Improving ℓ_1-Certified Robustness via Randomized Smoothing by Leveraging Box Constraints

Václav Voráček [1] **Matthias Hein** [1]

Abstract

Randomized smoothing is a popular method to certify robustness of image classifiers to adversarial input perturbations. It is the only certification technique which scales directly to datasets of higher dimension such as ImageNet. However, current techniques are not able to utilize the fact that any adversarial example has to lie in the image space, that is $[0, 1]^d$; otherwise, one can trivially detect it. To address this suboptimality, we derive new certification formulae which lead to significant improvements in the certified ℓ_1-robustness without the need of adapting the classifiers or change of smoothing distributions. Code is released at https://github.com/vvoracek/L1-smoothing.

1. Introduction

While neural networks have demonstrated excellent performance in a variety of tasks, they are susceptible to small (adversarial) changes of the input (Szegedy et al., 2014; Biggio et al.). Such behaviour is undesired, especially in the safety-critical applications. To mitigate the issue, initially the focus was on constructing empirically robust classifiers, and then check how the resulting model performs against adversarial attacks. However, such an approach only gives an upper bound on the actual robustness of the classifier and many initially considered promising methods later turned out to be broken (Athalye et al., 2018; Carlini et al., 2019; Tramer et al., 2020) due to stronger attacks. The only seemingly working method that does not produce any guarantees is adversarial training (Madry et al., 2018); but more powerful attacks show that the empirical robustness of classifiers is lower than originally claimed (Croce & Hein, 2020; Lin et al., 2022).

[1]Tübingen AI Center, University of Tübingen. Correspondence to: Václav Voráček <vaclav.voracek@uni-tuebingen.de>, Matthias Hein <matthias.hein@uni-tuebingen.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

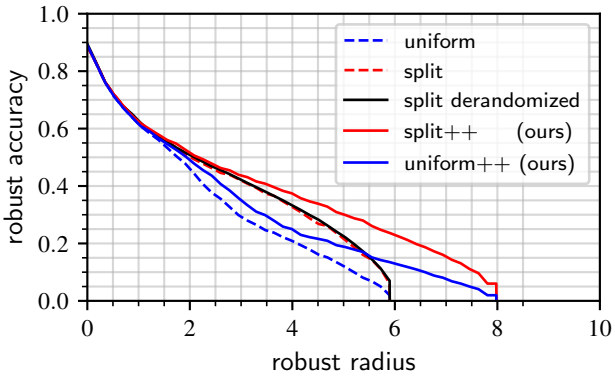

Figure 1: Certified ℓ_1 robust accuracies for CIFAR-10 dataset via randomized smoothing for three different types of noise. The curves uniform++ and split++ use the same networks and noise as uniform and split respectively, however, with the proposed improved certification we are able to significantly increase the certified robustness. The reported curves are pointwise maxima of robustness curves with different noise magnitudes.

Thus, an alternative approach is to certify robustness. Here, we are no longer interested in whether we can find (or fail to find) an adversarial example in the neighbourhood (called *threat model*), but rather focus on whether we can prove (or fail to prove) that there is no adversarial example. These methods roughly fall into three (arguably overlapping) categories:

- Propagate a "nice" set containing the threat model through the network; see, e.g., (Gowal et al., 2018; Wong et al., 2018).
- Force the Lipschitz constant of a model to be small; see, e.g., (Leino et al., 2021; Trockman & Kolter, 2021; Singla et al., 2022; Zhang et al., 2022a).
- Randomized smoothing; see, e.g., (Lecuyer et al., 2019; Cohen et al., 2019; Salman et al., 2019; Yang et al., 2020).

The two approaches discussed yielded either an upper bound (empirical robustness) or a lower bound (certified robustness) respectively on the actual adversarial robustness and in general, there are points for which the certification methods

cannot prove that they are robust, nor the attack is able to find an adversarial example. Although this property is undesirable, it is also inevitable in practice because the problem of finding adversarial examples even in ReLU networks is in NP-Complete (Katz et al., 2017). Therefore, determining the true robustness of a model is only possible for very small networks (Tjeng & Tedrake, 2017) and simple classifiers such as linear models, boosted decision stumps (Kantchelian et al., 2016) , or nearest neighbour (resp. prototype) classifiers (Wang et al., 2019; Saralajew et al., 2020; Voráček & Hein, 2022). Nevertheless, there is a line of work aiming at finding the actual robustness, or at least tightening the gap between the certifiable lower and upper bounds; see (Zhang et al., 2018; 2022b).

We discuss shortly the choice of the threat model. It is the perturbation set with respect to which we want to be robust. The common choices are the ℓ_p balls centered at the input points. While the choice of a threat model is always somewhat arbitrary, if it is not directly motivated by an application, the attacks (and defenses) to many interesting threat models strongly rely on techniques developed for the ℓ_p threat models for both empirical and certified robustness; see (Laidlaw et al., 2021; Voráček & Hein, 2022) for a perceptual metric threat model; (Wong et al., 2019; Levine & Feizi, 2020) for the Wasserstein distance threat model; (Brown et al., 2017; Metzen & Yatsura, 2021; Salman et al., 2022) for a patch threat model, in which the attacker is allowed to arbitrarily set the pixel values in a small patch. The choice of using different ℓ_p norms as threat models leads to qualitatively different adversarial perturbations. For example, when applying the ℓ_∞-threat model with a sufficiently small radius, the changes are typically imperceptible but affect every pixel. On the other hand, the ℓ_1-threat model allows for potentially significant changes in individual pixels, although limited to only a few of them. Instead of a given threat model, that is fixing the perturbation budget, one can also ask for the largest radius of a ball in a given norm in which the classifier does not change - the so called *robust radius*.

Definition 1.1. A classifier $f : \mathbb{R}^d \to \{0, 1\}$ is said to be *robust* at x with respect to a norm $\|\cdot\|$ with robust radius r if $\|x - y\| \leq r \implies f(x) = f(y)$ for every $y \in [0, 1]^d$.

The certified radii and perturbation magnitudes typically considered in empirical robustness for the ℓ_∞-norm are very similar. However, for the ℓ_1-norm, there is a significant difference in the radii considered between certified robustness and empirical defenses. For example, on ImageNet, radii around 4 are considered for certified robustness, while empirical defenses consider radii ranging from 60 to 255 (Croce & Hein, 2022; 2021). This suggests that there exists a gap between what can be certifiably attained and what appears to be empirically achievable for the ℓ_1-norm.

On the other hand it has been argued by Croce & Hein (2021) that ℓ_1-attacks are much more difficult and prone to fail compared to ℓ_∞-attacks and thus it could also be an overestimation problem. They conclude that the intersection of the image domain $[0, 1]^d$ and the ℓ_1-ball as the effective threat model has a quite different geometry than an ℓ_1-ball and construct their attack accordingly. Thus, our motivation is to consider the box-constraints even in the context of certification.

Contributions: In previous work of Levine & Feizi (2021) and Yang et al. (2020) on certified ℓ_1-robustness using randomized smoothing, it has been assumed that the input domain is $\mathbb{R}^d$, even though the techniques are mainly applied to image classifiers, where the domain is $[0, 1]^d$. We show in this paper that taking into account the box constraints of the image domain $[0, 1]^d$ can be used to certify significantly larger ℓ_1-balls than previous work. Our main result is based on the fact that the volume of the overlap of two ℓ_∞-balls when the centers of the balls are restricted to $[0, 1]^d$ behaves quite different from the unconstrained case which leads to an improved control of the smoothed classifier yielding the better guarantees. Our technique can be applied when the smoothing distribution is uniform Yang et al. (2020) as well as for the Splitting Noise of (Levine & Feizi, 2021). We also discuss an improved control of the failure probability as well the better scaling of the ℓ_1-certificates in the failure probability compared to ℓ_2 and ℓ_∞. Finally, we show in the experiments that our improved technique allows to certify much larger ℓ_1-radii than previous work.

1.1. Notation

Real interval between a, b is denoted $[a, b]$. We use Iverson brackets $[\![\text{statement}]\!]$ which is the indicator function of the set for which the statement is true. The floor function, $\lfloor x \rfloor$ stands for the maximal integer no larger than x:

$$\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\}.$$

The ℓ_p-ball with radius λ centered at $x \in \mathbb{R}^d$ is denoted as

$$\mathcal{B}_p(x, \lambda) = \{z \in \mathbb{R}^d \mid \|z - x\|_p \leq \lambda\}.$$

The uniform distribution on $\mathcal{B}_\infty(0, \lambda)$ in d dimensions is denoted $\mathcal{U}^d(\lambda)$. Volume of a set A is denoted as $Vol(A)$. A one hot vector with 1 at position i is denoted e_i and its dimension will be clear from the context. When the meaning is clear from context, we use f as a base classifier, q as a smoothing distribution, h as a smoothed classifier of f and H as the thresholded version of h as in Equation (1) and (2). Number of samples is denoted by n and the dimension is d.

2. Randomized smoothing

For the simplicity of exposure, we introduce randomized smoothing for the case of binary classification and discuss

the multiclass setting in Section 2.4. We start by treating the mathematical foundations of randomized smoothing and postpone the discussion on the algorithmic implementation to Section 2.5. We will also certify class 1 and the certification of class 0 is symmetric.

2.1. Mathematics of Randomized Smoothing

Randomized smoothing (Lecuyer et al., 2019) is a method that takes an arbitrary binary classifier $f : \mathbb{R}^d \to [0, 1]$ and a noise distribution q supported on $\mathbb{R}^d$; it produces a smoothed version h of the original classifier f:

$$h(x) = \mathop{\mathbb{E}}_{\varepsilon\sim q} f(x + \varepsilon). \tag{1}$$

For the thresholded classifier H defined as

$$H(x) = [\![h(x) > 0.5]\!] \tag{2}$$

we can certify adversarial robustness.

The intuition behind this is as follows: If the distribution q exhibits certain desirable characteristics (such as having a small total variation distance with its slightly shifted copy), for example, if it represents a uniform distribution within a hypercube with a radius larger than $\|\delta\|_1$ for some $\delta \in \mathbb{R}^d$, then $h(x)$ and $h(x + \delta)$ are both expectations of the same function under almost the same distribution; thus they should be similar. Therefore, if $h(x) \approx 1$, then also $h(x + \delta) > 0.5$ We formalize this intuition later in Proposition 2.1 for the case of ℓ_1 distance.

In (Yang et al., 2020), it has been argued and supported by both theoretical and experimental evidence that the optimal smoothing distribution for ℓ_1-robustness should have cubic superlevel sets. That is, a smoothing distribution with density q should satisfy that the set $U_t = \{x \in \mathbb{R}^d \mid q(x) \geq t\}$ is a hypercube for every t. In that case, we can express $q(x)$ as an (uncountable) mixture of uniform distributions supported in ℓ_∞-balls of specific radii and our proposed method can still be applied.

Proposition 2.1. *Let $f : \mathbb{R}^d \to [0, 1]$ and*

$$h(x) = \mathop{\mathbb{E}}_{\varepsilon\sim\mathcal{U}^d(\lambda)} f(x + \varepsilon).$$

Let B_1 and B_2 be the ℓ_∞-balls with radius λ centered at x, y respectively, then

$$h(y) \geq h(x) - 1 + \frac{Vol(B_1 \cap B_2)}{Vol(B_2)}.$$

Proof.

$$\begin{aligned} h(y) &= \frac{\int_{t\in B_2} f(t)dt}{Vol(B_2)} \geq \frac{\int_{t\in B_1\cap B_2} f(t)dt}{Vol(B_2)} \\ &= \frac{\int_{t\in B_1} f(t)dt - \int_{t\in B_1\setminus B_2} f(t)dt}{Vol(B_2)} \\ &\geq \frac{\int_{t\in B_1} f(t)dt - Vol(B_1 \setminus B_2)}{Vol(B_2)} \\ &= h(x) - 1 + \frac{Vol(B_1 \cap B_2)}{Vol(B_2)}, \end{aligned}$$

using $Vol(B_1 \setminus B_2) = Vol(B_2) - Vol(B_1 \cap B_2)$ and $Vol(B_1) = Vol(B_2)$. □

It remains to find a lower bound on the volume of intersection of two ℓ_∞-balls. For now, we present a simple bound and will return to a proper treatment later in Proposition 2.5 and Theorem 2.8.

Proposition 2.2. *Let B_1, B_2 be ℓ_∞-balls with radii λ centered at $x, y \in \mathbb{R}^d$ respectively; then*

$$\frac{Vol(B_1 \cap B_2)}{Vol(B_1)} \geq 1 - \frac{\|x - y\|_1}{2\lambda}.$$

Proof. The proof can be found in Appendix A.1. □

Now we have developed the intuition and tools, we are ready to state the foundational theorem of ℓ_1 robustness.

Theorem 2.3. *(Lee et al., 2019) Let $f : \mathbb{R}^d \to [0, 1]$ be a deterministic or random classifier. Then the smoothed classifier defined as:*

$$h(x) = \mathop{\mathbb{E}}_{\varepsilon\sim\mathcal{U}^d(\lambda)} [f(x + \varepsilon)]$$

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

Proof. Plugging the bound from Proposition 2.2 into Proposition 2.1 yields $h(y) - h(x) \leq \|x - y\|_1 / (2\lambda)$ for all $x, y \in \mathbb{R}^d$, thus it holds that $|h(y) - h(x)| \leq 1/(2\lambda) \|x - y\|_1$

□

Theorem 2.3 allows us to directly compute the radius λ of the ℓ_1 ball $\mathcal{B}_1(x, \lambda)$ such that it is classified by classifier H with the same label as $H(x)$.

Corollary 2.4 (of Theorem 2.3)**.** *Let h be a smoothed classifier as in Theorem 2.3. Then H (thresholding h at 0.5) is robust (for class 1) at x with certified radius*

$$r(x) = 2\lambda(h(x) - 1/2).$$

Proof. Using $h(y) \geq h(x) - \frac{\|x-y\|_1}{2\lambda} > \frac{1}{2}$ from Theorem 2.3 and solving for $\|x - y\|_1$ yields the result. □

2.2. Box Constraints

In the case of ℓ_1-robustness, the most successful methods use the uniform distribution in an ℓ_∞-ball as smoothing distribution. Thus, we focus on this case first and discuss the others later. In Proposition 2.1 we have established that the overlap of supports of the smoothing distributions is a crucial factor for the robustness certification. In the upcoming proposition, we show that considering box-constraints gives us a tighter upper bound on the minimal possible overlap.

Proposition 2.5. *Let B_1, B_2 be the ℓ_∞ balls with radii λ centered at $x, y \in [0,1]^d$; then*

$$\frac{Vol(B_1 \cap B_2)}{Vol(B_1)} \geq \left(1 - \frac{1}{2\lambda}\right)^{\lfloor \|x-y\|_1 \rfloor} \left(1 - \frac{\|x-y\|_1 - \lfloor \|x-y\|_1 \rfloor}{2\lambda}\right) \geq \left(1 - \frac{1}{2\lambda}\right)^{\|x-y\|_1}.$$

The very last inequality holds when $2\lambda \geq 1$. Both of the inequalities are attainable.

Proof. The proof can be found in Appendix A.2. □

The aim was to provide simple expressions in Proposition 2.5 so that we can express the certified radius in a closed form. Specifically, we can plug the result from Proposition 2.5 into Proposition 2.1, resulting in Theorem 2.6 and Corollary 2.7. In Figure 2, we can see the improvement achieved compared to the certificates based on Propositions 2.2 and 2.1 resulting in Corollary 2.4.

Theorem 2.6. *Let $f : \mathbb{R}^d \to [0,1]$ be a deterministic or random classifier. Then the smoothed classifier is defined as:*

$$h(x) = \mathop{\mathbb{E}}_{\varepsilon \sim \mathcal{U}^d(\lambda)} [f(x+\varepsilon)].$$

Then for $x, y \in [0,1]^d$ it holds that

$$|h(x) - h(y)| \leq 1 - \left(1 - \frac{1}{2\lambda}\right)^{\|x-y\|_1}.$$

Proof. Plugging the bound from Proposition 2.5 into Proposition 2.1. □

Corollary 2.7 (of Theorem 2.6)**.** *Let h be a smoothed classifier as in Theorem 2.6. Then H (thresholding h at 0.5) is robust (for class 1) at x with certified radius*

$$r(x) = \frac{\ln(1.5 - h(x))}{\ln(1 - \frac{1}{2\lambda})}.$$

Proof. Using $h(x) - \frac{1}{2} \leq 1 - \left(1 - \frac{1}{2\lambda}\right)^{\|x-y\|_1}$ from Theorem 2.6 and solving for $\|x-y\|_1$ yields the result. □

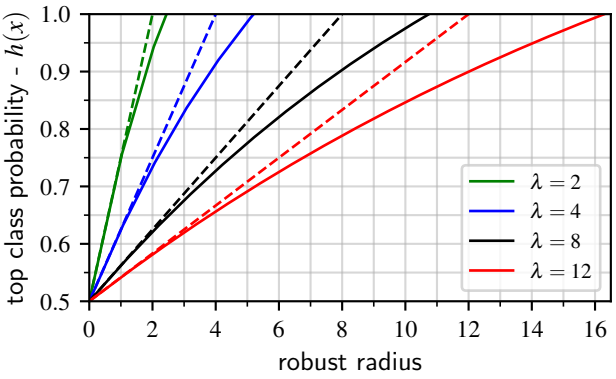

Figure 2: Conversion of top class probability of a smoothed classifier to the ℓ_1 certifiable robust radius for different noise magnitudes λ used for smoothing. The dashed lines are via Corollary 2.4 and the solid ones are via our Corollary 2.7.

We can further utilize the fact that the maximal possible difference between a potential adversarial example and the original image x at position i is at most $d_i = \max\{x_i, 1 - x_i\}$. In Proposition 2.5, we used an upper bound of $d_i \leq 1$. However, the following theorem establishes that for an image $x \in [0,1]^d$, we can find an image $y \in [0,1]^d$ that minimizes the intersection of $\mathcal{B}_\infty(x, \lambda)$ and $\mathcal{B}_\infty(y, \lambda)$ under the constraint $\|x-y\|_1 \leq c$ through a greedy coordinate-wise minimization approach. The improvements in certification are demonstrated in Example 2.9 and Figure 3.

Theorem 2.8. *Let $x \in [0,1]^d$. Let σ_i be an ordering induced by how far is x_i from boundary. That is:*

$$i \leq j \implies \min(x_{\sigma_i}, 1 - x_{\sigma_i}) \leq \min(x_{\sigma_j}, 1 - x_{\sigma_j}).$$

Then for any $c > 0$ such that there exists $y \in [0,1]^d$ with $\|x-y\|_1 = c$ it holds that

$$\inf_{y \in [0,1]^d \cap \mathcal{B}_1(x,c)} \frac{Vol(\mathcal{B}_\infty(x,\lambda) \cap \mathcal{B}_\infty(y,\lambda))}{Vol(\mathcal{B}_\infty(x,\lambda))} = \left(\prod_{i=1}^{T} \left(1 - \frac{\max\{x_{\sigma_i}, 1 - x_{\sigma_i}\}}{2\lambda}\right)\right)\left(1 - \frac{U}{2\lambda}\right)$$

where

$$T = \max_{k \in \mathbb{N}} \quad s.t. \sum_{i=1}^{i=k} \max(x_{\sigma_i}, 1 - x_{\sigma_i}) \leq c,$$

and

$$U = c - \sum_{i=1}^{i=T} \max(x_{\sigma_i}, 1 - x_{\sigma_i}).$$

Proof. The proof can be found in Appendix A.3 □

Theorem 2.8 is a clear generalization of Proposition 2.5 when we choose $x = \mathbf{0}$ in Theorem 2.8. Similarly, Proposition 2.2 can be seen as another corollary with a minor effort.

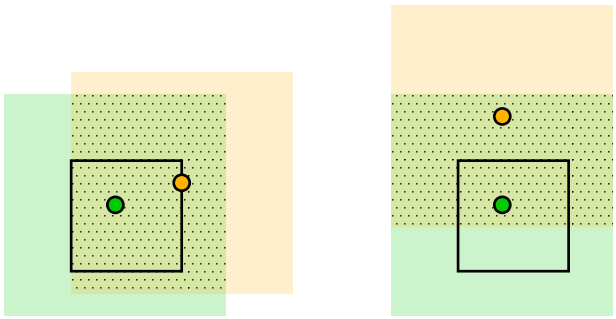

Figure 3: Effect of box constraints on the minimal overlap of two ℓ_∞ balls. The green point is at $(0.4, 0.6)$ while the orange one is at a distance 0.8 in ℓ_1-norm. On the left is depicted the minimal possible overlap considering box constraints (cf. Theorem 2.8), while on the right is the minimal possible overlap without considering box constraints (cf. Proposition 2.2). See Example 2.9 for further discussion.

Therefore, the proofs involve subtle variations of the same underlying idea.

Theorem 2.8 can be used for certification with Proposition 2.1. The certified radius cannot be expressed in a closed form but can be efficiently computed after sorting the coordinates since the volume of the intersection is an increasing piecewise-linear function of the robust radius.

Example 2.9. Consider a point $x = (0.4, 0.6)$ and a smoothing distribution uniform in $\mathcal{B}_\infty(0, 1)$, that is $\lambda = 1$. We want to certify the thresholded version of h when $h(x) = 0.88$. If we don't consider the box constraints, we can only certify robust radius via Corollary 2.4: $2\lambda(0.88 - 0.5) < 0.8$. However, if we consider the box constraints, we can certify robust radius 0.8 via Proposition 2.1 and Theorem 2.8. This is a consequence of the fact that the lower bound on the volume of intersections of two ℓ_∞ balls in Proposition 2.2 (in this case 0.6) is weaker than the (exact one) in Theorem 2.8 (in this case 0.63), see Figure 3 for the illustration.

2.3. Smoothing with Splitting Noise

An alternative ℓ_1-certification method proposed by Levine & Feizi (2021) uses a splitting noise. We show that if the splitting noise is independent in every dimension, then our certification from Corollary 2.7 can be directly applied here. For the simplicity, we introduce only the splitting noise with $\lambda \geq 0.5$ since the general version is more complicated and $0 < \lambda < 0.5$ can only be used to certify small radii; thus we do not improve the certification for that case. The fundamental concept behind the splitting noise is that, at every coordinate, we have two options: either we add uniform noise from the interval $[0, 1]$, or we set the value at that coordinate to 1. The strength of the noise λ determines the frequency at which each of these procedures occurs.

Theorem 2.10 (Theorem 2 of Levine & Feizi (2021))**.** *For any $f : \mathbb{R}^d \to [0, 1]$, and $\lambda \geq 0.5$ let $\mathbf{s} \in [0, 2\lambda]^d$ be a random variable whose (not necessarily independent) marginals follow the uniform distribution on $[0, 2\lambda]$. Then define*

$$\tilde{x}_i(s_i) = \min(s_i, 1) + [\![x_i > s_i]\!], \quad \forall i$$
$$h(x) = \mathop{\mathbb{E}}_{s}\left[f(\tilde{x}(s))\right].$$

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

Let us take a closer look at the distribution $\tilde{x}(s)$ for some $x \in [0, 1]$ and s uniformly drawn from interval $[0, 2\lambda]$. We split the inspection in three cases:

1. With probability $x/(2\lambda)$: $s \leq x$, then $s \leq 1$ and $\tilde{x}(s) = 1 + s$.
2. With probability $(1 - x)/(2\lambda)$: $x \leq s \leq 1$, then $\tilde{x}(s) = s$.
3. With probability $1 - 1/(2\lambda)$: $1 \leq s$, then $\tilde{x}(s) = 1$.

Thus, $\tilde{x}(s)$ is a mixture of a uniform distribution on $[x, 1+x]$ and a constant random variable at 1 with respective mixture coefficients $1/(2\lambda)$ and $1-1/(2\lambda)$ respectively. Thus, when $\lambda = 0.5$, the splitting noise distribution and uniform distribution in $\mathcal{B}_\infty(x, \lambda)$ are equal. Given this observation, it comes at no surprise that the techniques we used to improve the certification in the case of uniform noise in $\mathcal{B}_\infty(x, \lambda)$, resulting in Corollary 2.7, can be used also in the case of splitting noise.

Theorem 2.11. *Let the assumptions be as in Theorem 2.10 and additionally let the marginals of s be independent. Then Proposition 2.1 holds when the uniform noise is replaced by the splitting noise.*

Proof. Take $x, y \in \mathbb{R}^d$ and a noise sample $s \in \mathcal{U}^d(\lambda)$. At every position we have

$$\tilde{x}_i(s_i) = \min(s_i, 1) + [\![x_i > s_i]\!],$$

thus, in order to $\tilde{x}_i(s_i) \neq \tilde{y}_i(s_i)$, it has to be the case that

$$[\![x_i > s_i]\!] \neq [\![y_i > s_i]\!]$$

which happens with probability $\frac{|x_i - y_i|}{2\lambda}$. Let $R \subset [0, 2\lambda]^d$ such that for every $s \in R$ we have $\tilde{x}(s) = \tilde{y}(s)$. The probability that $\tilde{x}(s)$ and $\tilde{y}(s)$ are equal is exactly

$$\frac{Vol(R)}{(2\lambda)^d} = \prod_{i=1}^{d}\left(1 - \frac{|x_i - y_i|}{2\lambda}\right)$$
$$= \frac{Vol(\mathcal{B}_\infty(x, \lambda) \cap \mathcal{B}_\infty(y, \lambda))}{Vol(\mathcal{B}_\infty(x, \lambda))}$$

since s has independent marginals. Then we mimick the proof of Proposition 2.1:

$$\begin{aligned} h(y) &= \frac{\int_{s\in[0,2\lambda]^d} f(\tilde{y}(s))ds}{(2\lambda)^d} \geq \frac{\int_{s\in R} f(\tilde{y}(s))ds}{(2\lambda)^d} \\ &= \frac{\int_{s\in R} f(\tilde{x}(s))ds}{(2\lambda)^d} \\ &= \frac{\int_{s\in[0,2\lambda]^d} f(\tilde{x}(s))ds - \int_{s\in[0,2\lambda]^d\setminus R} f(\tilde{x}(s))ds}{(2\lambda)^d} \\ &\geq h(x) - \frac{(2\lambda)^d - Vol(R)}{(2\lambda)^d} \\ &= h(x) - 1 + \frac{Vol(\mathcal{B}_\infty(x,\lambda)\cap\mathcal{B}_\infty(y,\lambda))}{Vol(\mathcal{B}_\infty(x,\lambda))}. \end{aligned}$$

□

Theorem 2.11 shows that smoothing with both uniform and splitting noise are captured by Proposition 2.1. Thus, the certification methods from Corollaries 2.4 and 2.7 developed for uniform noise can be also used for the splitting noise and for the clarity, we will keep using uniform noise in the discussions.

2.3.1. Deterministic Splitting Noise

The splitting noise has another useful property. If the noise in different coordinates is not independent, then we can evaluate the expectation exactly using $2q\lambda$ evaluations of the base classifier, where q here stands for the number of quantization levels which is commonly 256. We refer the reader to Levine & Feizi (2021) for the details. This has two benefits; one, the provided certificates are deterministic and second, they are faster to compute - although this is not inherent. We can use less samples to estimate the expectation of the base classifier under the smoothing noise. See Subsection 2.7 and Figure 4 for more details. Our method cannot be applied in this deterministic case because we can no longer rely on the independence of the splitting noise across different coordinates.

2.4. Multiclass Classification

We introduced the randomized smoothing machinery for the task of binary classification. In the K-class setting we define randomized smoothing as follows; let $f : \mathbb{R}^d \to \{e_1, e_2, \dots e_K\}$ where e_i are one-hot vectors with 1 at position i be the base classifier. The smoothed classifier is then

$$h(x) = \mathop{\mathbb{E}}_{\varepsilon\sim q} f(x+\varepsilon)$$

and we certify robustness for its thresholded version; i.e., for

$$H(x) = \arg\max_{i=1}^{K} h(x)_i.$$

A possible approach to the multiclass setting is straightforward; just consider all the other classes as a one big class. That is, treat the multiclass classifier $f(x)$ as if it would be a binary classifier $f(x)_y$ when certifying class y. This approach is commonly taken in the randomized smoothing literature (Salman et al., 2019; Cohen et al., 2019) to avoid problems with estimation of class probabilities. Thus, we can directly use all the theory we have developed so far for the binary classification. However, this simplification may come at a high cost. Consider for example a classification task with $k = 1000$ classes with $h(x)_1 = 0.4$ and $h(x)_i < 0.001$ for the other classes. Then with the discussed conversion we are not able to certify class 1. However, as we will see, H can be moderately robust at x.

Proposition 2.12. *Let $f : \mathbb{R}^d \to \{e_1, e_2, \dots e_K\}$ be a base classifier, its smoothed version be h and H be the thresholded version of h and the smoothing distribution be $\mathcal{U}^d(\lambda)$. Let also $H(x) = A$. Then for any point y it holds that*

$$\frac{h(y)_A - h(y)_B}{2} \geq \frac{h(x)_A - h(x)_B}{2} - 1 + \frac{Vol(B_1\cap B_2)}{Vol(B_2)}$$

where B is an arbitrary class and $B_1 = \mathcal{B}_\infty(x,\lambda)$ and $B_2 = \mathcal{B}_\infty(y,\lambda)$,

Proof. We subtract the inequalities from Proposition 2.1 applied to $h(\cdot)_A$ and $h(\cdot)_B$. □

In order to certify the thresholded classifier in the multiclass setting, we have to ensure that $\frac{h(y)_A - h(y)_B}{2} \geq 0$ in the notation of Proposition 2.12.

Corollary 2.13 (of Proposition 2.12 and 2.2)**.** *Let the notation be as in Proposition 2.12. Then H is certifiably robust at x with robust radius*

$$r(x) = \lambda\left(h(x)_A - h(x)_B\right),$$

where $H(x) = A$ and B is the runner-up class.

Corollary 2.14 (of Proposition 2.12 and 2.5)**.** *Let the notation be as in Proposition 2.12. Then H is certifiably robust at x with robust radius*

$$r(x) = \frac{\ln\left(1 - \frac{h(x)_A - h(x)_B}{2}\right)}{\ln\left(1-\frac{1}{2\lambda}\right)}$$

where $H(x) = A$ and B is the runner-up class.

2.5. Algorithmic Implementation

We have covered the mathematical foundations of randomized smoothing. However, the exact expectation in Equation (1) is usually intractable to evaluate; therefore, Monte Carlo

sampling is used to estimate it (whence *randomized* smoothing). As a consequence, the technique is stochastic and we cannot guarantee that it will always produce a valid radius. Still, we can control the probability of bad luck during sampling and the certificates are (by design) computed so that they are correct with $1-\alpha = 99.9\%$ probability. We discuss the procedure in Section 2.6. We emphasize that the certificates are for the actual classifier induced by h defined using an expectation. For the same reason, we should be careful with querying the classifier h - it brings another source of randomness. To control all this randomness, one commonly uses $n = 100\,000$ samples to estimate $h(x)$ in Equation (1) which leads to long certification and inference times. However, it is not necessary as we will discuss in Section 2.7

The choice of a smoothing distribution q is crucial for the performance. Some popular choices are normal distribution (for ℓ_2- and ℓ_∞-threat models) and uniform distributions in ℓ_p balls; see (Yang et al., 2020) for a thorough inspection of many smoothing distributions and respective certifications or (Dvijotham et al., 2020) for a general certification framework that is applicable for virtually any smoothing distribution. Specially, for smoothing with normal distribution with covariance matrix $\sigma^2 I_d$, it was shown by Cohen et al. (2019) that h is robust at a with radius $\sigma\Phi^{-1}(h(a))$ under the ℓ_2-threat model, where Φ^{-1} is the quantile function of standard normal distribution. For the ℓ_∞-threat model the radius $\sigma\Phi^{-1}(h(a))/\sqrt{d}$ can be certified as it fits into a ℓ_2-ball of radius $\sigma\Phi^{-1}(h(a))$.

2.6. Controlling Failure Probability

In order to provide high probability certificates, we need to estimate some of the class probabilities. A standard way to perform certification is by first evaluating a few (usually $n_0 = 64$, however, we use 256 in the experiments) noisy samples to estimate the top-1 class of $h(x)$ that is certified in a second step via one-versus-all approach as discussed in Section 2.4. This approach has the benefit that one only needs to estimate the parameter of a binomial distribution and one can easily control the failure probability via Clopper-Pearson tail bounds. However, this approach has its downsides pointed out in Subsection 2.4. Thus, we will follow Proposition 2.12 for the certification. We need to estimate not just top-1 class probability, but also the top-2 class probability. We use the standard Bonferroni correction to estimate them. That is, we estimate both, top-1 and top-2 class probability with allowed failure probability $\alpha/2$ via Clopper-Pearson tail bounds. Therefore, by a union bound, both of the estimated values are simultaneously correct with probability at least $1-\alpha$.

See Algorithm 1 for the actual certification via Corollary 2.14. We note that there is no guarantee that $\hat{A}$ nor $\hat{B}$ correspond to the actual two most probable classes. However, it does not matter. The value p_A does not exceed $h(x)_A$ with probability at least $1-\alpha/2$; thus, it also cannot exceed $\max_k h(x)_k$. Similarly, even if the actual top-2 class is not $\hat{B}$, then p_B is overestimating $h(x)_B$ with probability at least $1-\alpha/2$. Thus, the failure probability is at most α. The function LOWCONFBOUND (resp. UPPCONFBOUND) computes lower (resp. upper) confidence interval for $h(x)_A$ (resp. $h(x)_B$). We discuss its implementation in Subsection 3.2.

Algorithm 1 Randomized Smoothing Certification

procedure SAMPLEUNDERNOISE(f, x, n, λ)
  `counts` $\leftarrow [0, 0, \ldots, 0]$
  for $i \leftarrow 1, n$ **do**
    $x' \leftarrow$ `noise`(x, λ)
    `counts` $\leftarrow$ `counts` $+ f(x')$
  return `counts`

procedure CERTIFY($f, x, n_0, n, \lambda, \alpha$)
  `counts0` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n_0, λ)
  $\hat{A} \leftarrow$ top index in `counts0`
  `counts` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n, λ)
  $\hat{B} \leftarrow$ top index in `counts` but not $\hat{A}$
  $p_A \leftarrow$ LOWCONFBOUND(`counts`$[\hat{A}], n, \alpha/2)$
  $p_B \leftarrow$ UPPCONFBOUND(`counts`$[\hat{B}], n, \alpha/2)$
  if $p_A > p_B$ **then**
    return prediction $\hat{A}$ and radius

$$\frac{\ln\left(1-\frac{p_A-p_B}{2}\right)}{\ln\left(1-\frac{1}{2\lambda}\right)}$$

  else
    return ABSTAIN

2.7. Influence of the Number of Samples

We discuss the influence of the number of samples used to estimate the output of a smoothed classifier and the required confidence on the certified accuracy. In the ℓ_1-case that we have covered, certification using Corollaries 2.4, 2.7 and their multiclass counterparts scales roughly linearly in the estimated probability

$$\ln\left(\frac{3}{2}-h(x)-\epsilon\right) \geq \ln\left(\frac{3}{2}-h(x)\right) - 2\epsilon$$

for $\epsilon > 0$ and $h(x)+\epsilon \leq 1$. We used that $\sup_{x\in[\frac{1}{2},\frac{3}{2}]} \ln'(x) = 2$. The width of confidence intervals (for $p \geq 0.5$) when $p \approx 1$ scales roughly as $\frac{-\ln(\alpha)}{n}$ for confidence level α, while when $p \not\approx 1$, the width scales as $\sqrt{\frac{-\ln(\alpha)}{n}}$ as follows from the Bennett's inequality. A (sub

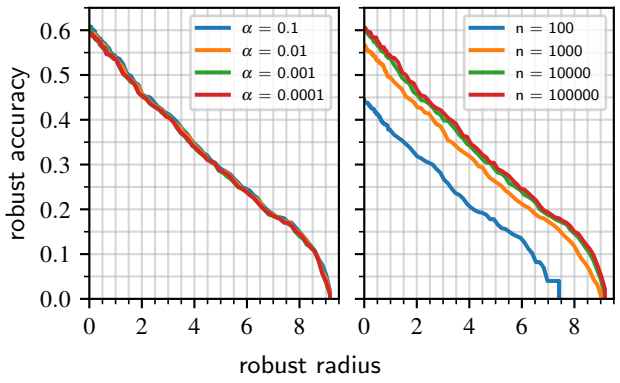

Figure 4: Effect of the choice of α (on the left; $n = 10\,000$) and of n (on the right; $\alpha = 0.001$) on the ℓ_1-robustness curve. Experiment is on CIFAR10 with $\lambda = 6.92$ and certification according to Corollary 2.7.

optimal) universal bound by Hoeffding's inequality yields

$$t = \sqrt{\frac{\ln(\alpha)}{-2n}}$$

as a width of the (one-sided) confidence interval; see Appendix B for the derivation. Thus, the estimation error of the robust radius gecreases at least as $\sqrt{\frac{-\ln \alpha}{n}}$ and significantly faster when the estimated probability is close to 1.

In practice, it might not be necessary to push for the largest n possible. See Figure 4 for certification with different choices of numbers of noise samples and choices of α to get an impression of how much the robustness curves are influenced by these hyperparameters.

We note that this finding does not transfer to ℓ_2- and ℓ_∞-smoothing where the normal distribution is dominantly used as a smoothing distribution. This is because for these cases the certificates are not linear in the estimated probability. They are computed as $\sigma\Phi^{-1}(p)$, where Φ^{-1} grows arbitrarily steeply as p approaches 1. For example, if we have a constant base classifier, $\sigma = 1$, and $n \in \{10\,000, 100\,000, 1\,000\,000\}$, then we can certify radii 3.20, 3.81 and 4.35 respectively which makes a huge difference. See Figure 8 of (Cohen et al., 2019) for more details. To conclude this subsection, we demonstrated that ℓ_1 certification via randomized smoothing requires significantly less samples than in the ℓ_2 and ℓ_∞ cases in order to reasonably controll the estimation error of a robust radius.

3. Experiments

We performed an extensive evaluation on CIFAR-10 (Krizhevsky et al., 2009) and ImageNet-1k (Russakovsky et al., 2015) and demonstrate the improvements comparing to (Levine & Feizi, 2021) and (Yang et al., 2020). To ensure a fairness of the evaluation, we follow the experimental setup that is identical in both mentioned papers but we perform it over a wider range of smoothing distribution parameters. Following previous work, we report standard deviations of the distribution instead of the λ parameter that we have used throughout the paper. The conversion is that σ corresponds to $\lambda := \sqrt{3}\sigma$. Namely, for CIFAR-10, we evaluated on $\sigma \in \{0.15, 0.25, \ldots 3.5, \ldots 8, 9, 10, 12\}$, where in the first gap, the spacing is 0.25 and in the second it is 0.5 and for ImageNet we used $\sigma \in \{0.5, 1.25, 2, 2.75, 3.5, 4.5, 5.5\}$.

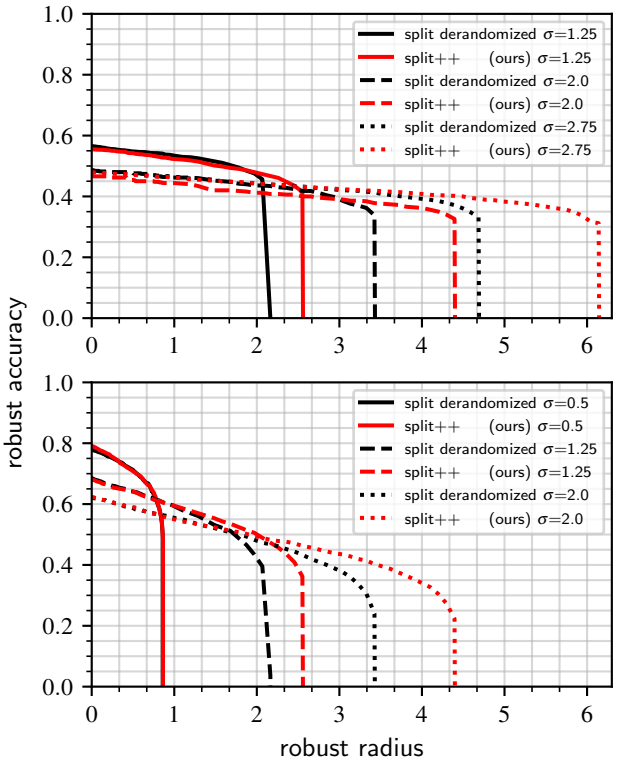

Figure 5: Comparison of certification using deterministic splitting noise and with independent splitting noise and our improved certification. The models were trained with stability training. ImageNet is shown in the top figure, CIFAR-10 in the bottom figure.

3.1. Training

For ImageNet we used a ResNet-50 trained for 30 epochs; and for CIFAR-10 we used a WideResNet-40-2 trained for 120 epochs. The optimizer is SGD with learning rate 0.1, Nesterov momentum 0.9 and weight decay 0.0001 with cosine annealing learning rate schedule and batch size is 64 for both models. We experimented with the two following types of training (with a slight abuse of notation in the case of splitting noise):

1. standard training with cross entropy loss

$$\mathcal{L}(x, y) = -\log\big(f(x+\delta)_y\big), \quad \delta \sim q$$

on noise-augmented data points as suggested in (Cohen et al., 2019).

2. stability training (Li et al., 2019) (is roughly twice as expensive) with loss

$$\begin{aligned}\mathcal{L}(x, y) =& C \cdot \mathrm{KL}\big(f(x+\delta_1) \parallel f(x+\delta_2)\big) \\ & - \log\big(f(x+\delta_1)_y\big), \quad \delta_1, \delta_2 \sim q\end{aligned}$$

where C is a hyperparameter chosen as $C = 6$ following (Carmon et al., 2019).

In the case when the smoothing distribution q is a uniform distribution, upgrading standard training to stability training helps significantly as observed by Levine & Feizi (2021) and Yang et al. (2020). However, for the splitting noise the benefits are less apparent and sometimes it even hurts. Nevertheless, according to our experiments, for every radius at which we evaluate robustness, the best performing model was trained with stability training.

3.2. Certification Results

Following the literature, we set the probability of certificate being incorrect to be at most $\alpha = 0.001$ for all methods. The only exception is smoothing with deterministic splitting noise which is always correct. Unless stated otherwise, we use 10 000 noise samples for the certification and 256 to estimate the top-1 class. The confidence intervals are computed using Python function `proportion_confint` from package `statsmodels.stats.proportion` implementing the Clopper-Pearson method. We call the method with $\alpha = 0.002$ because the confidence interval returned is central and the coverage is $1 - \alpha/2$ in both tails. For CIFAR-10 dataset we certify 2 000 images from the test set, while for ImageNet we certify the same subset of 500 images as Cohen et al. (2019) and Levine & Feizi (2021).

In Figure 5 we empirically demonstrate that with the improved certification we are able to certify significantly larger ℓ_1-radii both on ImageNet and CIFAR-10. In Appendix C, there is an additional extensive comparison of the proposed method with the current state of the art.

In Figure 6 we show how the choice of the certification scheme (binary or multiclass) affects the robustness curve. This explains why we observe a similar performance of splitting noise method with its derandomized counterpart, while Levine & Feizi (2021) (who used the binary certification scheme) observed significantly weaker performance.

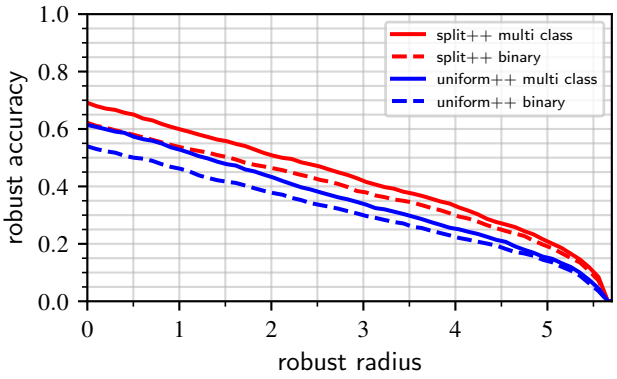

Figure 6: Comparison of the robustness curves for the binary (Corollary 2.7) and multiclass (Corollary 2.14) certification approach. Note that the multiclass approach (almost strictly) outperforms the binary one. Certification was done by our methods, but the conclusion holds for the standard bounds from Corollary 2.4 and 2.13 as well. The setting is standard training, CIFAR-10, $\sigma = 2.5$.

4. Conclusions

In this paper we have shown that incorporating the constraint of image classifiers that input points have to lie in the image domain $[0, 1]^d$ leads to significantly improved certified ℓ_1-radii. The application of our framework is essentially for free and can be directly applied to randomized smoothing using uniform or splitting noise. Our experiments show that we can certify significantly larger ℓ_1-radii than previous work but there still remains a gap to what seems possible in empirical ℓ_1-robustness which is an interesting question for future research for both empirical and certified robustness.

Acknowledgements

The authors thank the anonymous reviewers for their comments, which helped improve the quality of the manuscript. The authors acknowledge support from the DFG Cluster of Excellence "Machine Learning – New Perspectives for Science", EXC 2064/1, project number 390727645 and the Carl Zeiss Foundation in the project "Certification and Foundations of Safe Machine Learning Systems in Healthcare". The authors are thankful for the support of Open Philanthropy.

References

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks

against machine learning at test time. In *ECML*.

Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. In *NeurIPS*, 2019.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Croce, F. and Hein, M. Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers. In *ICML*, 2021.

Croce, F. and Hein, M. Adversarial robustness against multiple ℓ_p-threat models at the price of one and how to quickly fine-tune robust models to another threat model. In *ICML*, 2022.

Dvijotham, K. D., Hayes, J., Balle, B., Kolter, Z., Qin, C., Gyorgy, A., Xiao, K., Gowal, S., and Kohli, P. A framework for robustness certification of smoothed classifiers using f-divergences. In *ICLR*, 2020.

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

Kantchelian, A., Tygar, J., and Joseph, A. Evasion and hardening of tree ensemble classifiers. In *ICML*, 2016.

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In *CAV*, 2017.

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

Laidlaw, C., Singla, S., and Feizi, S. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021.

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

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

Leino, K., Wang, Z., and Fredrikson, M. Globally-robust neural networks. In *ICML*, 2021.

Levine, A. and Feizi, S. Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. In *AISTATS*, 2020.

Levine, A. J. and Feizi, S. Improved, deterministic smoothing for ℓ_1 certified robustness. In *ICML*, 2021.

Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. In *NeurIPS*, 2019.

Lin, W., Lucas, K., Bauer, L., Reiter, M. K., and Sharif, M. Constrained gradient descent: A powerful and principled evasion attack against neural networks. In *ICML*, 2022.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Metzen, J. H. and Yatsura, M. Efficient certified defenses against patch attacks on image classifiers. In *ICLR*, 2021.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

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 *NeurIPS*, 2019.

Salman, H., Jain, S., Wong, E., and Madry, A. Certified patch robustness via smoothed vision transformers. In *CVPR*, 2022.

Saralajew, S., Holdijk, L., and Villmann, T. Fast adversarial robustness certification of nearest prototype classifiers for arbitrary seminorms. In *NeurIPS*, 2020.

Singla, S., Singla, S., and Feizi, S. Improved deterministic l_2 robustness on CIFAR-10 and CIFAR-100. In *ICLR*, 2022.

Steele, J. M. The cauchy-schwarz master class: An introduction to the art of mathematical inequalities, 2004.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Tjeng, V. and Tedrake, R. Verifying neural networks with mixed integer programming. preprint, arXiv:1711.07356v1, 2017.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020.

Trockman, A. and Kolter, J. Z. Orthogonalizing convolutional layers with the cayley transform. In *ICLR*, 2021.

Voráček, V. and Hein, M. Provably adversarially robust nearest prototype classifiers. In *ICML*, 2022.

Wang, L., Liu, X., Yi, J., Zhou, Z.-H., and Hsieh, C.-J. Evaluating the robustness of nearest neighbor classifiers: A primal-dual perspective. *arXiv preprint, arXiv:1906.03972*, 2019.

Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *NeurIPS*, 2018.

Wong, E., Schmidt, F., and Kolter, Z. Wasserstein adversarial examples via projected sinkhorn iterations. In *ICML*, 2019.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I., and Li, J. Randomized smoothing of all shapes and sizes. In *ICML*, 2020.

Zhang, B., Jiang, D., He, D., and Wang, L. Boosting the certified robustness of l-infinity distance nets. In *ICLR*, 2022a.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In *NeurIPS*, 2018.

Zhang, H., Wang, S., Xu, K., Li, L., Li, B., Jana, S., Hsieh, C.-J., and Kolter, J. Z. General cutting planes for bound-propagation-based neural network verification. In *NeurIPS*, 2022b.

A. Proofs of Proposition 2.2,2.5 and Theorem 2.8

The central quantity in the proofs is

$$\frac{Vol(\mathcal{B}_\infty(0,\lambda)\cap\mathcal{B}_\infty(z,\lambda))}{Vol(\mathcal{B}_\infty(0,\lambda))}=\prod_{i=1}^{d}\left(\frac{2\lambda-z_i}{2\lambda}\right)=\prod_{i=1}^{d}\left(1-\frac{z_i}{2\lambda}\right) \tag{3}$$

for $z\in[0,2\lambda]^d$. We show that (3) is a Schur-concave function and is therefore minimized by a maximal element w.r.t. the majorization order. The following definitions and propositions are adopted from Steele (2004).

Definition A.1. Let $x,y\in\mathbb{R}^d$. We write $x\succeq y$ (x *weakly majorizes* y) if for all $1\leq k\leq d$ it holds that

$$\sum_{i=1}^{k}x_i\geq\sum_{i=1}^{k}y_i.$$

If further $\sum_{i=1}^{d}x_i=\sum_{i=1}^{d}y_i$, we write $x\succ y$ (x majorizes y).

Definition A.2. A function $f:\mathcal{X}\to\mathbb{R}$ is said to be Schur-concave if for all $x,y\in\mathcal{X}$ such that $x\succ y$ it holds that $f(x)\leq f(y)$.

Proposition A.3. *Let $f:\mathcal{X}\to\mathbb{R}$ be a differentiable symmetric function. Then it is Schur-concave if*

$$(x_i-x_j)\left(\frac{\partial f}{\partial x_i}-\frac{\partial f}{\partial x_j}\right)\leq 0.$$

Proposition A.4. *Function (3) is Schur-concave. It further holds that for all x,y such that $x\succeq y$, $f(x)\geq f(y)$.*

Proof. Let $X=(0,2\lambda)^d$. Function f is positive; thus is Schur-concave on X if and only if $g(x)=\log(f(x))$ is Schur-concave since log is an increasing function. Then

$$(x_i-x_j)\left(\frac{\partial g}{\partial x_i}-\frac{\partial g}{\partial x_j}\right)=(x_i-x_j)\left(\frac{1}{x_i-2\lambda}-\frac{1}{x_j-2\lambda}\right)\leq 0$$

because $x_i\geq x_j\iff 0>x_i-2\lambda\geq x_j-2\lambda\iff\frac{1}{x_i-2\lambda}\leq\frac{1}{x_j-2\lambda}$.

Since f is continuous on X and symmetric, f is Schur-concave on $[0,2\lambda]^d$. The second claim follows since f is a decreasing function in every coordinate on $[0,2\lambda]^d$. □

A.1. Proof of Proposition 2.2

Proposition A.5. *Let B_1,B_2 be ℓ_∞-balls with radii λ centered at $x,y\in\mathbb{R}^d$ respectively; then*

$$\frac{Vol(B_1\cap B_2)}{Vol(B_1)}\geq 1-\frac{\|x-y\|_1}{2\lambda}.$$

Proof. Let $x,y\in\mathbb{R}^d$ and $c=\|x-y\|_1$. It holds that

$$\begin{aligned}
&\inf_{z\in\mathbb{R}^d_+\cap\langle\mathbf{1},z\rangle\leq c}\prod_{i=1}^{d}\left(1-\frac{z_i}{2\lambda}\right)\\
=&\inf_{u,v\in\mathbb{R}^d\cap\|u-v\|_1\leq c}\prod_{i=1}^{d}\left(1-\frac{|u_i-v_i|}{2\lambda}\right)\\
\leq&\prod_{i=1}^{d}\left(1-\frac{|x_i-y_i|}{2\lambda}\right)\\
=&\frac{Vol(\mathcal{B}_\infty(x,\lambda)\cap\mathcal{B}_\infty(y,\lambda))}{Vol(\mathcal{B}_\infty(x,\lambda))}\\
=&\frac{Vol(B_1\cap B_2)}{Vol(B_1)}
\end{aligned}$$

so it is sufficient to show that

$$\inf_{z\in\mathbb{R}^d_+\cap\langle\mathbf{1},z\rangle\leq c}\prod_{i=1}^{d}\left(1-\frac{z_i}{2\lambda}\right)=1-\frac{\|x-y\|_1}{2\lambda}. \tag{4}$$

The objective of optimization problem in (4) equals to (3) and is Schur-concave according to Proposition A.4. Thus is minimized by (e.g.,) $z=(c,0,0,\dots)$. In that case, the value of the objective is $1-\frac{c}{2\lambda}=1-\frac{\|x-y\|_1}{2\lambda}$. □

A.2. Proof of Proposition 2.5

Proposition A.6. *Let B_1, B_2 be the ℓ_∞ balls with radiii λ centered at $x, y\in[0,1]^d$; then*

$$\begin{aligned}&\frac{Vol(B_1\cap B_2)}{Vol(B_1)}\geq\\&\left(1-\frac{1}{2\lambda}\right)^{\lfloor\|x-y\|_1\rfloor}\left(1-\frac{\|x-y\|_1-\lfloor\|x-y\|_1\rfloor}{2\lambda}\right)\\&\geq\left(1-\frac{1}{2\lambda}\right)^{\|x-y\|_1}.\end{aligned}$$

The very last inequality holds when $2\lambda\geq 1$. Both of the inequalities are attainable.

Proof. Let $x, y\in[0,1]^d$ and $c=\|x-y\|_1$. It holds that

$$\begin{aligned}&\inf_{z\in[0,1]^d\cap\langle\mathbf{1},z\rangle\leq c}\prod_{i=1}^{d}\left(1-\frac{z_i}{2\lambda}\right)\\=&\inf_{u,v\in[0,1]^d\cap\|u-v\|_1\leq c}\prod_{i=1}^{d}\left(1-\frac{|u_i-v_i|}{2\lambda}\right)\\\leq&\prod_{i=1}^{d}\left(1-\frac{|x_i-y_i|}{2\lambda}\right)\\=&\frac{Vol(\mathcal{B}_\infty(x,\lambda)\cap\mathcal{B}_\infty(y,\lambda))}{Vol(\mathcal{B}_\infty(x,\lambda))}\\=&\frac{Vol(B_1\cap B_2)}{Vol(B_1)}\end{aligned}$$

so it is sufficient to show that

$$\inf_{z\in\mathbb{R}^d_+\cap\langle\mathbf{1},z\rangle\leq c}\prod_{i=1}^{d}\left(1-\frac{z_i}{2\lambda}\right)=\left(1-\frac{1}{2\lambda}\right)^{\lfloor\|x-y\|_1\rfloor}\left(1-\frac{\|x-y\|_1-\lfloor\|x-y\|_1\rfloor}{2\lambda}\right). \tag{5}$$

The objective of optimization problem in (5) equals to (3) and is Schur-concave according to Proposition A.4. Thus, it is minimized by a vector z such that $z_i=1$ at $\lfloor c\rfloor$ positions and $z_i=c-\lfloor c\rfloor$ at another position which is clearly a maximal element w.r.t. the majorization order. In that case, the value of the objective is

$$\left(1-\frac{1}{2\lambda}\right)^{\lfloor c\rfloor}\left(1-\frac{c-\lfloor c\rfloor}{2\lambda}\right).$$

If $x=\mathbf{0}$ and $y=z$, the inequality is tight. Furthermore, due to the convexity of the exponential function, we have for $a\geq 1$ and $0\leq x\leq 1$ that

$$\left(1-\frac{1}{a}\right)^{x}\leq 1-\frac{x}{a}.$$

Thus, we can simplify (5) to

$$\inf_{z\in\mathbb{R}^d_+\cap\langle\mathbf{1},z\rangle\leq c}\prod_{i=1}^{d}\left(1-\frac{z_i}{2\lambda}\right)\geq\left(1-\frac{1}{2\lambda}\right)^{\|x-y\|_1}$$

finishing the proof. □

A.3. Proof of Theorem 2.8

Theorem A.7. *Let $x \in [0,1]^d$. Let σ_i be an ordering induced by how far is x_i from boundary. That is;*

$$i \leq j \implies \min(x_{\sigma_i}, 1 - x_{\sigma_i}) \leq \min(x_{\sigma_j}, 1 - x_{\sigma_j}).$$

Then for any $c > 0$ such that there exists $y \in [0,1]^d$ with $\|x - y\|_1 = c$ it holds that

$$\inf_{y \in [0,1]^d \cap \mathcal{B}_1(x,c)} \frac{Vol(\mathcal{B}_\infty(x,\lambda) \cap \mathcal{B}_\infty(y,\lambda))}{Vol(\mathcal{B}_\infty(x,\lambda))} = \left(\prod_{i=1}^{T} \left(1 - \frac{\max\{x_{\sigma_i}, 1 - x_{\sigma_i}\}}{2\lambda}\right)\right)\left(1 - \frac{U}{2\lambda}\right)$$

where

$$T = \max_{k \in \mathbb{N}} \quad s.t. \sum_{i=1}^{i=k} \max(x_{\sigma_i}, 1 - x_{\sigma_i}) \leq c,$$

and

$$U = c - \sum_{i=1}^{i=T} \max(x_{\sigma_i}, 1 - x_{\sigma_i}).$$

Proof. We equivalently rewrite the problem as

$$\inf_{z \in \mathcal{X} \cap \langle \mathbf{1}, z \rangle = c} \frac{Vol(\mathcal{B}_\infty(x,\lambda) \cap \mathcal{B}_\infty(y,\lambda))}{Vol(\mathcal{B}_\infty(x,\lambda))} \tag{6}$$

where $\mathcal{X} = \bigtimes_{i=1}^{d} [0, \max\{x_i, 1 - x_i\}]$ so that from z we recover y as $y_i = x_i \pm z_i$ where either $0 \leq x_i + z_i \leq 1$ or $0 \leq x_i - z_i \leq 1$.

The objective of the optimization problem (6) is again (3). Thus, we only need to find a maximizing element w.r.t. the majorization order and the Theorem describes how to find it. To see that, we notice that every $1 \leq k \leq d$, the sequence

$$z_i = \begin{cases} \max\{x_i, 1 - x_i\}, & \text{if } \sigma_i^{-1} < T + 1 \\ U, & \text{if } \sigma_i^{-1} = T + 1 \\ 0, & \text{if } \sigma_i^{-1} > T + 1 \end{cases}$$

where σ_i^{-1} is the inverse ordering, that is, $\sigma_{\sigma_i^{-1}} = i$, maximizes $\sum_{i=1}^{k} z_i$ under the constraint $\sum_{i=1}^{d} z_i = c$. □

B. Hoeffding's bound

Theorem B.1 (Hoeffding's inequality)**.** *Let $X_1, \dots X_n$ be random variables with $\frac{1}{n}\mathbb{E}[\sum_{i=1}^{n} X_i] = \mu$ and $0 \leq X_i \leq 1$. Then it holds that*

$$\mathbb{P}\left(\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) - \mu \geq t\right) \leq e^{-2t^2 n}.$$

for any $t \geq 0$.

We rewrite the inequality as

$$\mathbb{P}\left(\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) - t \geq \mu\right) \leq e^{-2t^2 n} \leq \alpha.$$

We want to compute t - that is, how much do we need to subtract from the average so that the probability that the result of the subtraction will be larger than the mean is small.

$$e^{-2t^2 n} = \alpha \quad \implies \quad t = \sqrt{\frac{\ln(\alpha)}{-2n}}$$

C. Ablations

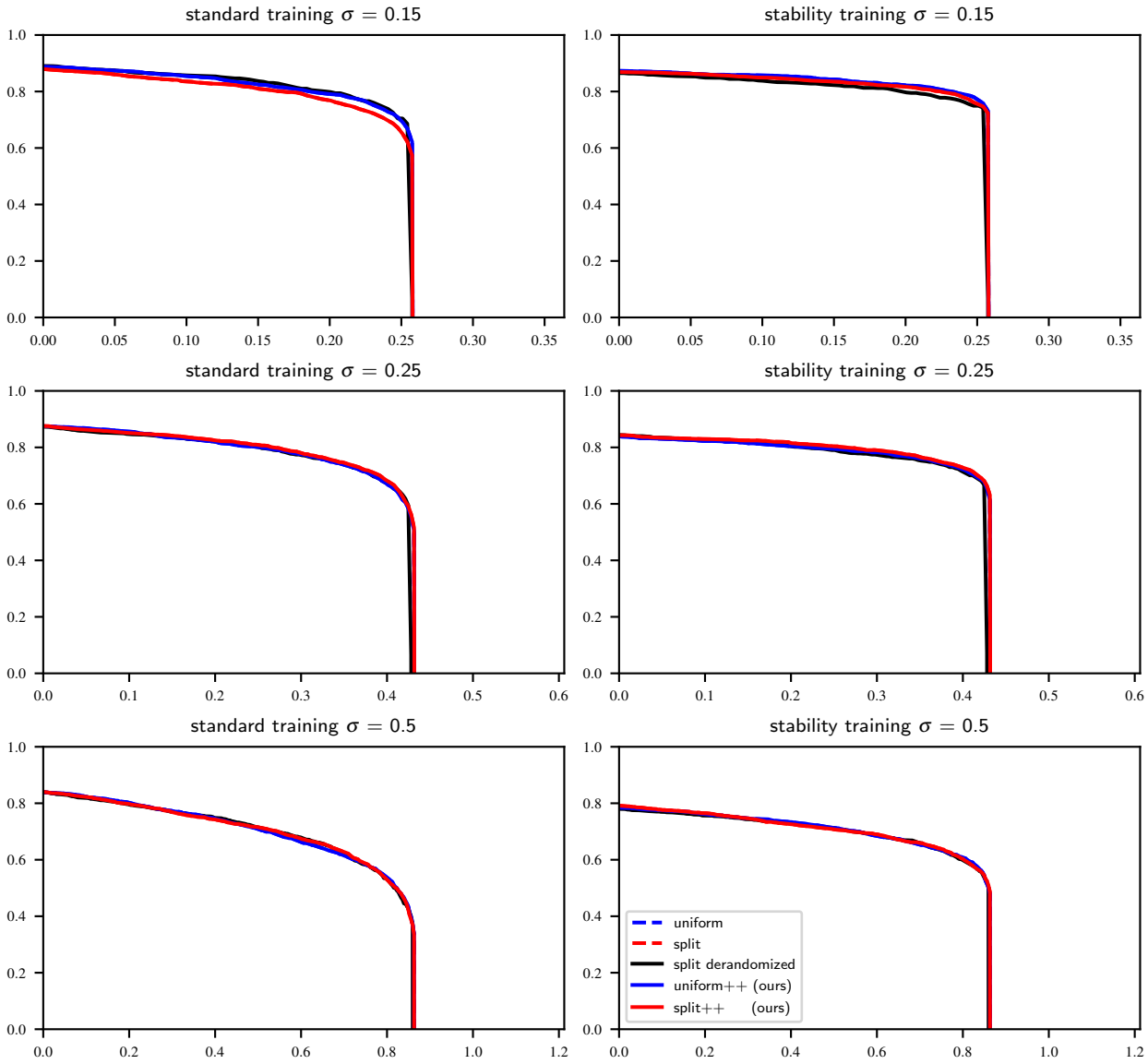

Figure 7: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns.

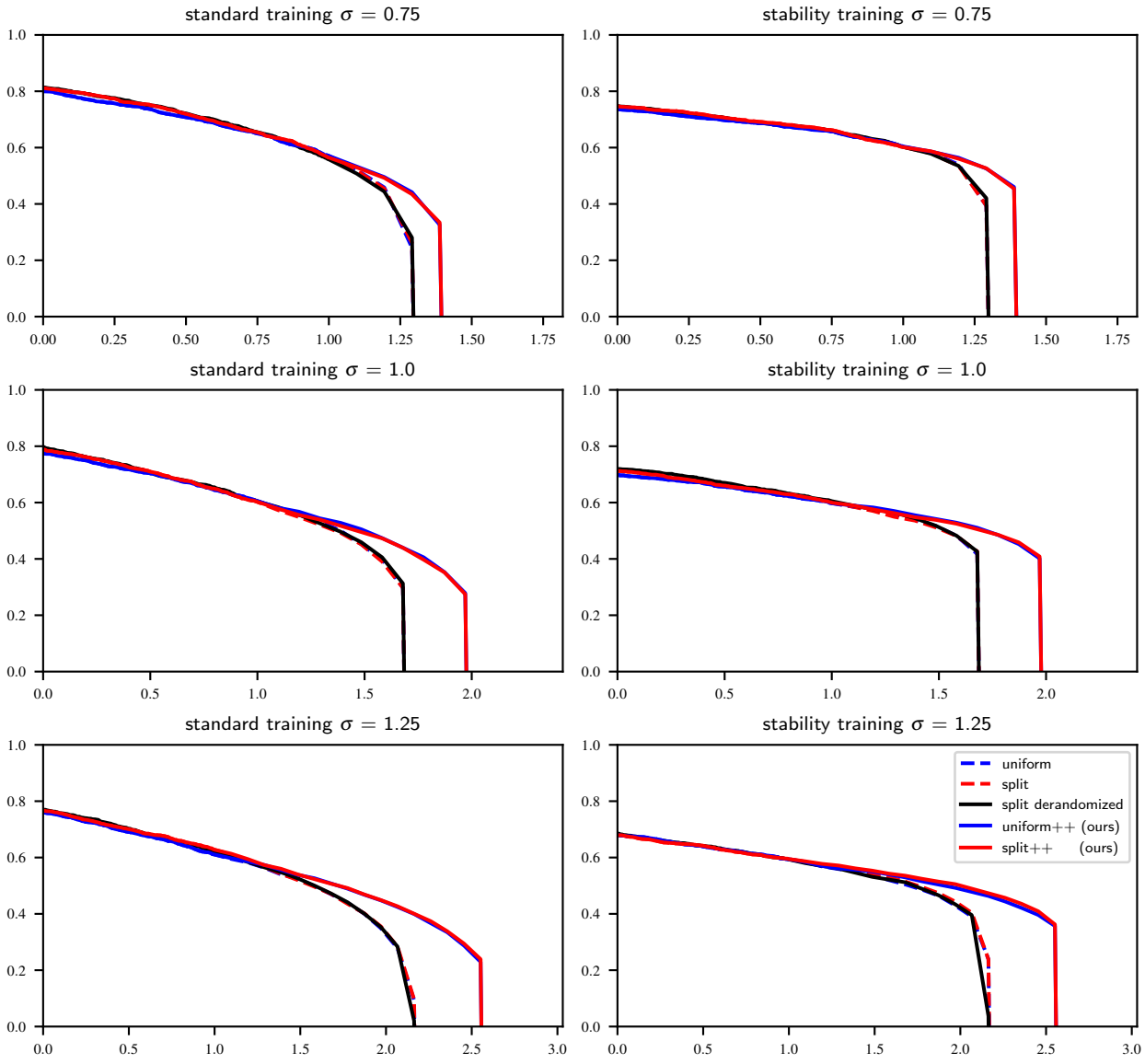

Figure 8: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns.

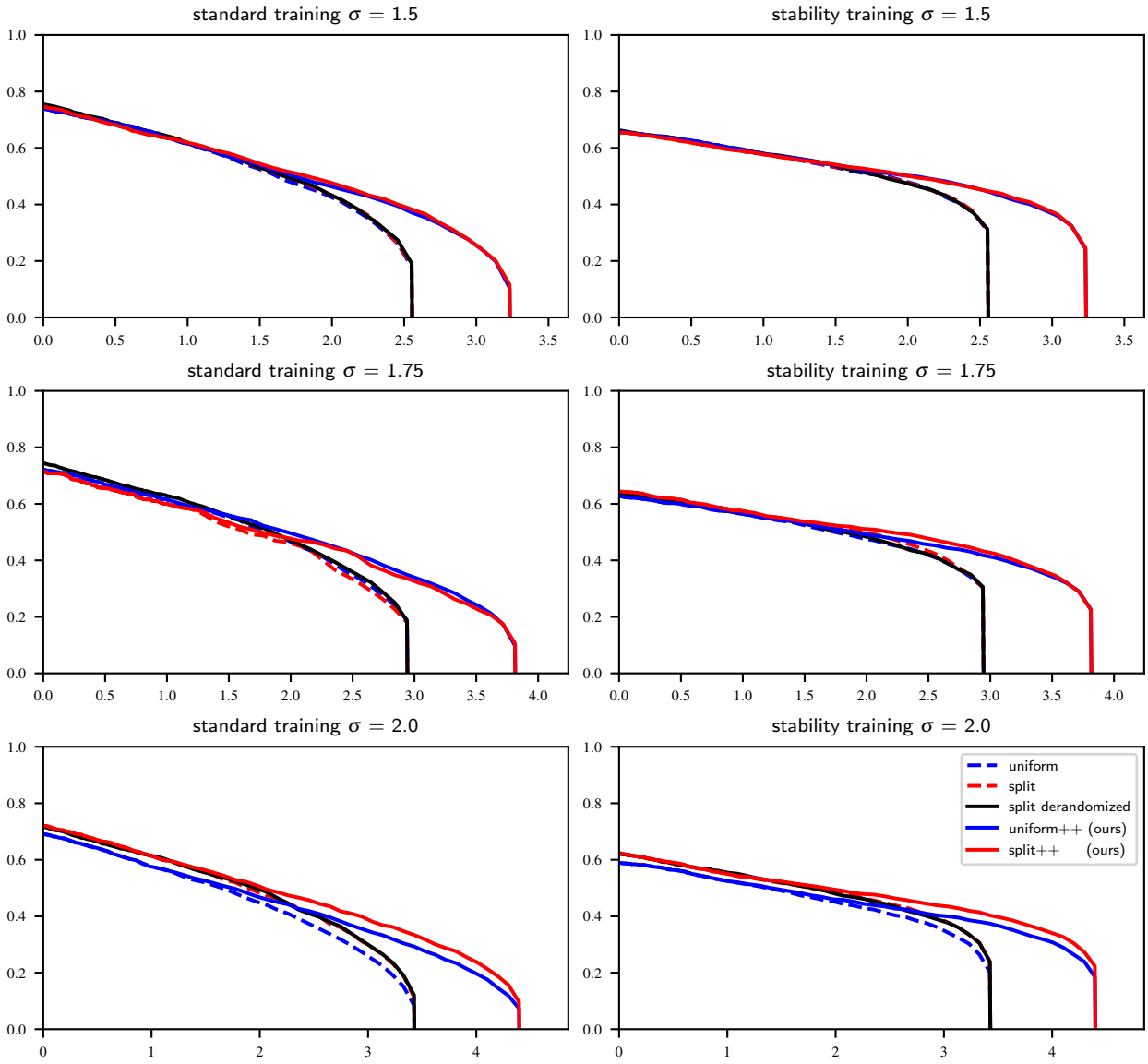

Figure 9: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns.

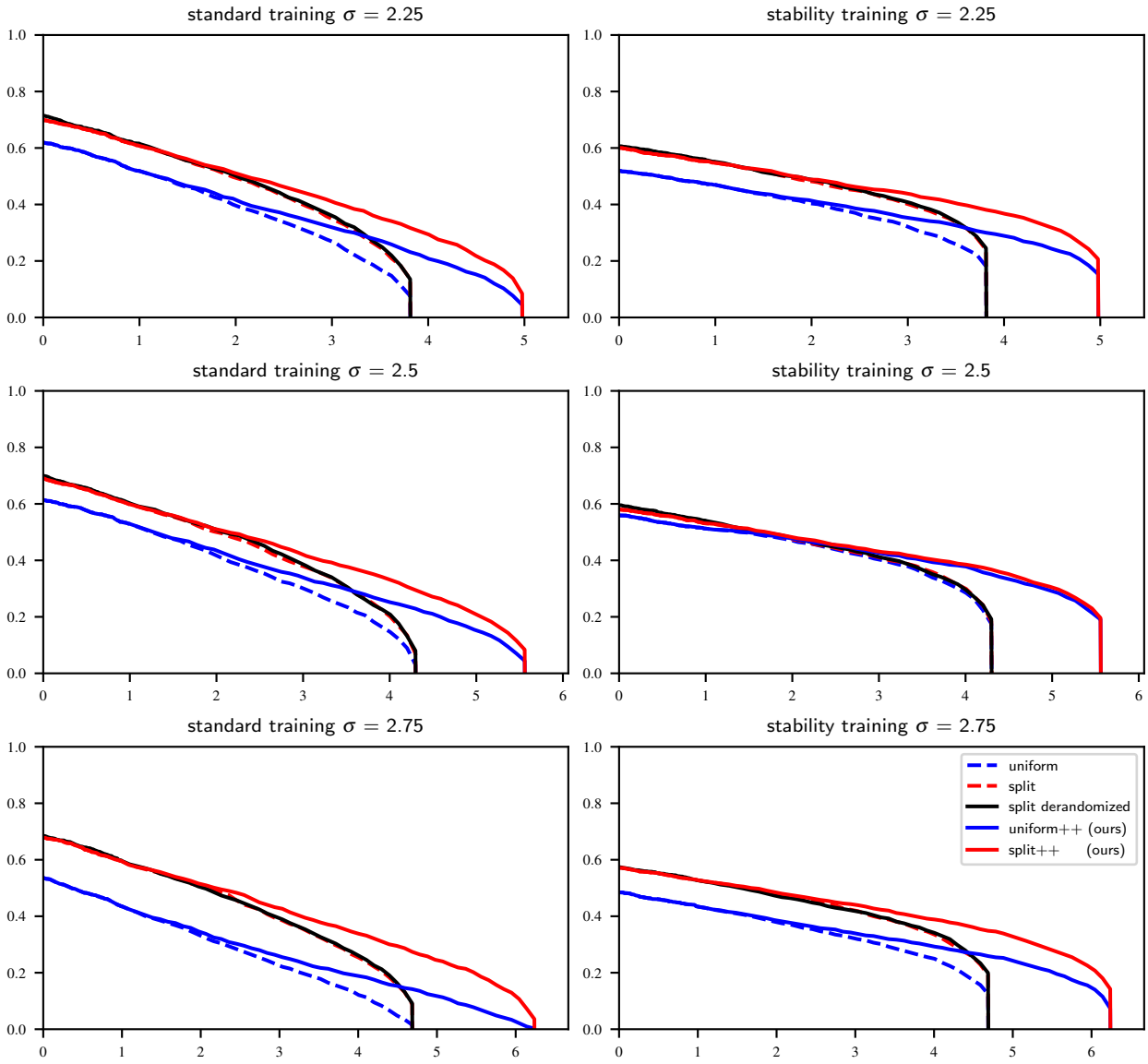

Figure 10: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns.

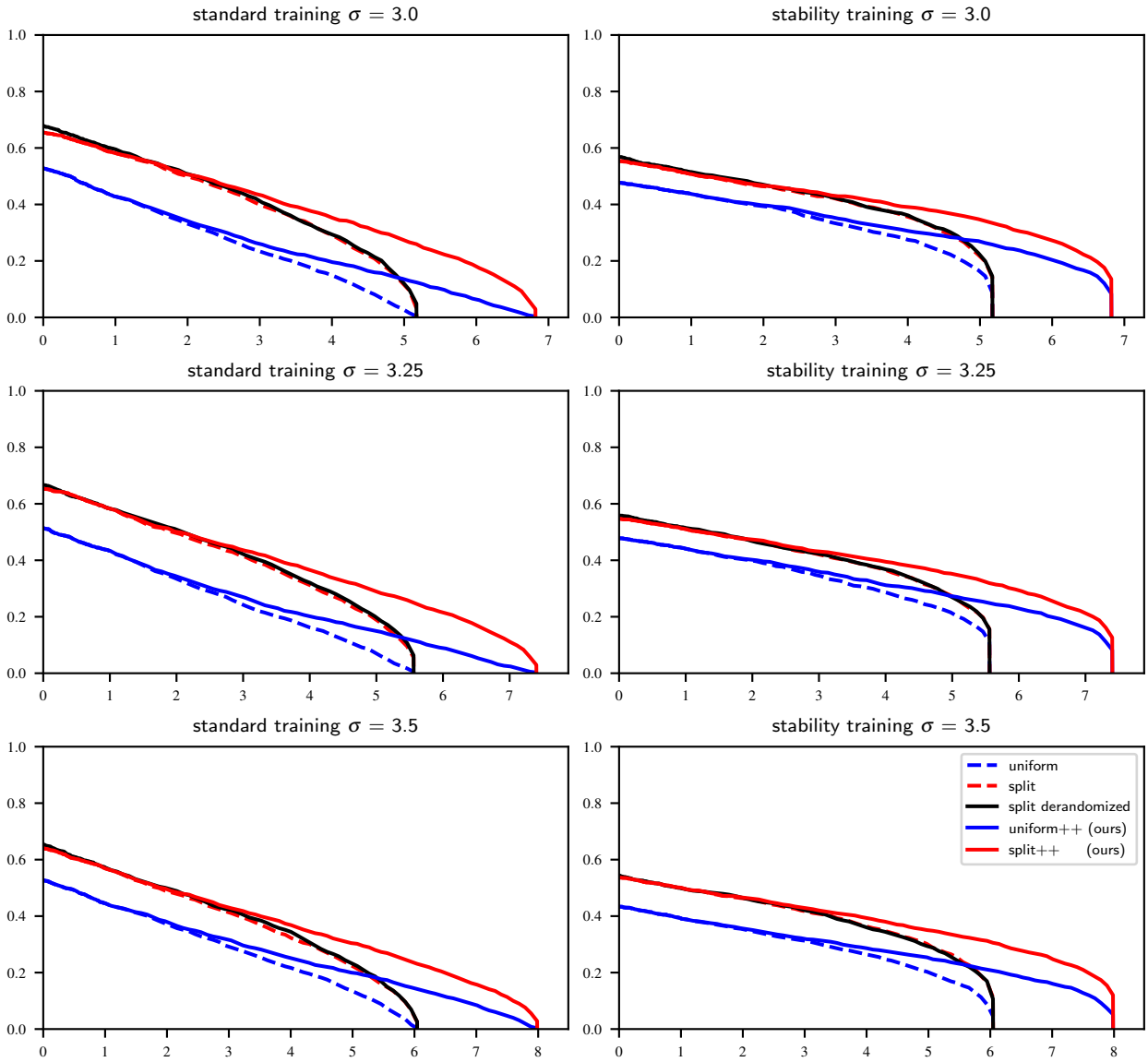

Figure 11: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns.

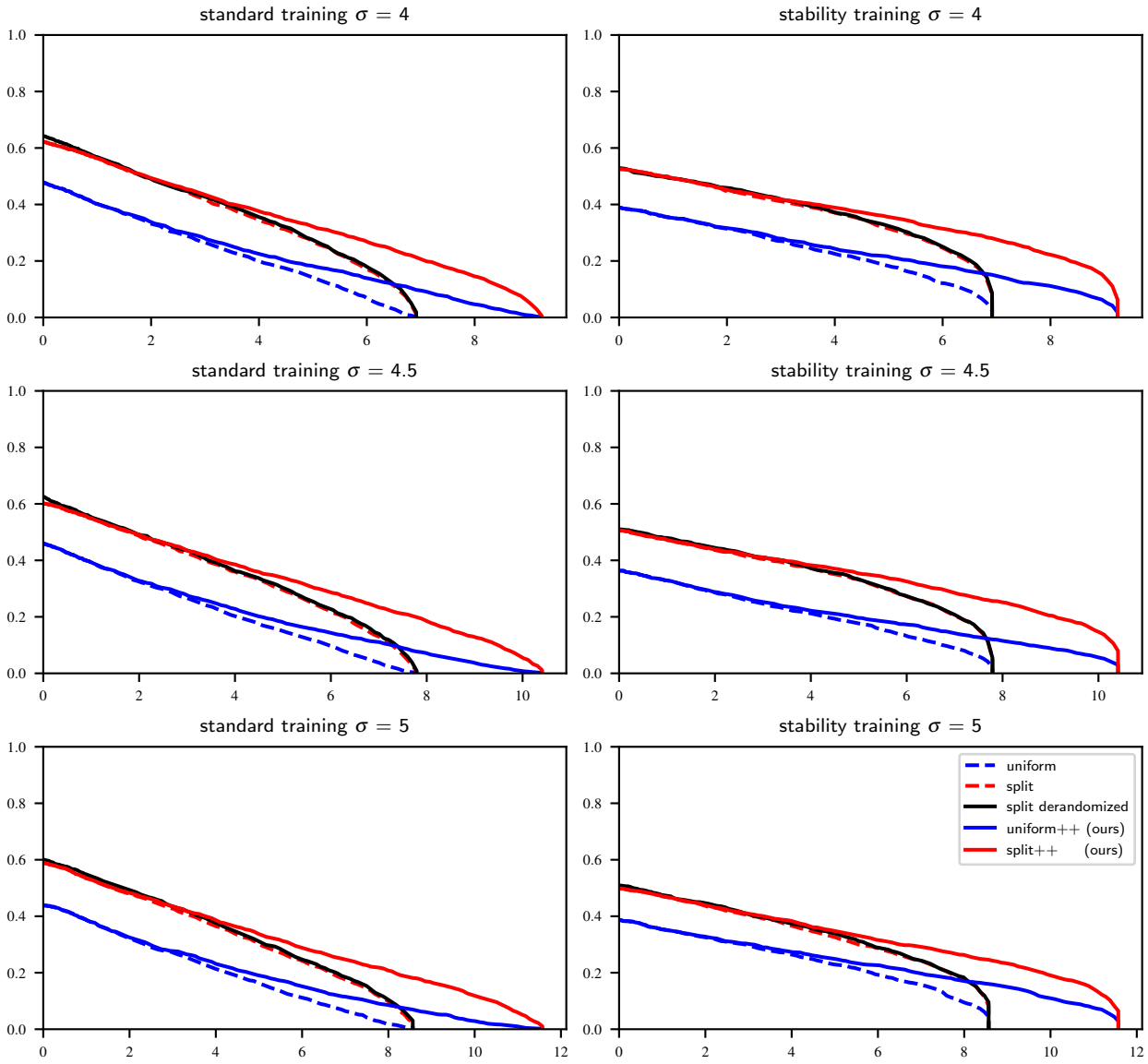

Figure 12: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns.

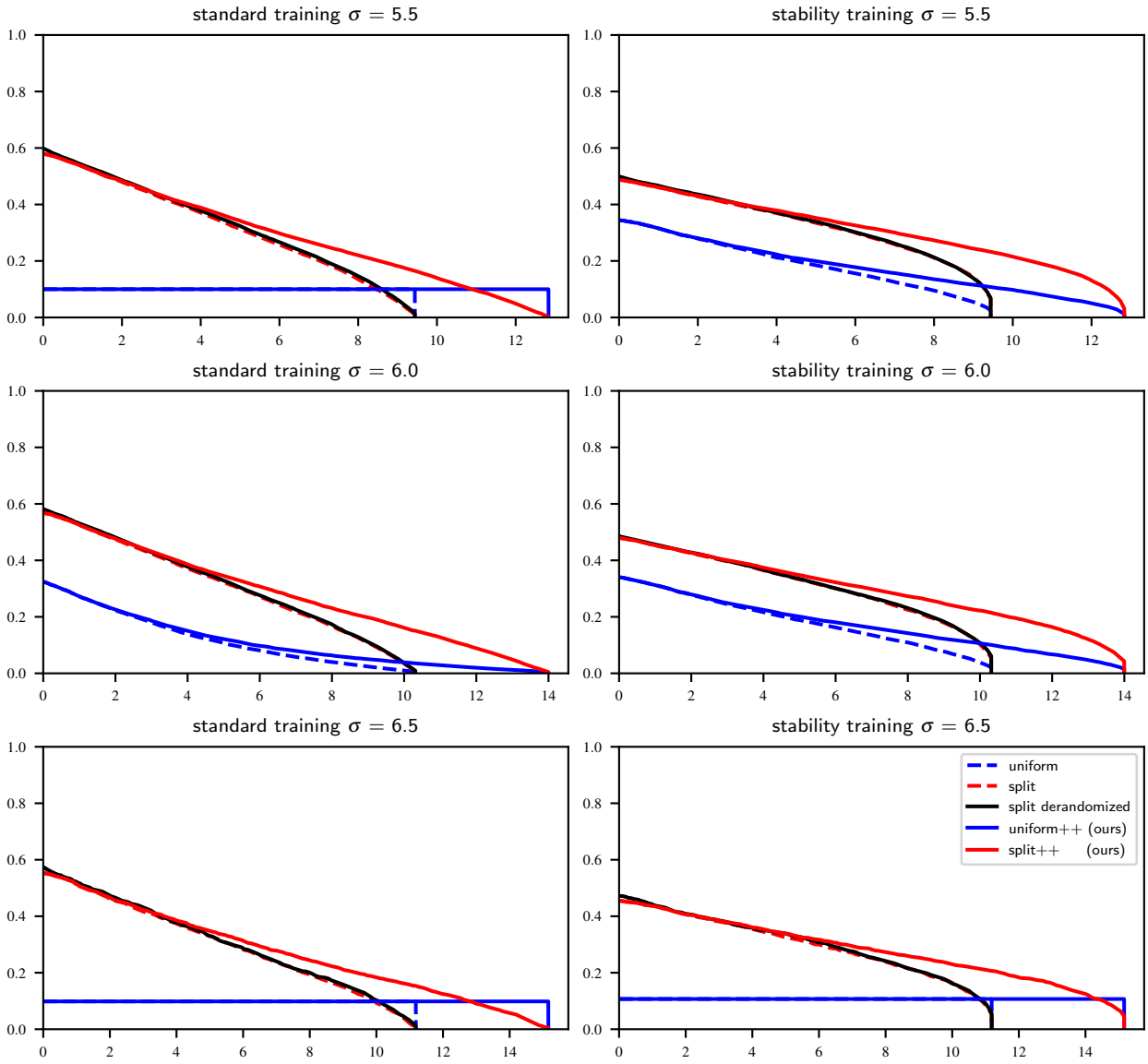

Figure 13: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns. Note that the uniform noise training sometimes converges to an (apparently) constant classifier

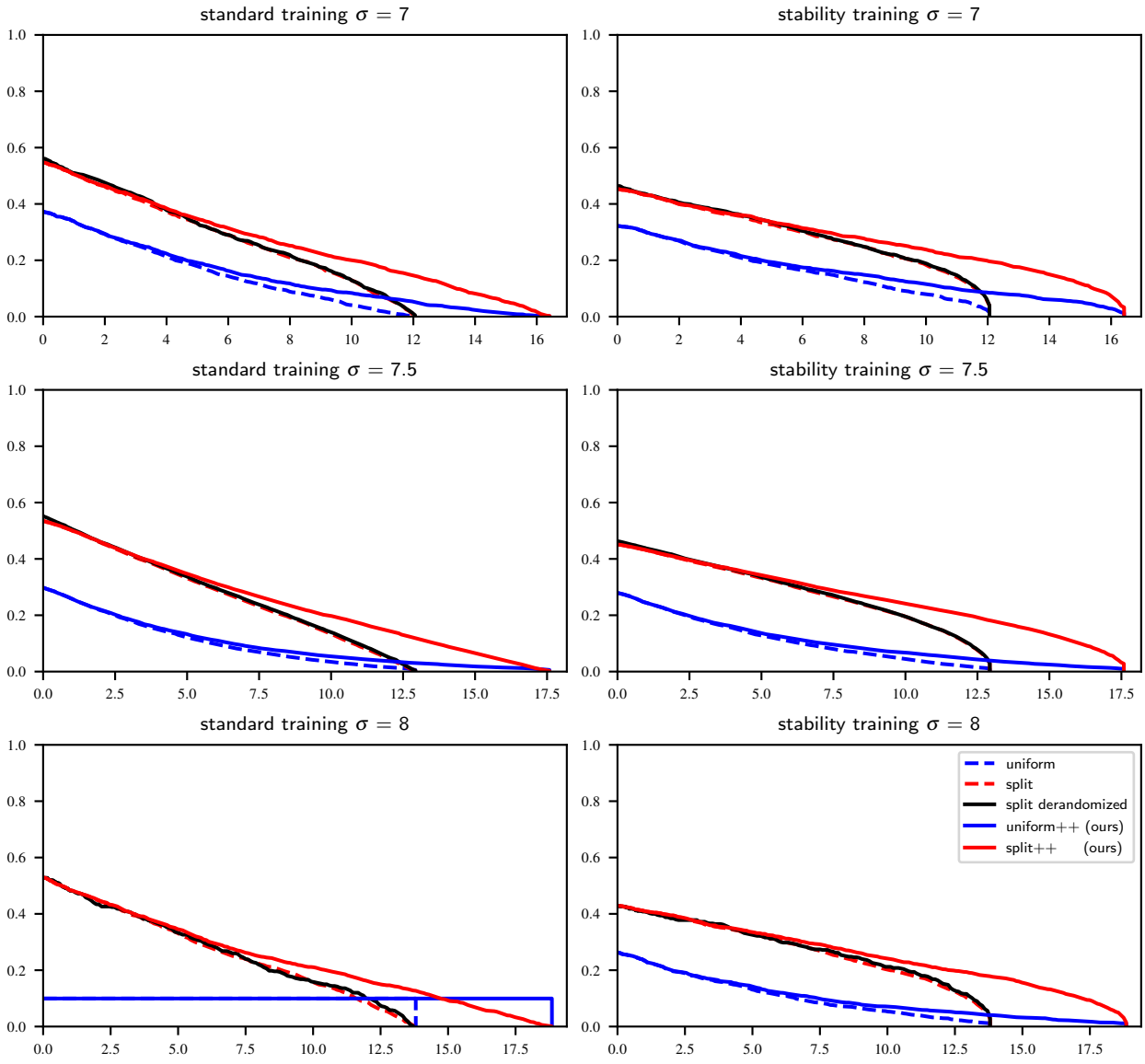

Figure 14: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns. Note that the uniform noise training sometimes converges to an (apparently) constant classifier

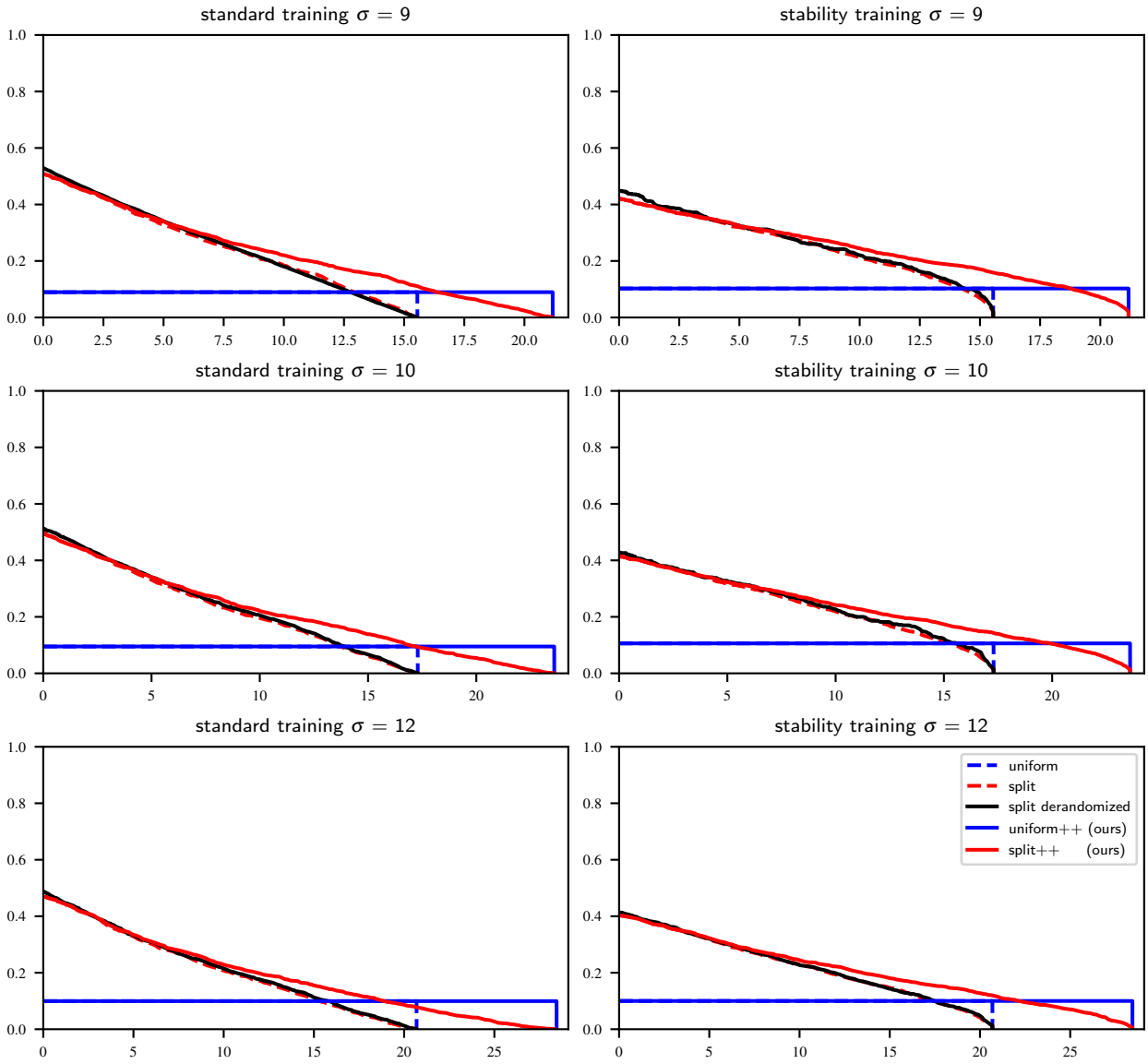

Figure 15: Robustness curves on CIFAR-10 for different methods. The noise magnitudes differ in rows and the training method differ in columns. Note that the uniform noise training converges to an (apparently) constant classifier

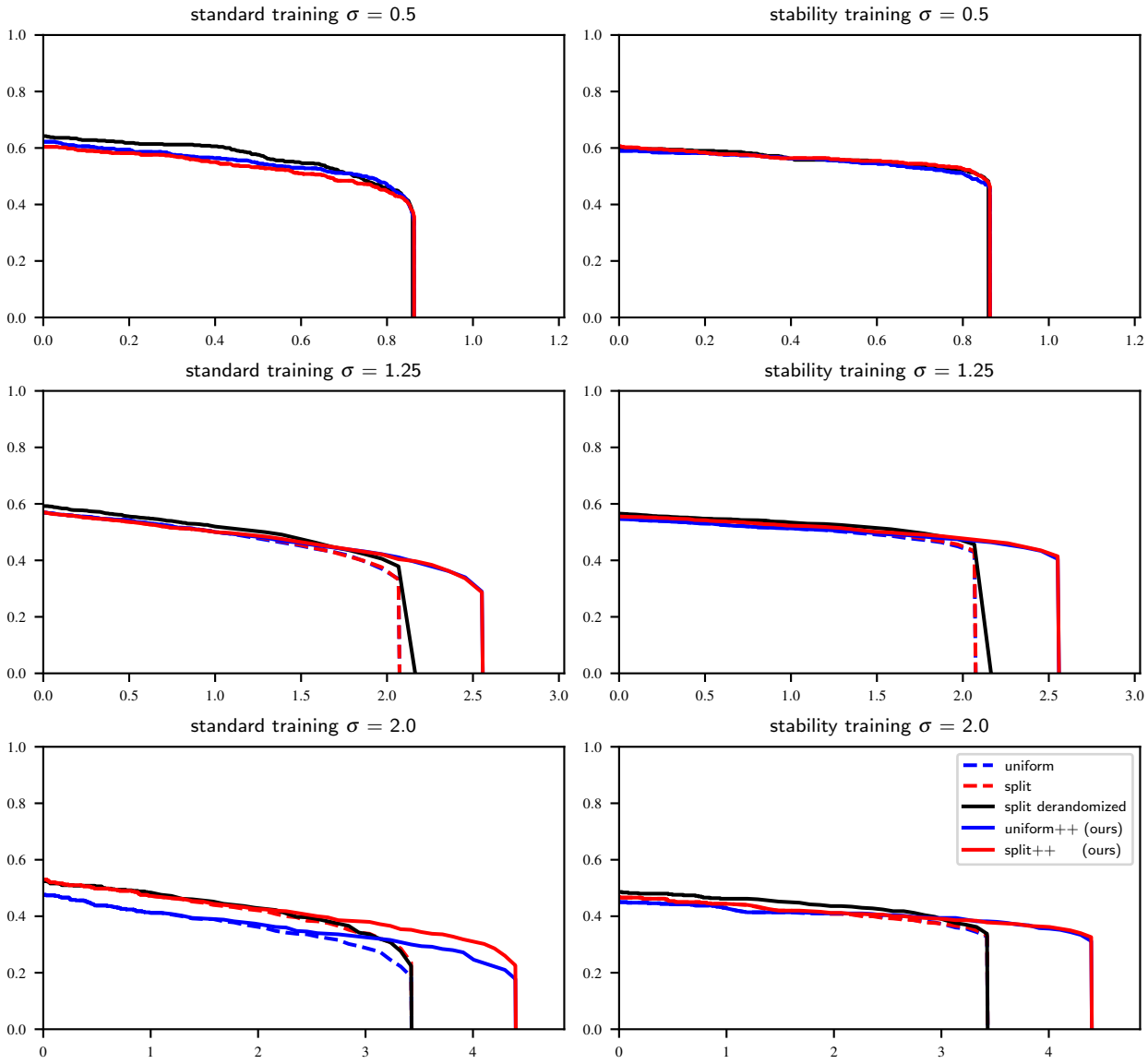

Figure 16: Robustness curves on ImageNet for different methods. The noise magnitudes differ in rows and the training method differ in columns.

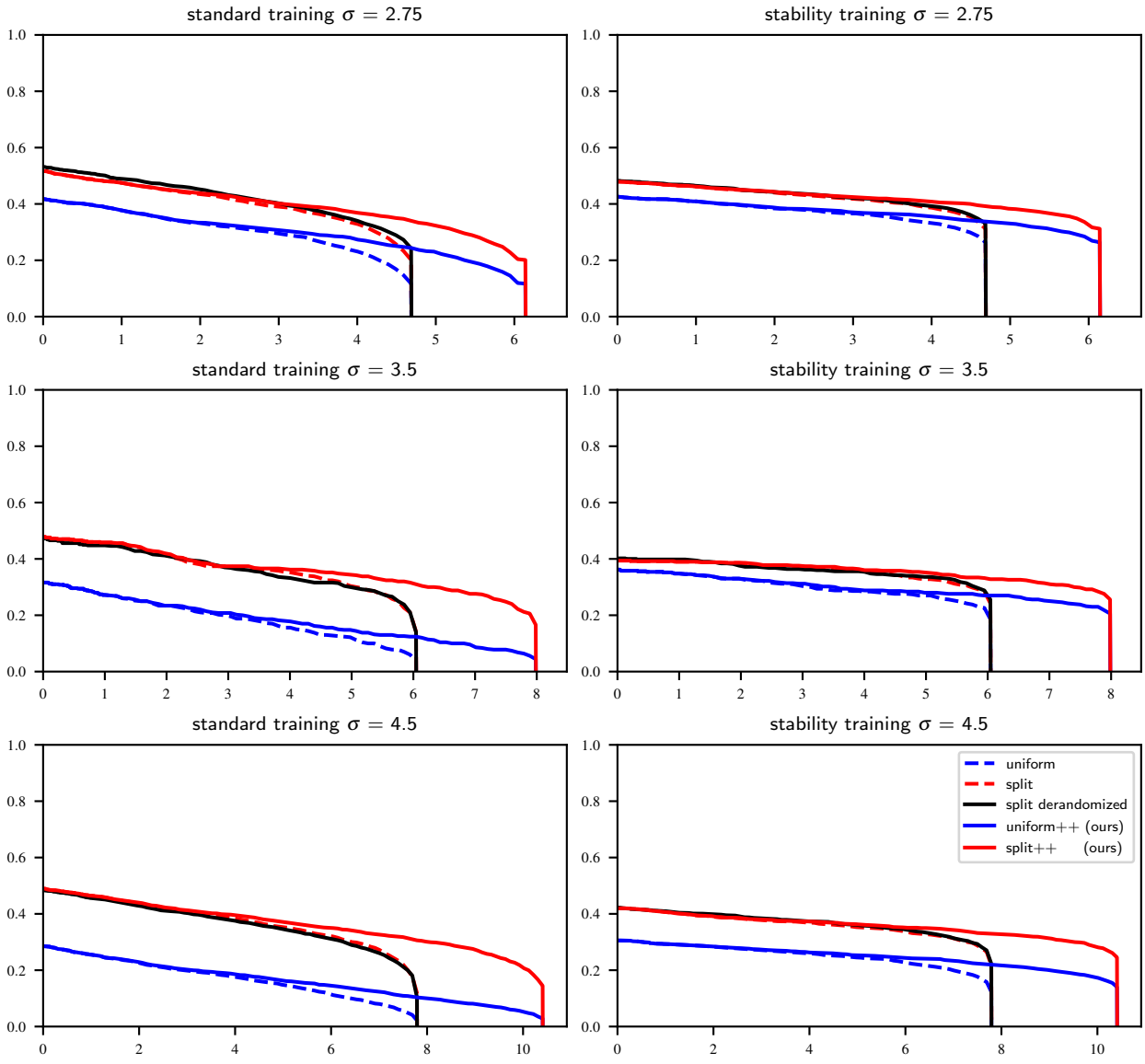

Figure 17: Robustness curves on ImageNet for different methods. The noise magnitudes differ in rows and the training method differ in columns.