the-art. The determinism of our method also leads to significantly faster certificate computation. Code is available at: `https://github.com/alevine0/smoothingSplittingNoise`.

1. Introduction and Related Works

Adversarial robustness in machine learning is a broad and widely-studied field which characterizes the *worst-case* behavior of machine learning systems under small input perturbations (Szegedy et al., 2013; Goodfellow et al., 2014; Carlini & Wagner, 2017). One area of active research is the design of *certifiably-robust* classifiers where, for each input $\boldsymbol{x}$, one can compute a magnitude ρ, such that *all* perturbed inputs $\boldsymbol{x}'$ within a radius ρ of $\boldsymbol{x}$ are guaranteed to be classified in the same way as $\boldsymbol{x}$. Typically, ρ represents a distance in an ℓ_p norm: $\|\boldsymbol{x} - \boldsymbol{x}'\|_p \leq \rho$, for some p which depends on the technique used.[1]

A variety of techniques have been proposed for certifiably ℓ_p-robust classification (Wong & Kolter, 2018; Gowal et al., 2018; Raghunathan et al., 2018; Tjeng et al., 2019; Zhang et al., 2018; Singla & Feizi, 2020). Among these are techniques that rely on Lipschitz analysis: if a classifier's logit functions can be shown to be Lipschitz-continuous, this immediately implies a robustness certificate (Li et al., 2019b; Anil et al., 2019). In particular, consider a classifier with logits $\{p_A, p_B, p_C, ...\}$, all of which are c-Lipschitz. Suppose for an input $\boldsymbol{x}$, we have $p_A(\boldsymbol{x}) > p_B(\boldsymbol{x}) \geq p_C(\boldsymbol{x}) \geq ....$ Also suppose the gap between the largest and the second largest logits is d (i.e. $p_A(\boldsymbol{x}) - p_B(\boldsymbol{x}) = d$). The Lipschitzness implies that for all $\boldsymbol{x}'$ such that $\|\boldsymbol{x} - \boldsymbol{x}'\| < d/(2c)$, $p_A(\boldsymbol{x}')$ will still be the largest logit: in this ball,

$$p_A(\boldsymbol{x}') > p_A(\boldsymbol{x}) - \frac{d}{2} \geq p_{\text{others}}(\boldsymbol{x}) + \frac{d}{2} > p_{\text{others}}(\boldsymbol{x}'), \quad (1)$$

where the first and third inequalities are due to Lipschitzness. Certification techniques based on *randomized smoothing* (Cohen et al., 2019; Salman et al., 2019; Zhai et al., 2020; Mohapatra et al., 2020; Jeong & Shin, 2020; Mohapatra et al., 2020; Yang et al., 2020; Lee et al., 2019; Lecuyer et al., 2019; Li et al., 2019a; Teng et al., 2020), are, at the time of writing, the only robustness certification techniques that scale to tasks as complex as ImageNet classification. (See Li et al. (2020) for a recent and comprehensive review and comparison of robustness certification methods.) In randomized smoothing methods, a "base" classifier is used

[1]Department of Computer Science, University of Maryland, College Park, Maryland, USA. Correspondence to: Alexander Levine <alevine0@cs.umd.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]Certifiably-robust classifiers for non-ℓ_p threat models have also been proposed including sparse (ℓ_0) adversarial attacks (Levine & Feizi, 2020b; Lee et al., 2019), Wasserstein attacks (Levine & Feizi, 2020c), geometric transformations (Fischer et al., 2020) and patch attacks (Chiang et al., 2020; Levine & Feizi, 2020a; Xiang et al., 2020; Zhang et al., 2020). However, developing improved certified defenses under ℓ_p adversarial threat models remains an important problem and will be our focus in this paper.

to classify a large set of randomly-perturbed versions $(\boldsymbol{x} + \epsilon)$ of the input image $\boldsymbol{x}$ where ϵ is drawn from a fixed distribution. The final classification is then taken as the plurality-vote of these classifications on noisy versions of the input. If samples $\boldsymbol{x}$ and $\boldsymbol{x}'$ are close, the distributions of $(\boldsymbol{x} + \epsilon)$ and $(\boldsymbol{x}' + \epsilon)$ will substantially overlap, leading to provable robustness. Salman et al. (2019) and Levine et al. (2019) show that these certificates can in some cases be understood as certificates based on Lipschitz continuity where the *expectation* of the output of the base classifier (or a function thereof) over the smoothing distribution is shown to be Lipschitz.

Randomized smoothing for the ℓ_1 threat model was previously proposed by Lecuyer et al. (2019); Mohapatra et al. (2020); Li et al. (2019a); Teng et al. (2020); Lee et al. (2019) and Yang et al. (2020). Yang et al. (2020) shows the best ℓ_1 certification performance (using a certificate originally presented by Lee et al. (2019) without experiments). These methods use *additive* smoothing noise and provide *high-probability* certificates, with a failure rate that depends on the number of noisy samples. Outside of the setting of certified robustness, practical attacks in the ℓ_1 threat model have additionally been studied (Chen et al., 2018).

In this work, we propose a *non-additive* smoothing method for ℓ_1-certifiable robustness on quantized data that is *deterministic*. By "quantized" data, we mean data where each feature value occurs on a discrete level. For example, standard image files (including standard computer vision datasets, such as ImageNet and CIFAR-10) are quantized, with all pixel values belonging to the set $\{0, 1/255, 2/255, ..., 1\}$. As Carlini & Wagner (2017) notes, if a natural dataset is quantized, adversarial examples to this dataset must also be quantized (in order to be recognized/saved as valid data at all). Therefore, our assumption of quantized data is a rather loose constraint which applies to many domains considered in adversarial machine learning. We call our method **D**eterministic **S**moothing with **S**plitting **N**oise (**DSSN**). DSSN produces *exact* certificates, rather than high-probability ones. It also produces certificates in substantially less time than randomized smoothing because a large number of noise samples are no longer required. In addition to these benefits, the certified radii generated by DSSN are significantly larger than those of the prior state-of-the-art.

To develop DSSN, we first propose a randomized method, Smoothing with Splitting Noise (**SSN**). Rather than simple additive noise, SSN uses "splitting" noise to generate a noisy input $\tilde{\boldsymbol{x}}$: first, we generate a noise vector $\boldsymbol{s}$ to split the input domain $[0, 1]^d$ into subdivisions. Then, the noisy input $\tilde{\boldsymbol{x}}$ is just the center of whichever sub-division $\boldsymbol{x}$ belongs to. In contrast to prior smoothing works, this noise model is *non-additive*.

In contrast to additive uniform noise where the noise components (ϵ_i's in ϵ) are independently distributed, in SSN, the splitting vector components (s_i's in $\boldsymbol{s}$) *do not* need to be independently distributed. Thus, unlike the additive uniform smoothing where noise vectors must be drawn from a d-dimensional probability distribution, in SSN, the splitting vectors can be drawn from a *one-dimensional* distribution. In the quantized case, the splitting vector can be further reduced to a choice between a small number of elements, leading to a derandomized version of SSN (i.e. DSSN).

Below, we summarize our contributions:

- We propose a novel randomized smoothing method, **SSN**, for the ℓ_1 adversarial threat model (Theorem 2).
- We show that **SSN** effectively requires smoothing in *one-dimension* (instead of d), thus it can be efficiently derandomized, yielding a deterministic certifiably robust classification method called **DSSN**.
- On ImageNet and CIFAR-10, we empirically show that DSSN significantly outperforms previous smoothing-based robustness certificates, effectively establishing a new state-of-the-art.

1.1. Prior Works on Deterministic Smoothing

While this work is, to the best of our knowledge, the first to propose a deterministic version of a randomized smoothing algorithm to certify for a norm-based threat model without resricting the base classifier or requiring time exponential in the dimension d of the input, prior deterministic "randomized smoothing" certificates have been proposed:

- **Certificates for non-norm (ℓ_0-like) threat models**. This includes certificates against patch adversarial attacks (Levine & Feizi, 2020a); as well as poisoning attacks under a label-flipping (Rosenfeld et al., 2020) or whole-sample insertion/deletion (Levine & Feizi, 2021) threat-model. These threat models are "ℓ_0-like" because the attacker entirely corrupts some portion of the data, rather than just distorting it. Levine & Feizi (2020a) and Levine & Feizi (2021) deal with this by ensuring that only a bounded fraction of base classifications can possibly be simultaneously exposed to any of the corrupted data. In the respective cases of patch adversarial attacks and poisoning attacks, it is shown that this can be done with a finite number of base classifications. Rosenfeld et al. (2020)'s method, by contrast, is based on the randomized ℓ_0 certificate proposed by Lee et al. (2019), and is discussed below.
- **Certificates for restricted classes of base classifiers**. This includes k-nearest neighbors (Weber et al., 2020) (for ℓ_2 poisoning attacks) and linear models (Rosenfeld et al., 2020) (for label-flipping poisoning attacks). In

these cases, existing randomized certificates are evaluated exactly for a restricted set of base classifier models. (Cohen et al. (2019) and Lee et al. (2019)'s methods, respectively.) It is notable that these are both poisoning certificates: in the poisoning setting, where the corrupted data is the training data, true randomized smoothing is less feasible, because it requires training very large ensemble of classifiers to achieve desired statistical properties. Weber et al. (2020) also attempts this directly, however.

- **Certificates requiring time exponential in dimension** d. This includes, in particular, a concurrent work, (Kao et al., 2020), which provides deterministic ℓ_2 certificates. In order to be practical, this method requires that the first several layers of the network be Lipschitz-bounded by means other than smoothing. The "smoothing" is then applied only in a low-dimensional space. Kao et al. (2020) note that this method is unlikely to scale to ImageNet.

2. Notation

Let $\boldsymbol{x}$, $\boldsymbol{x}'$ represent two points in $[0,1]^d$. We assume that our input space is bounded: this assumption holds for many applications (e.g., pixel values for image classification). If the range of values is not $[0,1]$, all dimensions can simply be scaled. A "base" classifier function will be denoted as $f : \mathbb{R}^d \to [0,1]$. In the case of a multi-class problem, this may represent a single logit.

Let $\delta := \boldsymbol{x}' - \boldsymbol{x}$, with components δ_1, ..., δ_d. A function $p : [0,1]^d \to [0,1]$ is said to be c-Lipschitz with respect to the ℓ_1 norm iff:

$$|p(\boldsymbol{x}) - p(\boldsymbol{x}')| \leq c\|\delta\|_1, \quad \forall \boldsymbol{x}, \boldsymbol{x}'. \tag{2}$$

Let $\mathcal{U}(a,b)$ represent the uniform distribution on the range $[a,b]$, and $\mathcal{U}^d(a,b)$ represent a random d-vector, where each component is *independently* uniform on $[a,b]$.

Let $\mathbf{1}_{(\text{condition})}$ represent the indicator function, and $\mathbb{1}$ be the vector $[1,1,...]^T$. In a slight abuse of notation, for $z \in \mathbb{R}, n \in \mathbb{R}^+$, let $z \bmod n := z - n\lfloor \frac{z}{n} \rfloor$ where $\lfloor \cdot \rfloor$ is the floor function; we will also use $\lceil \cdot \rceil$ as the ceiling function. For example, $9.5 \bmod 2 = 1.5$.

We will also discuss quantized data. We will use q for the number of quantizations. Let

$$[a,b]_{(q)} := \left\{ i/q \;\middle|\; \lceil aq \rceil \leq i \leq \lfloor bq \rfloor \right\}. \tag{3}$$

In particular, $[0,1]_{(q)}$ denotes the set $\{0, 1/q, 2/q, ..., (1-q)/q, 1\}$. Let $\mathbf{x}, \mathbf{x}'$ represent two points in $[0,1]^d_{(q)}$. A domain-quantized function $p : [0,1]^d_{(q)} \to [0,1]$ is said to be c-Lipschitz with respect to the ℓ_1 norm iff:

$$|p(\mathbf{x}) - p(\mathbf{x}')| \leq c\|\delta\|_1, \quad \forall \mathbf{x}, \mathbf{x}' \in [0,1]^d_{(q)}, \tag{4}$$

where $\delta := \mathbf{x}' - \mathbf{x}$. The uniform distribution on the set $[a,b]_{(q)}$ is denoted $\mathcal{U}_{(q)}(a,b)$.

3. Prior Work on Uniform Smoothing for ℓ_1 Robustness

Lee et al. (2019) proposed an ℓ_1 robustness certificate using uniform random noise:

Theorem 1 (Lee et al. (2019))**.** *For any $f : \mathbb{R}^d \to [0,1]$ and parameter $\lambda \in \mathbb{R}^+$, define:*

$$p(\boldsymbol{x}) := \mathop{\mathbb{E}}_{\epsilon \sim \mathcal{U}^d(-\lambda,\lambda)} \left[f(\boldsymbol{x} + \epsilon) \right]. \tag{5}$$

Then, $p(.)$ is $1/(2\lambda)$-Lipschitz with respect to the ℓ_1 norm.

Yang et al. (2020) later provided a theoretical justification for the uniform distribution being optimal among *additive* noise distributions for certifying ℓ_1 robustness[2]. Yang et al. (2020) also provided experimental results on CIFAR-10 and ImageNet which before our work were the state-of-the-art ℓ_1 robustness certificates.

Following Cohen et al. (2019), Yang et al. (2020) applied the smoothing method to a "hard" (in Salman et al. (2019)'s terminology) base classifier. That is, if the base classifier returns the class c on input $\boldsymbol{x} + \epsilon$, then $f_c(\boldsymbol{x} + \epsilon) = 1$, otherwise $f_c(\boldsymbol{x} + \epsilon) = 0$. Also following Cohen et al. (2019), in order to apply the certificate in practice, Yang et al. (2020) first takes $N_0 = 64$ samples to estimate the plurality class A, and then uses $N = 100,000$ samples to lower-bound $p_A(\boldsymbol{x})$ (the fraction of noisy samples classified as A) with high probability. The other smoothed logit values ($p_B(\boldsymbol{x})$, etc.) can then all be assumed to be $\leq 1 - p_A(\boldsymbol{x})$. This approach has the benefit that each logit does not require an independent statistical bound, and thus reduces the estimation error, but has the drawback that certificates are impossible if $p_A(\boldsymbol{x}) \leq 0.5$, creating a gap between the clean accuracy of the smoothed classifier and the certified accuracy near $\rho = 0$.

We note that the stated Theorem 1 is slightly more general than the originally stated version by Lee et al. (2019): the original version assumed that only $p_A(\boldsymbol{x})$ is available, as in the above estimation scheme, and therefore just gave the ℓ_1 radius in which $p_A(\boldsymbol{x}')$ is guaranteed to remain ≥ 0.5. For completeness, we provide a proof of the more general form (Theorem 1) in the appendix.

In this work, we show that by using *deterministic smoothing* with *non-additive noise*, improved certificates can be achieved, because we (i) avoid the statistical issues presented above (by estimating all smoothed logits *exactly*), and (ii) improve the performance of the base classifier itself.

[2] More precisely, Yang et al. (2020) suggested that distributions with d-cubic level sets are optimal for ℓ_1 robustness.

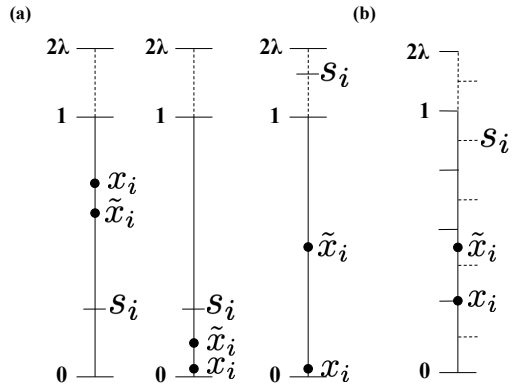

Figure 1. (a) Definition of $\tilde{\boldsymbol{x}}$ in the $\lambda \geq 0.5$ case. If $s_i \in [0, 1)$, then it "splits" the interval $[0, 1]$: $\tilde{x}_i$ is the center of whichever sub-interval x_i occurs in. If $s_i > 1$, $\tilde{x}_i = 0.5$, and no information about the original pixel is kept. (b) An example of $\tilde{\mathbf{x}}$ in the *quantized* $\lambda \geq 0.5$ case. Here, $q = 4$ and $2\lambda = 5/4$. We see that $\mathbf{x}_i = 1/4$ lies directly on a quantization level, while $s_i = 7/8$ lies on a half-step between quantization levels. We choose s_i to lie on "half-steps" for the sake of symmetry: the range of $\tilde{\mathbf{x}}_i$ is symmetrical around $1/2$.

4. Our Proposed Method

In this paper, we describe a new method, Smoothing with Splitting Noise (**SSN**), for certifiable robustness against ℓ_1 adversarial attacks. In this method, for each component x_i of $\boldsymbol{x}$, we randomly split the interval $[0, 1]$ into sub-intervals. The noised value $\tilde{x}_i$ is the middle of the sub-interval that contains x_i. We will show that this method corresponds closely to the uniform noise method, and so we continue to use the parameter λ. The precise correspondence will become clear in Section 4.2.1: however, for now, λ can be interpreted as controlling (the inverse of) the frequency with which the interval $[0, 1]$ is split into sub-intervals. We will show that this method, unlike the additive uniform noise method, can be efficiently derandomized. For simplicity, we will first consider the case corresponding to $\lambda \geq 0.5$, in which at most two sub-intervals are created, and present the general case later.

Theorem 2 ($\lambda \geq 0.5$ Case)**.** *For any $f : \mathbb{R}^d \to [0, 1]$, and $\lambda \geq 0.5$ let $\boldsymbol{s} \in [0, 2\lambda]^d$ be a random variable with a fixed distribution such that:*

$$s_i \sim \mathcal{U}(0, 2\lambda), \quad \forall i. \tag{6}$$

*Note that the components $s_1, ..., s_d$ are **not** required to be distributed independently from each other. Then, define:*

$$\tilde{x}_i := \frac{\min(s_i, 1) + \mathbf{1}_{x_i > s_i}}{2}, \quad \forall i \tag{7}$$

$$p(\boldsymbol{x}) := \mathop{\mathbb{E}}_{\boldsymbol{s}}[f(\tilde{\boldsymbol{x}})]. \tag{8}$$

Then $p(\cdot)$ is $1/(2\lambda)$-Lipschitz with respect to the ℓ_1 norm.

To understand the distribution of $\tilde{x}_i$, we can view s_i as "splitting" the interval $[0, 1]$ into two sub-intervals, $[0, s_i]$ and $(s_i, 1]$. $\tilde{x}_i$ is then the middle of whichever sub-interval contains x_i. If $s_i \geq 1$, then the interval $[0, 1]$ is not split, and $\tilde{x}_i$ assumes the value of the middle of the entire interval ($= 1/2$): see Figure 1-a.

Proof. Consider two arbitrary points $\boldsymbol{x}, \boldsymbol{x}'$ where $\delta := \boldsymbol{x}' - \boldsymbol{x}$. Note that $\max(x_i, x_i') - \min(x_i, x_i') = |x_i' - x_i| = |\delta_i|$. For a fixed vector $\boldsymbol{s}$, additionally note that $\tilde{x}_i = \tilde{x}_i'$ unless s_i falls between x_i and x_i' (i.e., unless $\min(x_i, x_i') \leq s_i < \max(x_i, x_i')$). Therefore:

$$\Pr_{\boldsymbol{s}}[\tilde{x}_i \neq \tilde{x}_i'] = \frac{|\delta_i|}{2\lambda}. \tag{9}$$

By union bound:

$$\begin{aligned} \Pr_{\boldsymbol{s}}[\tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}'}] &= \Pr_{\boldsymbol{s}}\left[\bigcup_{i=1}^{d} \tilde{x}_i \neq \tilde{x}_i'\right] \\ &\leq \sum_{i=1}^{d} \frac{|\delta_i|}{2\lambda} = \frac{\|\delta\|_1}{2\lambda} \end{aligned} \tag{10}$$

Then:

$$\begin{aligned} &|p(\boldsymbol{x}) - p(\boldsymbol{x}')| \\ &= \left| \mathop{\mathbb{E}}_{\boldsymbol{s}}[f(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{s}}[f(\tilde{\boldsymbol{x}}')] \right| \\ &= \left| \mathop{\mathbb{E}}_{\boldsymbol{s}}[f(\tilde{\boldsymbol{x}}) - f(\tilde{\boldsymbol{x}}')] \right| \\ &= \Bigg| \Pr_{\boldsymbol{s}}[\tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}'] \mathop{\mathbb{E}}_{\boldsymbol{s}}[f(\tilde{\boldsymbol{x}}) - f(\tilde{\boldsymbol{x}}') | \tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}'] \\ &\quad + \Pr_{\boldsymbol{s}}[\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}'] \mathop{\mathbb{E}}_{\boldsymbol{s}}[f(\tilde{\boldsymbol{x}}) - f(\tilde{\boldsymbol{x}}') | \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}'] \Bigg| \end{aligned} \tag{11}$$

Because $\mathbb{E}_{\boldsymbol{s}}[f(\tilde{\boldsymbol{x}}) - f(\tilde{\boldsymbol{x}}') | \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}']$ is zero, we have:

$$\begin{aligned} &|p(\boldsymbol{x}) - p(\boldsymbol{x}')| \\ &= \Pr_{\boldsymbol{s}}[\tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}'] \left| \mathop{\mathbb{E}}_{\boldsymbol{s}}[f(\tilde{\boldsymbol{x}}) - f(\tilde{\boldsymbol{x}}') | \tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}'] \right| \\ &\leq \frac{\|\delta\|_1}{2\lambda} \cdot 1 \end{aligned} \tag{12}$$

where in the final step, we used Equation 10, as well as the assumption that $f(\cdot) \in [0, 1]$. Thus, by the definition of Lipschitz-continuity, p is $1/(2\lambda)$-Lipschitz with respect to the ℓ_1 norm. □

It is important that we do **not** require that s_i's be independent. (Note the union bound in Equation 10: the inequality holds regardless of the joint distribution of the components of $\boldsymbol{s}$, as long as each s_i is uniform.) This allows us to develop a deterministic smoothing method below.

4.1. Deterministic SSN (DSSN)

If SSN is applied to quantized data (e.g. images), we can use the fact that the noise vector $\boldsymbol{s}$ in Theorem 2 is *not* required to have independently-distributed components to derive an efficient derandomization of the algorithm. In order to accomplish this, we first develop a quantized version of the SSN method, using input $\mathbf{x} \in [0,1]_q^d$ (i.e. $\mathbf{x}$ is a vector whose components belong to $\{0, 1/q, ..., 1\}$). To do this, we simply choose each of our splitting values s_i to be on one of the half-steps between possible quantized input values: $\boldsymbol{s} \in [0, 2\lambda - 1/q]_{(q)}^d + \mathbb{1}/(2q)$. We also require that 2λ is a multiple of $1/q$ (in experiments, when comparing to randomized methods with continuous λ, we use $\lambda' = \lfloor 2\lambda q \rfloor / 2q$.) See Figure 1-b.

Corollary 1 ($\lambda \geq 0.5$ Case)**.** *For any $f : \mathbb{R}^d \to [0,1]$, and $\lambda \geq 0.5$ (with 2λ a multiple of $1/q$), let $\boldsymbol{s} \in [0, 2\lambda - 1/q]_{(q)}^d + \mathbb{1}/(2q)$ be a random variable with a fixed distribution such that:*

$$s_i \sim \mathcal{U}_{(q)}\left(0, 2\lambda - 1/q\right) + 1/(2q), \quad \forall i. \tag{13}$$

*Note that the components $s_1, ..., s_d$ are **not** required to be distributed independently from each other. Then, define:*

$$\tilde{\mathbf{x}}_i := \frac{\min(s_i, 1) + \mathbf{1}_{\mathbf{x}_i > s_i}}{2}, \quad \forall i \tag{14}$$

$$p(\mathbf{x}) := \mathop{\mathbb{E}}_{\boldsymbol{s}}\left[f(\tilde{\mathbf{x}})\right]. \tag{15}$$

Then, $p(.)$ is $1/(2\lambda)$-Lipschitz with respect to the ℓ_1 norm on the quantized domain $\mathbf{x} \in [0,1]_{(q)}^d$.

Proof. Consider two arbitrary quantized points $\mathbf{x}, \mathbf{x}'$ where $\delta = \mathbf{x}' - \mathbf{x}$. Again note that $\max(\mathbf{x}_i, \mathbf{x}'_i) - \min(\mathbf{x}_i, \mathbf{x}'_i) = |\mathbf{x}'_i - \mathbf{x}_i| = |\delta_i|$. For a fixed vector $\boldsymbol{s}$, additionally note that $\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}'_i$ unless s_i falls between $\mathbf{x}_i$ and $\mathbf{x}'_i$ (i.e., unless $\min(\mathbf{x}_i, \mathbf{x}'_i) \leq s_i < \max(\mathbf{x}_i, \mathbf{x}'_i)$). Note that δ_i must be a multiple of $1/q$, and that there are exactly $q \cdot |\delta_i|$ discrete values that s_i can take such that the condition $\min(\mathbf{x}_i, \mathbf{x}'_i) \leq s_i < \max(\mathbf{x}_i, \mathbf{x}'_i)$ holds. This is out of $2\lambda q$ possible values over which s_i is uniformly distributed. Thus, we have:

$$\Pr_{\boldsymbol{s}}[\tilde{\mathbf{x}}_i \neq \tilde{\mathbf{x}'}_i] = \frac{|\delta_i|}{2\lambda} \tag{16}$$

The rest of the proof proceeds as in the continuous case (Theorem 2). □

If we required that s_i's be independent, an exact computation of $p(\mathbf{x})$ would have required evaluating $(2\lambda q)^d$ possible values of $\boldsymbol{s}$. This is not practical for large d. However, because we do not have this independence requirement, we can avoid this exponential factor. To do this, we first choose a single scalar splitting value s_{base}: each s_i is then simply a constant offset of s_{base}. We proceed as follows:

First, before the classifier is ever used, we choose a single, fixed, arbitrary vector $\mathbf{v} \in [0, 2\lambda - 1/q]_{(q)}^d$. In practice, $\mathbf{v}$ is generated pseudorandomly when the classifier is trained, and the seed is stored with the classifier so that the same $\mathbf{v}$ is used whenever the classifier is used. Then, at test time, we sample a scalar variable as:

$$s_{\text{base}} \sim \mathcal{U}_{(q)}(0, 2\lambda - 1/q) + 1/(2q). \tag{17}$$

Then, we generate each s_i by simply adding the base variable s_{base} to v_i:

$$\forall i, \quad s_i := (s_{\text{base}} + v_i) \mod 2\lambda \tag{18}$$

Note that the marginal distribution of each s_i is $s_i \sim \mathcal{U}_{(q)}\left(0, 2\lambda - 1/q\right) + 1/(2q)$, which is sufficient for our provable robustness guarantee. In this scheme, the only source of randomness at test time is the single random scalar s_{base}, which takes on one of $2\lambda q$ values. We can therefore evaluate the exact value of $p(\mathbf{x})$ by simply evaluating $f(\tilde{\mathbf{x}})$ a total of $2\lambda q$ times, for each possible value of s_{base}. Essentially, by removing the independence requirement, the splitting method allows us to replace a d-dimensional noise distribution with a *one*-dimensional noise distribution. In quantized domains, this allows us to efficiently derandomize the SSN method without requiring exponential time. We call this resulting deterministic method **DSSN**.

One may wonder why we do not simply use $s_1 = s_2 = s_3 ... = s_d$. While this can work, it leads to some undesirable properties when $\lambda > 0.5$. In particular, note that with probability $(2\lambda - 1)$, we would have all splitting values $s_i > 1$. This means that every element $\tilde{x}_i$ would be 0.5. In other words, with probability $(2\lambda - 1)/(2\lambda)$, $\tilde{\mathbf{x}} = 0.5 \cdot \mathbb{1}$. This restricts the expressivity of the smoothed classifier:

$$p(\mathbf{x}) = \frac{2\lambda - 1}{2\lambda} f(0.5 \cdot \mathbb{1}) + \frac{1}{2\lambda} \mathop{\mathbb{E}}_{\boldsymbol{s} < 1}\left[f(\tilde{\mathbf{x}})\right]. \tag{19}$$

This is the sum of a constant, and a function bounded in $[0, 1/(2\lambda)]$. Clearly, this is undesirable. By contrast, if we use an offset vector $\mathbf{v}$ as described above, not every component will have $s_i > 1$ simultaneously. This means that $\tilde{\boldsymbol{x}}$ will continue to be sufficiently expressive over the entire distribution of s_{base}.

4.2. Relationship to Uniform Additive Smoothing

In this section, we explain the relationship between SSN and uniform additive smoothing (Yang et al., 2020) with two main objectives:

1. We show that, for each element x_i, the *marginal* distributions of the noisy element $\tilde{x}_i$ of SSN and the noisy element $(x_i + \epsilon_i)$ of uniform additive smoothing are directly related to one another. However we show that,

for large λ, the distribution of uniform additive smoothing $(x_i + \epsilon_i)$ has an undesirable property which SSN avoids. This creates large empirical improvements in certified robustness using SSN, demonstrating an additional advantage to our method separate from derandomization.

2. We show that additive uniform noise does *not* produce correct certificates when using arbitrary joint distributions of ϵ. This means that it cannot be easily derandomized in the way that SSN can.

4.2.1. Relationship between Marginal Distributions of $\tilde{x}_i$ and $(x_i + \epsilon_i)$

To see the relationship between uniform additive smoothing and SSN, we break the marginal distributions of each component of noised samples into cases (assuming $\lambda \geq 0.5$):

$$x_i + \epsilon_i \sim \begin{cases} \mathcal{U}(x_i - \lambda, 1 - \lambda) & \text{w. prob. } \frac{1-x_i}{2\lambda} \\ \mathcal{U}(1 - \lambda, \lambda) & \text{w. prob. } \frac{2\lambda-1}{2\lambda} \\ \mathcal{U}(\lambda, x_i + \lambda) & \text{w. prob. } \frac{x_i}{2\lambda} \end{cases} \tag{20}$$

$$\tilde{x}_i \sim \begin{cases} \frac{\mathcal{U}(x_i,1)}{2} & \text{w. prob. } \frac{1-x_i}{2\lambda} \\ \frac{1}{2} & \text{w. prob. } \frac{2\lambda-1}{2\lambda} \\ \frac{\mathcal{U}(1,x_i+1)}{2} & \text{w. prob. } \frac{x_i}{2\lambda} \end{cases} \tag{21}$$

We can see that there is a clear correspondence (which also justifies our re-use of the parameter λ.) In particular, we can convert the marginal distribution of uniform additive noise to the marginal distribution of SSN by applying a simple mapping: $\tilde{x}_i \sim g(x_i + \epsilon_i)$ where:

$$g(z) := \begin{cases} \frac{z+\lambda}{2} & \text{if } z < 1 - \lambda \\ \frac{1}{2} & \text{if } 1 - \lambda < z < \lambda \\ \frac{z-\lambda+1}{2} & \text{if } z > \lambda \end{cases} \tag{22}$$

For $\lambda = 0.5$, this is a simple affine transformation:

$$\tilde{x}_i \sim 1/2(x_i + \epsilon_i) + 1/4 \tag{23}$$

In other words, in the case of $\lambda = 0.5$, $\tilde{x}_i$ is also uniformly distributed. However, for $\lambda > 0.5$, Equation 20 reveals an unusual and undesirable property of using uniform additive noise: *regardless of the value of* x_i, there is always a fixed probability $\frac{2\lambda-1}{2\lambda}$ that the smoothed value $x_i + \epsilon_i$ is uniform on the interval $[1 - \lambda, \lambda]$. Furthermore, this constant probability represents the only case in which $(x_i + \epsilon_i)$ can assume values in this interval. These values therefore carry no information about x_i and are all equivalent to each other. However, if λ is large, this range dominates the total range of values of $x_i + \epsilon_i$ which are observed (See Figure 2-b.) By contrast, in SSN, while there is still a fixed $\frac{2\lambda-1}{2\lambda}$ probability that the smoothed component $\tilde{x}_i$ assumes a "no information" value, this value is always *fixed* ($\tilde{x}_i = 1/2$). Empirically, this dramatically improves performance when λ is large. Intuitively, this is because when using uniform additive smoothing, the base classifier must *learn to ignore* a very wide range of values (all values in the interval $[1 - \lambda, \lambda]$) while in SSN, the base classifier only needs to learn to ignore a specific constant "no information" value $1/2$.[3] Figure 2 compares the two representations schematically, and Figure 3 compares the two noise representations visually.

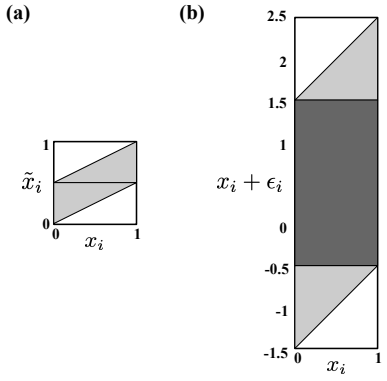

Figure 2. Range of noise values possible for each sample feature x_i, under (a) SSN, for any $\lambda \geq 0.5$ and (b) uniform additive smoothing, $\lambda = 1.5$. Possible pairs of clean and noise values are shown in grey (both light and dark). In uniform additive smoothing, note that all values of $x_i + \epsilon_i$ in the range [-0.5,1.5], shown in dark grey, can correspond to *any* value of x_i. This means that these values of $x_i + \epsilon_i$ carry no information about x_i whatsoever. By contrast, using SSN, only the value $\tilde{x}_i = 1/2$ has this property.

4.2.2. Can Additive Uniform Noise Be Derandomized?

As shown above, in the $\lambda = 0.5$ case, SSN leads to marginal distributions which are simple affine transformations of the marginal distributions of the uniform additive smoothing. One might then wonder whether we can derandomize additive uniform noise in a way similar to DSSN. In particular, one might wonder whether arbitrary joint distributions of ϵ can be used to generate valid robustness certificates with uniform additive smoothing, in the same way that arbitrary joint distributions of $\boldsymbol{s}$ can be used with SSN. It turns out that this is not the case. We provide a counterexample (for $\lambda = 0.5$) below:

Proposition 1. *There exists a base classifier* $f : \mathbb{R}^2 \to [0, 1]$ *and a joint probability distribution* $\mathcal{D}$*, such that* $\epsilon_1, \epsilon_2 \sim \mathcal{D}$ *has marginals* $\epsilon_1 \sim \mathcal{U}(-0.5, 0.5)$ *and* $\epsilon_2 \sim$

[3] Note that this use of a "no information" value bears some similarity to the "ablation" value in Levine & Feizi (2020b), a randomized smoothing defense for ℓ_0 adversarial attacks

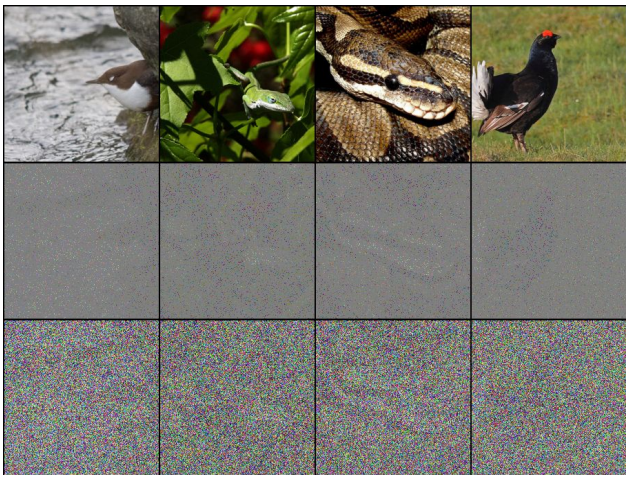

Figure 3. ImageNet images (top), under DSSN noise (middle) and uniform additive noise (bottom). In all cases, $\sigma(=\lambda/\sqrt{3}) = 3.5$. The images with additive noise are scaled to fit in the $[0, 1]$ range in order to be displayed. The images appear to be somewhat more visually discernible under DSSN noise, compared to additive noise.

$\mathcal{U}(-0.5, 0.5)$ *where for*

$$p(\boldsymbol{x}) := \mathop{\mathbb{E}}_{\epsilon\sim\mathcal{D}}\left[f(\boldsymbol{x}+\epsilon)\right], \tag{24}$$

$p(.)$ *is* ***not*** 1*-Lipschitz with respect to the* ℓ_1 *norm.*

Proof. Consider the base classifier $f(\boldsymbol{z}) := \mathbf{1}_{z_1>0.4+z_2}$, and let ϵ be distributed as $\epsilon_1 \sim \mathcal{U}(-0.5, 0.5)$ and $\epsilon_2 = \epsilon_1$. Consider the points $\boldsymbol{x} = [0.8, 0.2]^T$ and $\boldsymbol{x}' = [0.6, 0, 4]^T$. Note that $\|\delta\|_1 = 0.4$. However,

$$\begin{aligned} p(\boldsymbol{x}) &= \mathop{\mathbb{E}}_{\epsilon}[f(\boldsymbol{x}+\epsilon)] = \mathop{\mathbb{E}}_{\epsilon_1}[f(.8+\epsilon_1, .2+\epsilon_1)] = 1 \\ p(\boldsymbol{x}') &= \mathop{\mathbb{E}}_{\epsilon}[f(\boldsymbol{x}'+\epsilon)] = \mathop{\mathbb{E}}_{\epsilon_1}[f(.6+\epsilon_1, .4+\epsilon_1)] = 0 \end{aligned} \tag{25}$$

Thus, $|p(\boldsymbol{x}) - p(\boldsymbol{x}')| > \|\delta\|_1$. □

In the appendix, we provide intuition for this, by demonstrating that despite having similar *marginal* distributions, the *joint* distributions of $\tilde{\boldsymbol{x}}$ and $(\boldsymbol{x}+\epsilon)$ which can be generated by SSN and additive uniform noise, respectively, are in fact quite different. An example is shown in Figure 4.

4.3. General Case, including $\lambda < 0.5$

In the case $\lambda < 0.5$, we split the $[0, 1]$ interval not only at $s_i \in [0, 2\lambda]$, but also at every value $s_i + 2\lambda n$, for $n \in \mathbb{N}$. An example is shown in Figure 5. Note that this formulation covers the $\lambda \geq 0.5$ case as well (the splits for $n \geq 1$ are simply not relevant).

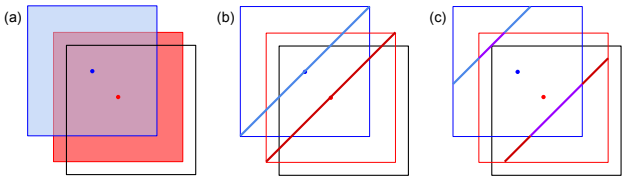

Figure 4. Comparison of independent uniform additive noise, correlated uniform additive noise, and correlated SSN, in $\mathbb{R}^2$ for $\lambda = 0.5$. In all figures, the blue and red points represent points $\boldsymbol{x}$ and $\boldsymbol{x}'$ and the black border represents the range $[0, 1]^2$. (a) Distributions of $\boldsymbol{x}+\epsilon$ and $\boldsymbol{x}'+\epsilon$ for independent uniform additive noise. The robustness guarantee relies on the significant overlap of the shaded regions, representing the sampled distributions. Note that by Equation 23, these are also the distributions of for $2\tilde{\boldsymbol{x}} - \mathbb{1}/2$ and $2\tilde{\boldsymbol{x}}' - \mathbb{1}/2$ using SSN with s_1 and s_2 distributed independently. (b) Using correlated additive noise ($\epsilon_1 = \epsilon_2$) does *not* produce an effective robustness certificate: the sampled distributions $\boldsymbol{x}+\epsilon$ and $\boldsymbol{x}'+\epsilon$ (blue and red lines) do not overlap. (c) Using correlated splitting noise ($s_1 = s_2$) produces an effective robustness certificate, because distributions of $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$ overlap significantly. Here, for consistency in scaling, we show the distributions of $2\tilde{\boldsymbol{x}} - \mathbb{1}/2$ and $2\tilde{\boldsymbol{x}}' - \mathbb{1}/2$ (blue line and red line), with the overlap shown as purple. Note that this is a *one-dimensional* smoothing distribution, and therefore can be efficiently derandomized.

Theorem 2 (General Case)**.** *For any* $f : \mathbb{R}^d \to [0, 1]$*, and* $\lambda > 0$ *let* $\boldsymbol{s} \in [0, 2\lambda]^d$ *be a random variable, with a fixed distribution such that:*

$$s_i \sim \mathcal{U}(0, 2\lambda), \quad \forall i. \tag{26}$$

Note that the components $s_1, ..., s_d$ *are* ***not*** *required to be distributed independently from each other. Then, define:*

$$\tilde{x}_i := \frac{\min(2\lambda\lceil\frac{x_i - s_i}{2\lambda}\rceil + s_i, 1)}{2} \tag{27}$$

$$+ \frac{\max(2\lambda\lceil\frac{x_i - s_i}{2\lambda} - 1\rceil + s_i, 0)}{2}, \quad \forall i \tag{28}$$

$$p(\boldsymbol{x}) := \mathop{\mathbb{E}}_{\boldsymbol{s}}\left[f(\tilde{\boldsymbol{x}})\right]. \tag{29}$$

Then, $p(.)$ *is* $1/(2\lambda)$*-Lipschitz with respect to the* ℓ_1 *norm.*

The proof for this case, as well as its derandomization, are provided in the appendix. As with the $\lambda \geq 0.5$ case, the derandomization allows for $p(\boldsymbol{x})$ to be computed exactly using $2\lambda q$ evaluations of f.

5. Experiments

We evaluated the performance of our method on CIFAR-10 and ImageNet datasets, matching all experimental conditions from (Yang et al., 2020) as closely as possible (further details are given in the appendix.) Certification performance data is given in Table 1 for CIFAR-10 and Figure 7 for Imagenet. Note that instead of using the hyperparameter λ, we

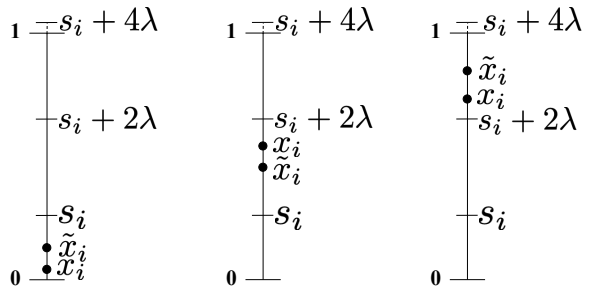

Figure 5. Example of $\tilde{x}_i$ in the $\lambda < 0.5$ case. In this case, the interval $[0, 1]$ is split into sub-intervals $[0, s_i]$, $(s_i, s_i + 2\lambda]$, and $(s_i + 2\lambda, 1]$. $\tilde{x}_i$ is assigned to the middle of whichever of these intervals x_i falls into.

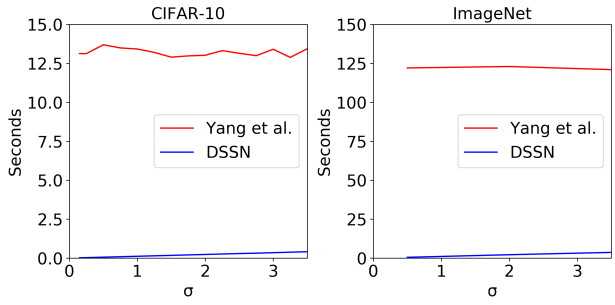

Figure 6. Comparison of the certification time per image of DSSN and Yang et al. (2020)'s uniform additive noise method. We used a single NVIDIA 2080 Ti GPU.

report experimental results in terms of $\sigma = \lambda/\sqrt{3}$: this is to match (Yang et al., 2020), where this gives the standard deviation of the uniform noise.

We find that DSSN significantly outperforms Yang et al. (2020) on both datasets, particularly when certifying for large perturbation radii. For example, at $\rho = 4.0$, DSSN provides a 36% certified accuracy on CIFAR-10, while uniform additive noise provides only 27% certified accuracy. In addition to these numerical improvements, DSSN certificates are *exact* while randomized certificates hold only with *high-probability*. Following Yang et al. (2020), all certificates reported here for randomized methods hold with 99.9% probability: there is no such failure rate for DSSN.

Additionally, the certification runtime of DSSN is reduced compared to Yang et al. (2020)'s method. Although in contrast to Yang et al. (2020), our certification time scales linearly with the noise level, the fact that Yang et al. (2020) uses 100,000 smoothing samples makes our method much faster even at the largest tested noise levels: see Figure 6. For example, on CIFAR-10 at $\sigma = 3.5$, we achieve an average runtime of 0.41 seconds per image, while Yang et al. (2020)'s method requires 13.44 seconds per image.

Yang et al. (2020) tests using both standard training on noisy samples as well as stability training (Li et al., 2019a): while our method dominates in both settings, we find that the stability training leads to less of an improvement in our methods, and is in some cases detrimental. For example, in Table 1, the best certified accuracy is always higher under stability training for uniform additive noise, while this is not the case for DSSN at $\rho < 3.0$. Exploring the cause of this may be an interesting direction for future work.[4]

In Figure 8, we compare the uniform additive smoothing method to DSSN, as well the *randomized* form of SSN with independent splitting noise. At mid-range noise levels, the primary benefit of our method is due to derandomization; while at large noise levels, the differences in noise representation discussed in Section 4.2.1 become more relevant. In the appendix, we provide complete certification data at all tested noise levels, using both DSSN and SSN with independent noise, as well as more runtime data. Additionally we further explore the effect of the noise representation: given that Equation 22 shows a simple mapping between (the marginal distributions of) SSN and uniform additive noise, we tested whether the gap in performance due to noise representations can be eliminated by a "denoising layer", as trained in (Salman et al., 2020). We did not find evidence of this: the gap persists even when using denoising.

6. Conclusion

In this work, we have improved the state-of-the-art smoothing-based robustness certificate for the ℓ_1 threat model, and provided the first scalable, general-use derandomized "randomized smoothing" certificate for a norm-based adversarial threat model. To accomplish this, we proposed a novel *non-additive* smoothing method. Determining whether such methods can be extended to other ℓ_p norms remains an open question for future work.

Acknowledgements

This project was supported in part by NSF CAREER AWARD 1942230, HR00111990077, HR001119S0026, HR00112090132, NIST 60NANB20D134 and Simons Fellowship on "Foundations of Deep Learning."

References

Anil, C., Lucas, J., and Grosse, R. Sorting out Lipschitz function approximation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International*

[4] On CIFAR-10, Yang et al. (2020) also tests using semi-supervised and transfer learning approaches which incorporate data from other datasets. We consider this beyond the scope of this work where we consider only the supervised learning setting.

	$\rho = 0.5$	$\rho = 1.0$	$\rho = 1.5$	$\rho = 2.0$	$\rho = 2.5$	$\rho = 3.0$	$\rho = 3.5$	$\rho = 4.0$
Uniform Additive Noise	70.54% (83.97% @ σ=0.5)	58.43% (78.70% @ σ=1.0)	50.73% (73.05% @ σ=1.75)	43.16% (73.05% @ σ=1.75)	33.24% (69.56% @ σ=2.0)	25.98% (62.48% @ σ=2.5)	20.66% (53.38% @ σ=3.5)	17.12% (53.38% @ σ=3.5)
Uniform Additive Noise (+Stability Training)	71.09% (78.79% @ σ=0.5)	60.36% (74.27% @ σ=0.75)	52.86% (65.88% @ σ=1.5)	47.08% (63.32% @ σ=1.75)	42.26% (57.49% @ σ=2.5)	38.55% (57.49% @ σ=2.5)	33.76% (57.49% @ σ=2.5)	27.12% (57.49% @ σ=2.5)
DSSN - Our Method	**72.25%** (81.50% @ σ=0.75)	**63.07%** (77.85% @ σ=1.25)	**56.21%** (71.17% @ σ=2.25)	**51.33%** (67.98% @ σ=3.0)	**46.76%** (65.40% @ σ=3.5)	42.66% (65.40% @ σ=3.5)	38.26% (65.40% @ σ=3.5)	33.64% (65.40% @ σ=3.5)
DSSN - Our Method (+Stability Training)	71.23% (79.00% @ σ=0.5)	61.04% (71.29% @ σ=1.0)	54.21% (66.04% @ σ=1.5)	49.39% (64.26% @ σ=1.75)	45.45% (59.88% @ σ=2.5)	**42.67%** (57.16% @ σ=3.0)	**39.46%** (56.29% @ σ=3.25)	**36.46%** (54.96% @ σ=3.5)

Table 1. Summary of results for CIFAR-10. Matching Yang et al. (2020), we test on 15 noise levels ($\sigma \in \{0.15, 0.25n \text{ for } 1 \leq n \leq 14\}$). We report the best certified accuracy at a selection of radii ρ, as well as the clean accuracy and noise level of the associated classifier. Our method dominates at all radii, although stability training seems to be less useful for our method. Note that these statistics are based on reproducing Yang et al. (2020)'s results; they are all within ± 1.5 percentage points of Yang et al. (2020)'s reported statistics.

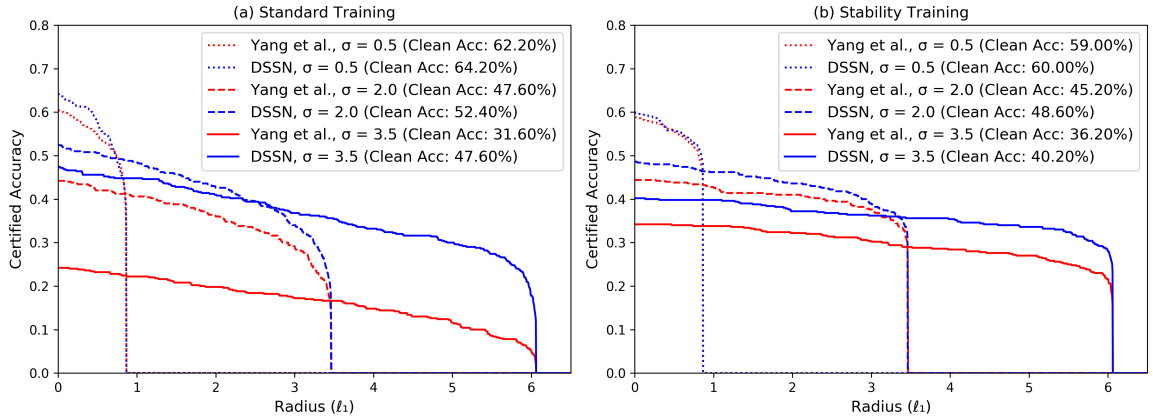

Figure 7. Results on ImageNet. We report results at three noise levels, with and without stability training. Our method dominates in all settings: however, especially at large noise, stability training seems to *hurt* our clean accuracy, rather than help it.

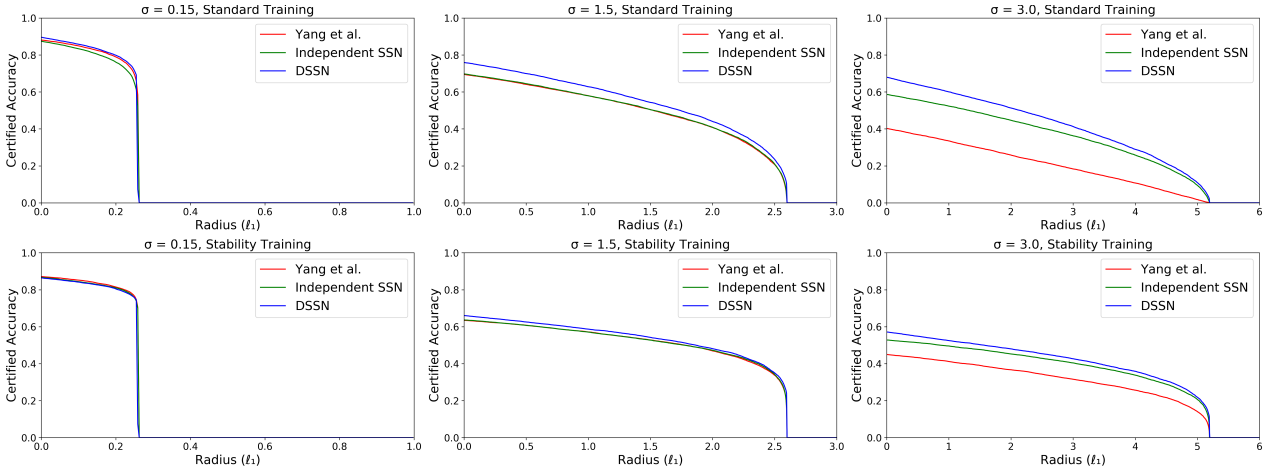

Figure 8. Comparison on CIFAR-10 of additive smoothing (Yang et al., 2020) to DSSN, as well as SSN with *random, independent* splitting noise, using the estimation scheme from (Yang et al., 2020). At very small levels of noise ($\sigma = 0.15$), there is little difference between the methods: in fact, with stability training, additive smoothing slightly outperforms DSSN. At intermediate noise levels, additive noise and independent SSN perform very similarly, but DSSN outperforms both. This suggests that, at this level, the primary benefit of DSSN is to eliminate estimation error (Section 3). At high noise levels, the largest gap is between additive noise and independent SSN, suggesting that in this regime, most of the performance benefits of DSSN are due to improved base classifier performance (Section 4.2.1).

Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pp. 291–301. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/anil19a.html.

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.

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Chiang, P., Ni, R., Abdelkader, A., Zhu, C., Studor, C., and Goldstein, T. Certified defenses for adversarial patches. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HyeaSkrYPH.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/cohen19c.html.

Fischer, M., Baader, M., and Vechev, M. Certified defense to image transformations via randomized smoothing. *Advances in Neural Information Processing Systems Foundation (NeurIPS)*, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Jeong, J. and Shin, J. Consistency regularization for certified robustness of smoothed classifiers. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 10558–10570. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/77330e1330ae2b086e5bfcae50d9ffae-Paper.pdf.

Kao, C.-C., Ko, J.-B., and Lu, C.-S. Deterministic certification to adversarial attacks via bernstein polynomial approximation, 2020.

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.-H., Yuan, Y., Chang, S., and Jaakkola, T. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *Advances in Neural Information Processing Systems*, pp. 4910–4921, 2019.

Levine, A. and Feizi, S. (de)randomized smoothing for certifiable defense against patch attacks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a.

Levine, A. and Feizi, S. Robustness certificates for sparse adversarial attacks by randomized ablation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4585–4593, 2020b.

Levine, A. and Feizi, S. Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020c.

Levine, A. and Feizi, S. Deep partition aggregation: Provable defenses against general poisoning attacks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YUGG2tFuPM.

Levine, A., Singla, S., and Feizi, S. Certifiably robust interpretation in deep learning. *arXiv preprint arXiv:1905.12105*, 2019.

Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, pp. 9464–9474, 2019a.

Li, L., Qi, X., Xie, T., and Li, B. Sok: Certified robustness for deep neural networks. *arXiv preprint arXiv:2009.04131*, 2020.

Li, Q., Haque, S., Anil, C., Lucas, J., Grosse, R., and Jacobsen, J. Preventing gradient attenuation in lipschitz constrained convolutional networks. In *NeurIPS*, 2019b.

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

Raghunathan, A., Steinhardt, J., and Liang, P. Semidefinite relaxations for certifying robustness to adversarial examples. In *Proceedings of the 32nd International Conference*

on Neural Information Processing Systems, NIPS'18, pp. 10900–10910, Red Hook, NY, USA, 2018. Curran Associates Inc.

Rosenfeld, E., Winston, E., Ravikumar, P., and Kolter, Z. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pp. 8230–8241. PMLR, 2020.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pp. 11292–11303, 2019.

Salman, H., Sun, M., Yang, G., Kapoor, A., and Kolter, J. Z. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems*, 33, 2020.

Singla, S. and Feizi, S. Second-order provable defenses against adversarial attacks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8981–8991. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/singla20a.html.

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

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.

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.

Weber, M., Xu, X., Karlas, B., Zhang, C., and Li, B. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904*, 2020.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5283–5292, 2018.

Xiang, C., Bhagoji, A., Sehwag, V., and Mittal, P. Patchguard: Provable defense against adversarial patches using masks on small receptive fields. *ArXiv*, abs/2005.10884, 2020.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I., and Li, J. Randomized smoothing of all shapes and sizes. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 10693–10705, 2020.

Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C.-J., and Wang, L. Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJx1Na4Fwr.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 4939–4948. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/d04863f100d59b3eb688a11f95b0ae60-Paper.pdf.

Zhang, Z., Yuan, B., McCoyd, M., and Wagner, D. Clipped bagnet: Defending against sticker attacks with clipped bag-of-features. *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 55–61, 2020.