SUPPLEMENTAL MATERIALS FOR: IMPROVING THE ACCURACY-ROBUSTNESS TRADE-OFF OF CLASSIFIERS VIA ADAPTIVE SMOOTHING

Yatong Bai[1], Brendon G. Anderson[1], Aerin Kim[2], Somayeh Sojoudi[1]
[1]*University of California, Berkeley* [2]*Scale AI*
{yatong_bai, bganderson, sojoudi}@berkeley.edu, aerinykim@gmail.com

A. Additional experiment results

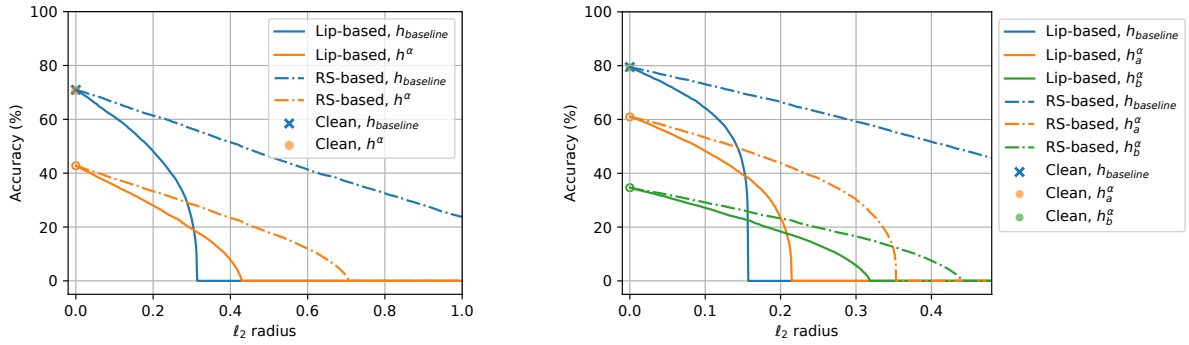
A.1. Visualization of the certified robust radii

Next, we visualize the certified robust radii presented in Theorem 1 and Theorem 2. In this subsection, the smoothing strength α is again a fixed value. Since a (Gaussian) RS model with smoothing covariance matrix $\sigma^2 I_d$ has an ℓ_2-Lipschitz constant $\sqrt{\frac{2}{\pi\sigma^2}}$, such a model can be used to simultaneously visualize both theorems, with Theorem 2 giving tighter certificates of robustness. Note that RS models with a larger smoothing variance certify larger radii but achieve lower clean accuracies, and vice versa. Here, we consider the CIFAR-10 dataset and select $g(\cdot)$ to be a ConvNeXT-T model with a clean accuracy of 97.25%, and use the RS models presented in [38] as $h(\cdot)$. For a fair comparison, we select an α value such that the clean accuracy of the constructed mixed classifier $h^\alpha(\cdot)$ matches that of another RS model $h_{\text{baseline}}(\cdot)$ with a smaller smoothing variance. The expectation term in the RS formulation is approximated with the empirical mean of 10000 random perturbations drawn from $\mathcal{N}(0, \sigma^2 I_d)$, and the certified radii of $h_{\text{baseline}}(\cdot)$ are calculated using Theorems 1 and 2 by setting α to 1. Fig. SM1 displays the calculated certified accuracies of $h^\alpha(\cdot)$ and $h_{\text{baseline}}(\cdot)$ at various attack radii. The ordinate "Accuracy" at a given abscissa "ℓ_2 radius" reflects the percentage of the test data for which the considered model gives a correct prediction as well as a certified radius at least as large as the ℓ_2 radius under consideration.

In both subplots of Fig. SM1, the certified robustness curves of $h^\alpha(\cdot)$ do not connect to the clean accuracy when α approaches zero. This is because Theorems 1 and 2 both consider robustness with respect to $h(\cdot)$ and do not issue certificates to test inputs at which $h(\cdot)$ makes incorrect predictions, even if $h^\alpha(\cdot)$ predicts correctly at some of these points. This is reasonable because we do not assume any robustness or Lipschitzness of $g(\cdot)$, and $g(\cdot)$ is allowed to be arbitrarily incorrect whenever the radius is non-zero.

The Lipschitz-based bound of Theorem 1 allows us to visualize the performance of the mixed classifier $h^\alpha(\cdot)$ when $h(\cdot)$ is an ℓ_2-Lipschitz model. In this case, the curves associated with $h^\alpha(\cdot)$ and $h_{\text{baseline}}(\cdot)$ intersect, with $h^\alpha(\cdot)$ achieving higher certified accuracy at larger radii and $h_{\text{baseline}}(\cdot)$ certifying more points at smaller radii. By adjusting α and the Lipschitz constant of $h(\cdot)$, it is possible to change the location of this intersection while maintaining the same level of clean accuracy. Therefore, the mixed classifier structure allows for optimizing the certified accuracy at a particular radius, while keeping the clean accuracy unchanged.

The RS-based bound from Theorem 2 captures the behavior of $h^\alpha(\cdot)$ when $h(\cdot)$ is an RS model. For both $h^\alpha(\cdot)$ and $h_{\text{baseline}}(\cdot)$, the RS-based bounds certify larger radii than the corresponding Lipschitz-based bounds. Nonetheless, $h_{\text{baseline}}(\cdot)$ can certify more points with the RS-based guarantee. Intuitively, this phenomenon suggests that RS models can yield correct but low-confidence predictions when under attack with a large radius, and thus may not be best-suited for our mixing operation, which relies on robustness with non-zero margins. In contrast, Lipschitz models, a more general and common class of models, exploit the mixing operation more effectively. Moreover, as shown in Fig. 4, empirically robust models often yield high-confidence predictions when under attack, making them more suitable to be used as the robust base classifier for $h^\alpha(\cdot)$.



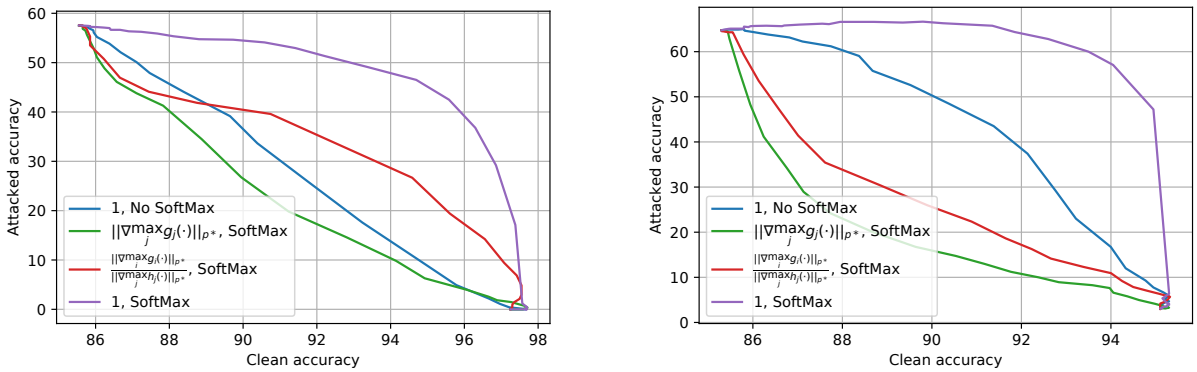
(a) $h_{\text{baseline}}(\cdot)$: RS with $\sigma = 0.5$.
$h^\alpha(\cdot)$ uses $\alpha = 0.76$;
$h(\cdot)$ is RS with $\sigma = 1$.

(b) $h_{\text{baseline}}(\cdot)$: RS with $\sigma = 0.25$.
Consider two mixed classifier examples:
$h_a^\alpha(\cdot)$ uses $\alpha = 0.76$ and $h_a(\cdot)$ is RS with $\sigma = 0.5$;
$h_b^\alpha(\cdot)$ uses $\alpha = 0.67$ and $h_b(\cdot)$ is RS with $\sigma = 1.0$.

Figure SM1. Comparing the certified accuracy-robustness trade-off of RS models and our mixed classifier using both Lipschitz-based (Lip-based) certificates and RS-based certificates (Theorems 1 and 2, respectively). The clean accuracies are the same between $h_{\text{baseline}}(\cdot)$ and $h^\alpha(\cdot)$ in each subfigure, and the empty circles represent discontinuity in the certified accuracy at radius 0.

A.2. Additional empirical supports for selecting $R_i(x) = 1$

In this section, we use additional empirical evidence (Figures SM2a and SM2b) to show that $R_i(x) = 1$ is the appropriate choice for the adaptive smoothing formulation, and that the post-SoftMax probabilities should be used for smoothing. While most of the experiments in this paper are based on ResNets, the architecture is chosen solely because of its popularity, and our method does not depend on any properties of ResNets. Therefore, for the experiment in Figure SM2a, we select an alternative architecture by using a more modern ConvNeXT-T model [24] pre-trained on ImageNet-1k as $g(\cdot)$. We also use a robust model trained via TRADES in place of an adversarially-trained network for $h(\cdot)$. Moreover, while most of our experiments are based on ℓ_∞ attacks, our method applies to all ℓ_p attack budgets. In Figure SM2b, we provide an example that considers the ℓ_2 attack. The experiment settings are summarized in Table SM1.



(a) ConvNeXT-T and TRADES WideResNet-34 under ℓ_∞ PGD attack.

(b) Standard and AT ResNet-18s under ℓ_2 PGD attack.

Figure SM2. Comparing the "attacked accuracy versus clean accuracy" curve of various options for $R_i(x)$ with alternative selections of base classifiers.

Table SM1. Experiment settings for comparing the choices of $R_i(x)$.

	PGD attack settings	$g(\cdot)$ Architecture	$h(\cdot)$ Architecture
Figure 1	ℓ_∞, $\epsilon = 8/255$, 10 Steps	Standard ResNet-18	ℓ_∞-adversarially-trained ResNet-18
Figure SM2a	ℓ_∞, $\epsilon = 8/255$, 20 Steps	Standard ConvNeXT-T	TRADES WideResNet-34
Figure SM2b	ℓ_2, $\epsilon = 0.5$, 20 Steps	Standard ResNet-18	ℓ_2-adversarially-trained ResNet-18

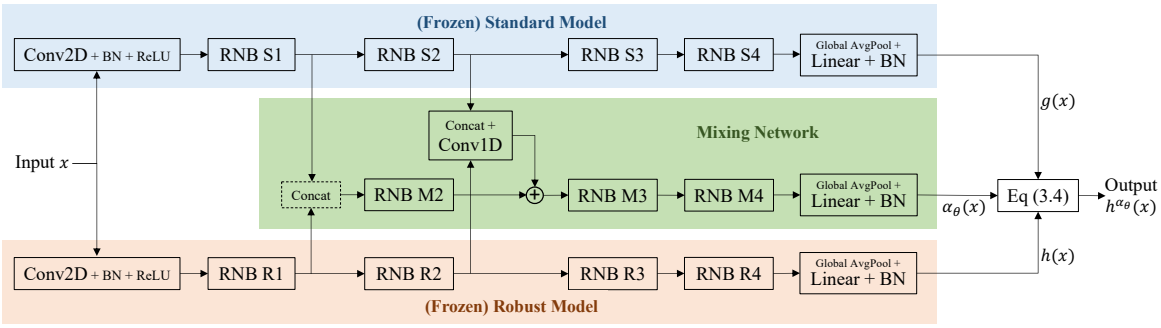


Figure SM3. The architecture of the mixed classifier introduced in Section 4 when applied to a pair of ResNet base models.

Figures SM2a and SM2b demonstrate that setting $R_i(x)$ to the constant 1 achieves the best trade-off curve between clean and attacked accuracy. Moreover, smoothing using the post-SoftMax probabilities outperforms the pre-SoftMax logits. This result aligns with the conclusions of Figure 1 and our theoretical analyses, demonstrating that various robust networks share the property of being more confident when classifying correctly than when making mistakes.

B. Implementation of the mixing network in experiments

Since our formulation does not depend on the architecture of the base classifiers, Figure 3 presents the design of the mixing network in the context of general standard and robust classifiers. In the experiments presented in Sec. 5.2, Both $g(\cdot)$ and $h(\cdot)$ are based on variants of the ResNet family, which share the general structure of having four main blocks. Thus, we present the structure of the mixed classifier with ResNet-like base models in Figure SM3. Following [26], we consider the initial Conv2D layer and the first ResNet block as the upstream layers. The embeddings extracted by the first Conv2D layers in $g(\cdot)$ and $h(\cdot)$ are concatenated before being provided to the mixing network $\alpha_\theta(\cdot)$. We further select the second ResNet block as the middle layers. For this layer, in addition to concatenating the embeddings from $g(\cdot)$ and $h(\cdot)$, we also attach a linear transformation layer (Conv1x1) to match the dimensions, reduce the number of features, and improve efficiency.

As mentioned in Sec. 4.1, the range of $\alpha_\theta(\cdot)$ can be constrained to be within $(\alpha_{\min}, \alpha_{\max}) \subseteq [0, 1]$ if certified robustness is desired. We empirically observe that setting $\alpha_{\max} - \alpha_{\min}$ to be 0.1 or 0.15 works well. This observation coincides with Fig. 4, which shows that a slight increase in α can greatly enhance the robustness at the most sensitive region. The value of $\alpha_{\min}$ can then be determined by enforcing a desired level of either clean validation accuracy or robustness. Following this guideline, for the two models demonstrated in Table 4, we set the ranges of $\alpha_\theta(\cdot)$ to be $(0.84, 0.99)$ and $(0.815, 0.915)$, respectively. Note that this range is only applied during validation. When training $\alpha_\theta(\cdot)$, we use the full $(0, 1)$ range for its output, so that the training-time adversary can generate strong and diverse attacks that fully exploit $\alpha_\theta(\cdot)$, which is crucial for securing the robustness of the mixing network.

B.1. Training loss for the mixing network

Consider the following two loss functions for training the mixing network $\alpha_\theta(\cdot)$:

- **Multi-class cross-entropy:** We minimize the multi-class cross-entropy loss of the combined classifier, which is the ultimate goal of the mixing network:

$$\min_\theta \mathbb{E}_{\substack{(x,y)\sim\mathcal{D} \\ \delta\sim\mathcal{F}}} \Big[\ell_{\mathrm{CE}}\big(h^{\alpha_\theta}(x+\delta), y\big) \Big], \tag{B.1}$$

  where ℓ_{CE} is the cross-entropy (CE) loss for logits and $y \in [c]$ is the label corresponding to x. The base classifiers $g(\cdot)$ and $h(\cdot)$ are frozen and not updated. Again, δ denotes the perturbation and the distribution $\mathcal{F}$ is arbitrary. In our experiments, to avoid overfitting to a particular attack radius, $\mathcal{F}$ is selected to be formed by perturbations with randomized radii.

- **Binary cross-entropy:** The optimal $\alpha^\star$ (parameterized by an optimal $\theta^\star$) that minimizes ℓ_{CE} in (B.1) can be estimated for each training point. Specifically, depending on whether the input is attacked and how it is

Table SM2. The PGD_{20} accuracy on CIFAR-10 with various loss hyperparameter settings. The setting is the same as in Table 2, and we consider both attack and defense in Setting B.

	$c_{\text{CE}} = 0$ $c_{\text{BCE}} = 1.5$	$c_{\text{CE}} = 0.5$ $c_{\text{BCE}} = 1$	$c_{\text{CE}} = 1$ $c_{\text{BCE}} = 0.5$	$c_{\text{CE}} = 1.5$ $c_{\text{BCE}} = 0$
$c_{\text{prod}} = 0$	54.5 %	52.8 %	53.8 %	54.4 %
$c_{\text{prod}} = 0.1$	54.3 %	54.1 %	54.0 %	54.1 %
$c_{\text{prod}} = 0.2$	55.1 %	54.2 %	54.3 %	53.9 %

attacked, either $g(\cdot)$ or $h(\cdot)$ should be prioritized. Thus, we treat the task as a binary classification problem and solve the optimization problem

$$\min_{\theta} \mathbb{E}_{\substack{(x,y)\sim\mathcal{D} \\ \delta\sim\mathcal{F}}} \Big[\ell_{\text{BCE}}\big(\alpha_\theta(x+\delta), \widetilde{\alpha}\big) \Big],$$

where ℓ_{BCE} is the binary cross-entropy (BCE) loss for probabilities and $\widetilde{\alpha} \in \{0, 1\}$ is the "pseudo label" for the output of the mixing network.

Using only the multi-class loss suffers from a distributional mismatch between the training set and the test set. The robust classifier $h(\cdot)$ may achieve a low loss on adversarial training data but a high loss on adversarial test data. For example, with the CIFAR-10 dataset and our ResNet-18 robust classifier, the PGD_{10} adversarial training accuracy is 93.01% while the PGD_{10} test accuracy is 45.55%. As a result, approximating (B.1) with empirical risk minimization on the training set does not effectively optimize the true risk. When the adversary attacks a test input x targeting $h(\cdot)$, the standard prediction $g(x)$ yields a lower loss than $h(x)$. However, if x is an attacked example in the training set, then the losses of $g(x)$ and $h(x)$ are similar, and the mixing network does not receive a strong incentive to choose $g(\cdot)$ when it detects an attack targeting $h(\cdot)$.

The binary loss, however, does not capture the potentially different sensitivity of each input. Certain inputs can be more vulnerable to adversarial attacks, and ensuring the correctness of the mixing network on these inputs is more crucial.

To this end, we propose a composite loss function that combines the above two components, providing incentives for the mixing network to select the standard classifier $g(\cdot)$ when appropriate, while forcing the mixing network to remain conservative. The composite loss for a data-label pair (x, y) is given by

$$\begin{aligned} \ell_{\text{composite}}\big(\theta, (x, y, \widetilde{\alpha})\big) = c_{\text{CE}} \cdot \ell_{\text{CE}}\big(h^{\alpha_\theta}(x+\delta), y\big) + c_{\text{BCE}} \cdot \ell_{\text{BCE}}\big(\alpha_\theta(x+\delta), \widetilde{\alpha}\big) \\ + c_{\text{prod}} \cdot \ell_{\text{CE}}\big(h^{\alpha_\theta}(x+\delta), y\big) \cdot \ell_{\text{BCE}}\big(\alpha_\theta(x+\delta), \widetilde{\alpha}\big), \end{aligned} \tag{B.2}$$

where the hyperparameters $c_{\text{CE}}, c_{\text{BCE}}$, and c_{prod} control the weights of the loss components. The effects of the constants are discussed below.

Since multiplying the three weight constants by the same number is equivalent to using a larger optimizer step size and is not the purpose of this ablation study, we fix $c_{\text{CE}} + c_{\text{BCE}} = 1.5$. To avoid the issue of becoming excessively conservative and always prioritizing the robust base model (as described in Appendix B.1), we add a batch normalization layer without trainable affine transform to the output of the mixing network. Additionally, note that since the mixing network has a single output, one can arbitrarily shift this output to achieve the desired balance between clean and attacked accuracies. For a fair and illustrative comparison, after training a mixing network $\alpha_\theta(\cdot)$ with each hyperparameter setting, we add an appropriate constant to the output of the $\alpha_\theta(\cdot)$ so that the clean accuracy of the overall model $h^{\alpha_\theta}(\cdot)$ is $90 \pm 0.02\%$, and compare the PGD_{20} attacked accuracy of $h^{\alpha_\theta}(\cdot)$ in Table SM2. As a baseline, when the smoothing strength α is a constant, the PGD_{20} accuracy is 52.6% when the clean accuracy is tuned to be 90% (the corresponding α value is 1.763). The above results demonstrate that $c_{\text{CE}} = 0$, $c_{\text{BCE}} = 1.5$, and $c_{\text{prod}} = 0.2$ works the best.

Our results also show that a small positive c_{prod} is generally beneficial. This makes sense because the CE loss is low for a particular input if both $g(\cdot)$ and $h(\cdot)$ correctly predict its class. Thus, the smoothing strength should not matter for such input, and therefore the BCE loss is weighted by a small number. Compared with using only the BCE loss, the product term of the CE and the BCE components is lenient on inputs correctly classified by the mixed model $h^{\alpha_\theta}(\cdot)$, while assigning more weight to the data that are incorrectly predicted.

Table SM3. Compare various selections of the mixing network's Sigmoid activation scaling factor.

Scale $= 0.5$	Scale $= 1$	Scale $= 2$	Scale $= 4$
55.1 %	55.5 %	55.7 %	55.6 %

Recall that the output range of $\alpha_\theta(\cdot)$ is $[0, 1]$, which is enforced by appending a Sigmoid output activation function. In addition to shifting, one can arbitrarily scale the Sigmoid activation's input. By performing this scaling, we effectively calibrate the confidence of the mixing network. In Table SM2, this scaling is set to the same constant for all settings. In Table SM3, we select the best loss parameter and analyze the validation-time Sigmoid scaling. Again, we shift the Sigmoid input so that the clean accuracy is $90 \pm 0.02\%$. While a larger scale benefits the performance on clean/attacked examples that are confidently recognized by the mixing network, an excessively large scale makes $h^{\alpha_\theta}(\cdot)$ less stable under attack. Table SM3 shows that applying a scaling factor of 2 yields the best result for the given experiment setting.

C. Additional Related Works

C.1. Adversarial attacks and defenses

The fast gradient sign method (FGSM) and PGD attacks based on the first-order maximization of the cross-entropy loss have traditionally been considered classic and straightforward attacks [15, 25]. However, these attacks have been shown to be insufficient as defenses designed against them are often easily circumvented [5, 8]. To this end, various attack methods based on alternative loss functions, Expectation Over Transformation, and black-box perturbations have been proposed. Such efforts include MultiTargeted attack loss [18], AutoAttack [14], adaptive attack [34, 35], minimal distortion attack [13], and many others, even considering attacking test-time defenses [12]. The diversity of attack methods has led to the creation of benchmarks such as RobustBench [11] and ARES-Bench [22] to unify the evaluation of robust models.

On the defense side, while adversarial training [25] and TRADES [38] have seen enormous success, such methods are often limited by a significantly larger amount of required training data [31]. Initiatives that construct more effective training data via data augmentation [16, 17, 29] and generative models [32, 37] have successfully produced more accurate and robust models. Improved versions of adversarial training [6, 19, 27, 33, 36] have also been proposed.

Moreover, ensemble-based defenses, such as random ensemble [23], diverse ensemble [1, 28], and Jacobian ensemble [10], have been proposed. In comparison, this work is distinct in that our mixing scheme uses two separate classifiers, incorporating one non-robust component while still ensuring the adversarial robustness of the overall design. By doing so, we take advantage of the high performance of modern pre-trained models, significantly alleviating the accuracy-robustness trade-off and achieving much higher overall performances. Additionally, unlike some previous ensemble initiatives, our formulation is deterministic and straightforward (in the sense of gradient propagation), making it easier to evaluate its robustness properly. The work [20] also explored assembling an accurate classifier and a robust classifier, but the method considered robustness against distribution shift in a non-adversarial setting and was based on different intuitions.

C.2. Adversarial input detectors

It has been shown that adversarial inputs can be detected via various methods. For example, [26] proposes to append an additional detection branch to an existing neural network, and uses adaptive adversarial data to train the detector in a supervised fashion. However, [7] has shown that it is possible to bypass this detection method. They constructed adversarial examples via the C&W attacks [8] and simultaneously targeted the classification branch and the detection branch by treating the two branches as an "augmented classifier". According to [7], the detector is effective against the types of attack that it is trained with, but not necessarily the attack types that are absent in the training data. It is thus reasonable to expect the detector to be able to detect a wide range of attacks if it is trained using sufficiently diverse types of attacks (including those targeting the detector itself). While exhaustively covering the entire adversarial input space is intractable, and it is unclear to what degree one needs to diversify the attack types in practice, our experiments show that our modified architecture based on [26] can recognize the state-of-the-art AutoAttack adversaries with a high success rate.

The literature has also considered alternative detection methods that mitigate the above challenges faced by detectors trained in a supervised fashion [9]. Such initiatives include unsupervised detectors [3, 4] and re-attacking [2]. Since universally effective detectors have not yet been discovered, this paper focuses on transferring the properties of the existing detector toward better overall robustness. Future advancements in the field of adversary detection can further enhance the performance of our method.

D. Proofs

D.1. Proof of Lemma 1

Lemma 1 (restated). *Let $x \in \mathbb{R}^d$ and $r \geq 0$. If it holds that $\alpha \in [\frac{1}{2}, 1]$ and $h(\cdot)$ is certifiably robust at x with margin $\frac{1-\alpha}{\alpha}$ and radius r, then the mixed classifier $h^\alpha(\cdot)$ is robust in the sense that $\arg\max_i h_i^\alpha(x+\delta) = \arg\max_i h_i(x)$ for all $\delta \in \mathbb{R}^d$ such that $\|\delta\|_p \leq r$.*

Proof. Suppose that $h(\cdot)$ is certifiably robust at x with margin $\frac{1-\alpha}{\alpha}$ and radius r. Since $\alpha \in [\frac{1}{2}, 1]$, it holds that $\frac{1-\alpha}{\alpha} \in [0, 1]$. Let $y = \arg\max_i h_i(x)$. Consider an arbitrary $i \in [c] \setminus \{y\}$ and $\delta \in \mathbb{R}^d$ such that $\|\delta\|_p \leq r$. Since $g_i(x+\delta) \in [0, 1]$, it holds that

$$\begin{aligned}\exp\left(h_y^\alpha(x+\delta)\right) - \exp\left(h_i^\alpha(x+\delta)\right) =& (1-\alpha)(g_y(x+\delta) - g_i(x+\delta)) + \alpha(h_y(x+\delta) - h_i(x+\delta)) \\ \geq& (1-\alpha)(0-1) + \alpha(h_y(x+\delta) - h_i(x+\delta)) \\ \geq& (\alpha - 1) + \alpha\left(\tfrac{1-\alpha}{\alpha}\right) = 0.\end{aligned}$$

Thus, it holds that $h_y^\alpha(x+\delta) \geq h_i^\alpha(x+\delta)$ for all $i \neq y$, and thus $\arg\max_i h_i^\alpha(x+\delta) = y = \arg\max_i h_i(x)$. □

D.2. Proof of Theorem 1

Theorem 1 (restated). *Suppose that Assumption 1 holds, and let $x \in \mathbb{R}^d$ be arbitrary. Let $y = \arg\max_i h_i(x)$. Then, if $\alpha \in [\frac{1}{2}, 1]$, it holds that $\arg\max_i h_i^\alpha(x+\delta) = y$ for all $\delta \in \mathbb{R}^d$ such that*

$$\left\|\delta\right\|_p \leq r_{\mathrm{Lip},p}^\alpha(x) \coloneqq \min_{i \neq y} \frac{\alpha\left(h_y(x) - h_i(x)\right) + \alpha - 1}{\alpha\left(\mathrm{Lip}_p(h_y) + \mathrm{Lip}_p(h_i)\right)}.$$

Proof. Suppose that $\alpha \in [\frac{1}{2}, 1]$, and let $\delta \in \mathbb{R}^d$ be such that $\|\delta\|_p \leq r_{\mathrm{Lip},p}^\alpha(x)$. Also let $i \in [c] \setminus \{y\}$. It holds that

$$\begin{aligned}h_y(x+\delta) - h_i(x+\delta) &= h_y(x) - h_i(x) + h_y(x+\delta) - h_y(x) + h_i(x) - h_i(x+\delta) \\ &\geq h_y(x) - h_i(x) - \mathrm{Lip}_p(h_y)\|\delta\|_p - \mathrm{Lip}_p(h_i)\|\delta\|_p \\ &\geq h_y(x) - h_i(x) - \left(\mathrm{Lip}_p(h_y) + \mathrm{Lip}_p(h_i)\right) r_{\mathrm{Lip},p}^\alpha(x) \geq \tfrac{1-\alpha}{\alpha}.\end{aligned}$$

Thus, $h(\cdot)$ is certifiably robust at x with margin $\frac{1-\alpha}{\alpha}$ and radius $r_{\mathrm{Lip},p}^\alpha(x)$. Hence, by Lemma 1, the claim holds. □

D.3. Proof of Theorem 2

Theorem 2 (restated). *Suppose that Assumption 2 holds, and let $x \in \mathbb{R}^d$ be arbitrary. Let $y = \arg\max_i h_i(x)$ and $y' = \arg\max_{i \neq y} h_i(x)$. Then, if $\alpha \in [\frac{1}{2}, 1]$, it holds that $\arg\max_i h_i^\alpha(x+\delta) = y$ for all $\delta \in \mathbb{R}^d$ such that*

$$\|\delta\|_2 \leq r_\sigma^\alpha(x) \coloneqq \frac{\sigma}{2}\Big(\Phi^{-1}\big(\alpha h_y(x)\big) - \Phi^{-1}\big(\alpha h_{y'}(x) + 1 - \alpha\big)\Big).$$

Proof. First, note that since every $\overline{h}_i(\cdot)$ is not 0 almost everywhere or 1 almost everywhere, it holds that $h_i(x) \in (0, 1)$ for all i and all x. Now, suppose that $\alpha \in [\frac{1}{2}, 1]$, and let $\delta \in \mathbb{R}^d$ be such that $\|\delta\|_2 \leq r_\sigma^\alpha(x)$. Let $\mu_\alpha \coloneqq \frac{1-\alpha}{\alpha}$ (conversely, $\alpha = \frac{1}{\mu_\alpha + 1}$). We construct a scaled classifier $\tilde{h} \colon \mathbb{R}^d \to \mathbb{R}^c$, whose i^{th} entry is defined as

$$\tilde{h}_i(x) = \begin{cases} \frac{\overline{h}_y(x)}{1+\mu_\alpha} = \alpha\overline{h}_y(x) & \text{if } i = y, \\ \frac{\overline{h}_i(x)+\mu_\alpha}{1+\mu_\alpha} = \alpha\overline{h}_i(x) + 1 - \alpha & \text{if } i \neq y. \end{cases}$$

Furthermore, define a scaled RS classifier $\hat{h}\colon \mathbb{R}^d \to \mathbb{R}^c$ by

$$\hat{h}(x) = \mathbb{E}_{\xi\sim\mathcal{N}(0,\sigma^2 I_d)}\left[\tilde{h}(x+\xi)\right].$$

Then, since it holds that

$$\tilde{h}_y(x) = \frac{\overline{h}_y(x)}{1+\mu_\alpha} \in \left(0, \frac{1}{1+\mu_\alpha}\right) \subseteq (0,1),$$
$$\tilde{h}_i(x) = \frac{\overline{h}_i(x)+\mu_\alpha}{1+\mu_\alpha} \in \left(\frac{\mu_\alpha}{1+\mu_\alpha}, 1\right) \subseteq (0,1), \quad \forall i \neq y,$$

it must be the case that $0 < \tilde{h}_i(x) < 1$ for all i and all x, and hence, for all i, the function $x \mapsto \Phi^{-1}\big(\hat{h}_i(x)\big)$ is ℓ_2-Lipschitz continuous with Lipschitz constant $\frac{1}{\sigma}$ (see [21, Lemma 1], or Lemma 2 in [30] and the discussion thereafter). Therefore,

$$\left|\Phi^{-1}\big(\hat{h}_i(x+\delta)\big) - \Phi^{-1}\big(\hat{h}_i(x)\big)\right| \leq \frac{\|\delta\|_2}{\sigma} \leq \frac{r^\alpha_\sigma(x)}{\sigma} \tag{D.1}$$

for all i. Applying (D.1) for $i = y$ yields that

$$\Phi^{-1}\big(\hat{h}_y(x+\delta)\big) \geq \Phi^{-1}\big(\hat{h}_y(x)\big) - \frac{r^\alpha_\sigma(x)}{\sigma}, \tag{D.2}$$

and, since Φ^{-1} is monotonically increasing and $\hat{h}_i(x) \leq \hat{h}_{y'}(x)$ for all $i \neq y$, applying (D.1) to $i \neq y$ gives that

$$\Phi^{-1}\big(\hat{h}_i(x+\delta)\big) \leq \Phi^{-1}\big(\hat{h}_i(x)\big) + \frac{r^\alpha_\sigma(x)}{\sigma} \leq \Phi^{-1}\big(\hat{h}_{y'}(x)\big) + \frac{r^\alpha_\sigma(x)}{\sigma}. \tag{D.3}$$

Subtracting (D.3) from (D.2) gives that

$$\Phi^{-1}\big(\hat{h}_y(x+\delta)\big) - \Phi^{-1}\big(\hat{h}_i(x+\delta)\big) \geq \Phi^{-1}\big(\hat{h}_y(x)\big) - \Phi^{-1}\big(\hat{h}_{y'}(x)\big) - \frac{2r^\alpha_\sigma(x)}{\sigma}$$

for all $i \neq y$. By the definitions of μ_α, $r^\alpha_\sigma(x)$, and $\hat{h}(x)$, the right-hand side of this inequality equals zero, and hence, since Φ is monotonically increasing, we find that $\hat{h}_y(x+\delta) \geq \hat{h}_i(x+\delta)$ for all $i \neq y$. Therefore,

$$\begin{aligned}\frac{h_y(x+\delta)}{1+\mu_\alpha} =& \mathbb{E}_{\xi\sim\mathcal{N}(0,\sigma^2 I_d)}\left[\frac{\overline{h}_y(x+\delta+\xi)}{1+\mu_\alpha}\right] = \hat{h}_y(x+\delta)\\ \geq& \hat{h}_i(x+\delta) = \mathbb{E}_{\xi\sim\mathcal{N}(0,\sigma^2 I_d)}\left[\frac{\overline{h}_i(x+\delta+\xi)+\mu_\alpha}{1+\mu_\alpha}\right] = \frac{h_i(x+\delta)+\mu_\alpha}{1+\mu_\alpha}.\end{aligned}$$

Hence, $h_y(x+\delta) \geq h_i(x+\delta) + \mu_\alpha$ for all $i \neq y$, so $h(\cdot)$ is certifiably robust at x with margin $\mu_\alpha = \frac{1-\alpha}{\alpha}$ and radius $r^\alpha_\sigma(x)$. Therefore, by Lemma 1, it holds that $\arg\max_i h^\alpha_i(x+\delta) = y$ for all $\delta \in \mathbb{R}^d$ such that $\|\delta\|_2 \leq r^\alpha_\sigma(x)$, which concludes the proof. □

D.4. Proof of Theorem 3

Theorem 3 (restated). *Let $\epsilon > 0$, $(x_1, y_1), (x_2, y_2) \sim \mathcal{D}$, and $y_1 \neq y_2$ (i.e., each input corresponds to a unique true label). Assume that $h_i(\cdot)$, $\|\nabla h_i(\cdot)\|_{p*}$, and $\|\nabla g_i(\cdot)\|_{p*}$ are all bounded and that there does not exist $z \in \mathbb{R}^d$ such that $\|z - x_1\|_p \leq \epsilon$ and $\|z - x_2\|_p \leq \epsilon$. Then, there exists a function $\alpha(\cdot)$ such that the assembled classifier $h^\alpha(\cdot)$ satisfies*

$$\mathbb{P}_{\substack{(x,y)\sim\mathcal{D}\\ \delta\sim\mathcal{F}}}\left[\arg\max_{i\in[c]} h^\alpha_i(x+\delta) = y\right] \geq \max\left\{\begin{matrix}\mathbb{P}_{(x,y)\sim\mathcal{D},\delta\sim\mathcal{F}}\left[\arg\max_{i\in[c]} g_i(x+\delta) = y\right],\\ \mathbb{P}_{(x,y)\sim\mathcal{D},\delta\sim\mathcal{F}}\left[\arg\max_{i\in[c]} h_i(x+\delta) = y\right]\end{matrix}\right\},$$

where $\mathcal{F}$ is an arbitrary distribution that satisfies $\mathbb{P}_{\delta\sim\mathcal{F}}\left[\|\delta\|_p > \epsilon\right] = 0$.

Proof. Since it is assumed that the perturbation balls of the data are non-overlapping, the true label y corresponding to each perturbed data $x+\delta$ with the property $\|\delta\|_p \leq \epsilon$ is unique. Therefore, the indicator function

$$\alpha(x+\delta) = \begin{cases} 0 & \text{if } \arg\max_{i\in[c]} g_i(x+\delta) = y, \\ 1 & \text{otherwise,} \end{cases}$$

satisfies that

$$\begin{aligned} \alpha(x+\delta) = 0 \quad &\text{if} \quad \arg\max_{i\in[c]} g_i(x+\delta) = y, \\ \alpha(x+\delta) = 1 \quad &\text{if} \quad \arg\max_{i\in[c]} g_i(x+\delta) \neq y \text{ and } \arg\max_{i\in[c]} h_i(x+\delta) = y. \end{aligned}$$

Therefore, it holds that

$$\begin{aligned} h_i^\alpha(x+\delta) = g_i(x+\delta) \quad &\text{if} \quad \arg\max_{i\in[c]} g_i(x+\delta) = y, \\ h_i^\alpha(x+\delta) = h_i(x+\delta) \quad &\text{if} \quad \arg\max_{i\in[c]} g_i(x+\delta) \neq y \text{ and } \arg\max_{i\in[c]} h_i(x+\delta) = y, \end{aligned}$$

implying that

$$\arg\max_{i\in[c]} h_i^\alpha(x+\delta) = y \quad \text{if} \quad \big(\arg\max_{i\in[c]} g_i(x+\delta) = y \ \text{ or } \arg\max_{i\in[c]} h_i(x+\delta) = y\big),$$

which leads to the desired statement. □

References

[1] G. Adam and R. Speciel. Evaluating ensemble robustness against adversarial attacks. *arXiv preprint arXiv:2005.05750*, 2020.

[2] M. A. Ahmadi, R. Dianat, and H. Amirkhani. An adversarial attack detection method in deep neural networks based on re-attacking approach. *Multimedia Tools and Applications*, 80(7):10985–11014, 2021.

[3] A. Aldahdooh, W. Hamidouche, and O. Déforges. Selective and features based adversarial example detection. *arXiv preprint arXiv:2103.05354*, 2021.

[4] A. Aldahdooh, W. Hamidouche, S. A. Fezza, and O. Déforges. Adversarial example detection for dnn models: A review and experimental comparison. *arXiv preprint arXiv:2105.00203*, 2021.

[5] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

[6] Y. Balaji, T. Goldstein, and J. Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019.

[7] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*, 2017.

[8] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[9] F. Carrara, F. Falchi, R. Caldelli, G. Amato, and R. Becarelli. Adversarial image detection in deep neural networks. *Multimedia Tools and Applications*, 78(3):2815–2835, 2019.

[10] K. T. Co, D. Martinez-Rego, Z. Hau, and E. C. Lupu. Jacobian ensembles improve robustness trade-offs to adversarial attacks. In *Artificial Neural Networks and Machine Learning*, 2022.

[11] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[12] F. Croce, S. Gowal, T. Brunner, E. Shelhamer, M. Hein, and T. Cemgil. Evaluating the adversarial robustness of adaptive test-time defenses. *arXiv preprint arXiv:2202.13711*, 2022.

[13] F. Croce and M. Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, 2020.

[14] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, 2020.

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

[16] S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[17] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. Mann. Improving robustness using generated data. *arXiv preprint arXiv:2110.09468*, 2021.

[18] S. Gowal, J. Uesato, C. Qin, P.-S. Huang, T. Mann, and P. Kohli. An alternative surrogate loss for pgd-based adversarial testing. *arXiv preprint arXiv:1910.09338*, 2019.

[19] X. Jia, Y. Zhang, B. Wu, K. Ma, J. Wang, and X. Cao. LAS-AT: Adversarial training with learnable attack strategy. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[20] A. Kumar, T. Ma, P. Liang, and A. Raghunathan. Calibrated ensembles can mitigate accuracy tradeoffs under distribution shift. In *The Conference on Uncertainty in Artificial Intelligence*, 2022.

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

[22] C. Liu, Y. Dong, W. Xiang, X. Yang, H. Su, J. Zhu, Y. Chen, Y. He, H. Xue, and S. Zheng. A comprehensive study on robustness of image classification models: Benchmarking and rethinking. *arXiv preprint arXiv:2302.14301*, 2023.

[23] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh. Towards robust neural networks via random self-ensemble. In *European Conference on Computer Vision*, 2018.

[24] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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

[26] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. In *International Conference on Learning Representations*, 2017.

[27] M. Pagliardini, G. Manunza, M. Jaggi, and T. Chavdarova. Improved generalization-robustness trade-off via uncertainty targeted attacks. Preprint, 2022.

[28] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, 2019.

[29] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

[30] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Annual Conference on Neural Information Processing Systems*, 2019.

[31] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially robust generalization requires more data. In *Annual Conference on Neural Information Processing Systems*, 2018.

[32] V. Sehwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, and P. Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International Conference on Learning Representations*, 2022.

[33] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! In *Annual Conference on Neural Information Processing Systems*, 2019.

[34] F. Tramèr, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. In *Annual Conference on Neural Information Processing Systems*, 2020.

[35] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

[36] H. Wang, T. Chen, S. Gui, T. Hu, J. Liu, and Z. Wang. Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free. In *Annual Conference on Neural Information Processing Systems*, 2020.

[37] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, and S. Yan. Better diffusion models further improve adversarial training. *arXiv preprint arXiv:2302.04638*, 2023.

[38] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019.