This WACV 2023 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

DE-CROP: Data-efficient Certified Robustness for Pretrained Classifiers

Gaurav Kumar Nayak* Ruchit Rawal* Anirban Chakraborty

Department of Computational and Data Sciences
Indian Institute of Science, Bangalore, India
{gauravnayak, ruchitrawal, anirban}@iisc.ac.in

Abstract

Certified defense using randomized smoothing is a popular technique to provide robustness guarantees for deep neural networks against l_2 adversarial attacks. Existing works use this technique to provably secure a pretrained non-robust model by training a custom denoiser network on entire training data. However, access to the training set may be restricted to a handful of data samples due to constraints such as high transmission cost and the proprietary nature of the data. Thus, we formulate a novel problem of "how to certify the robustness of pretrained models using only a few training samples". We observe that training the custom denoiser directly using the existing techniques on limited samples yields poor certification. To overcome this, our proposed approach (DE-CROP)[1] *generates class-boundary and interpolated samples corresponding to each training sample, ensuring high diversity in the feature space of the pretrained classifier. We train the denoiser by maximizing the similarity between the denoised output of the generated sample and the original training sample in the classifier's logit space. We also perform distribution level matching using domain discriminator and maximum mean discrepancy that yields further benefit. In white box setup, we obtain significant improvements over the baseline on multiple benchmark datasets and also report similar performance under the challenging black box setup.*

1. Introduction

Given large amounts of training data (such as ImageNet [8]) and compute power (powerful GPUs [29]), deep models are highly accurate on the respective tasks for which it is trained. However, these models can easily get fooled when they encounter adversarial images as input that are crafted using an adversarial attack [30]. A lot of efforts have been made in the literature to secure the models against adversarial attacks. Adversarial training [22, 13] is one of the most common ways to provide empirical defense where the models are trained to minimize the maximum training loss induced by adversarial samples. Such defenses are heuristic-based and are only robust to known or specific adversarial attacks. Powerful adversaries easily break them, hence are not truly robust against adversarial perturbations [1, 32, 7]. This motivated the researchers to develop methods where a trained model can be guaranteed to have a constant prediction in the input neighborhood. Such methods that provide formal guarantees are called certified defense methods [25, 34, 9, 17, 23].

Randomized smoothing [3, 21, 19, 20, 6] is a certified defense technique used to provide provable robustness against l_2 adversarial perturbations. It outperforms other certification methods and is also scalable to deep neural networks due to its architectural independence. Using randomized smoothing, any base classifier can be converted to a smoothed classifier, which is certifiably robust against l_2 attacks as it has the desirable property of constant predictions within the l_2 ball around the input. The prediction by the smoothed classifier on an input image is simply the most probable class predicted by the base classifier on random gaussian perturbations of the input. Note that the higher the probability with which the base classifier predicts the most probable class as the correct class, the higher the certified radius [6]. However, any vanilla trained / off-the-shelf classifier would not be robust to the input corrupted by gaussian noise. The predicted most probable class can be incorrect or may be predicted with very low confidence leading to poor certification. Thus, models are often trained from scratch using the gaussian perturbed samples [6] as augmentation, and also with adversarial training [26].

Training from scratch on gaussian-noise augmentation is not always a viable option, especially when the large pretrained models are shared as an API, either as a white box or a black box. Also, retraining these large cumbersome models adds a lot of computational overload. To avoid this, Salman *et al*. [27] proposed a "denoised smoothing" technique where a custom-trained denoiser is prepended before the pretrained classifier. Though their approach provides

*denotes equal contribution.

[1] Project Page: https://sites.google.com/view/decrop

certified robustness to pretrained models, they use the entire training data for training the denoiser. In fact, all the previous certification methods also assume the availability of entire training data. This assumption is unrealistic as the API provider may not share the whole large-scale training set. Due to heavy transmission costs associated with complete training data or proprietary reasons, they may provide access to only a few training samples. In such cases, when we perform denoised smoothing [27] directly, the certified accuracy decreases significantly as the number of training samples reduces from 100% to 1% as shown in Fig. 1.

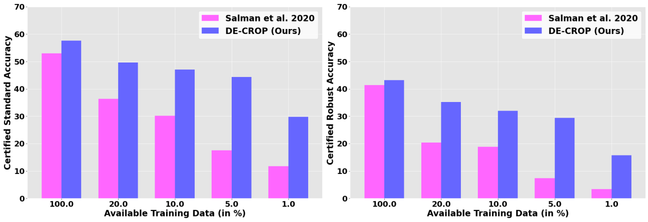

Figure 1. The certified accuracy (both standard and robust) of existing method drops significantly as the training set size of CIFAR-10 reduces. Our method (*DE-CROP*) overcomes this issue by obtaining significant gains in certified performance against l_2 perturbations within radius 0.25, across different training data budgets. We use gaussian noise with σ as 0.5 for certification.

In this work, we attempt to provide certified robustness to a pretrained non-robust model in the challenging setting of limited training data. We first explore whether the performance of denoised smoothing can be improved using different approaches such as weight decay as a regularization scheme or even generate extra data using augmentation and mix-up strategies [36] to avoid overfitting. However, these traditional approaches only show a marginal improvement (refer to Tables 1, 2). Hence, we propose our approach of **d**ata-**e**fficient **c**ertified **ro**bustness (dubbed as '*DE-CROP*') (refer to Sec. 4 and Fig. 2, 3) that generates better samples, yielding diverse features on the pretrained classifier similar to complete training data.

Since generative models are hard to train and perform poorly (on downstream tasks) in the presence of limited data, we adopt a simple and intuitive approach to generate additional data to minimize overfitting. We achieve this by generating adversarial samples (corresponding to our limited data) that we term 'boundary' samples. Adversarial samples act as an upper bound to the decision boundary [24] while maintaining the semantic content of the image, allowing the denoiser to learn from samples in its neighborhood. Moreover, we also generate interpolated samples (lying between original and boundary samples) that further increase the data diversity in the feature space. We train the denoiser by maximizing the similarity between the pretrained classifier's features on the original data and denoised perturbed generated/original data. We achieve this by performing orthogonal modifications at two granularities: a) instance-level (using cosine-similarity) and b) distribution-level (using Maximum Mean Desprecancy [14] and negative gradients from a Domain-discriminator [11]). In our experiments, we observed that although cosine-similarity improved denoised performance by exploiting the discrimnativeness of the pretrained classifier, its benefits were limited as it only operates at an instance level. Thus, inspired by fundamental ideas (such as maximum mean discrepancy and domain discriminator) in domain adaptation, we formulate the objective of obtaining correct predictions on denoised images by reducing the distribution discrepancy between feature representations of the original clean inputs and the denoised gaussian perturbed output. As shown in Fig. 1, our method DE-CROP significantly improves certified performance across different sample budgets (100%, 20%, 10%, 5%, 1%) on CIFAR-10 compared to [27].

Our contributions are summarized as follows:

- Given only limited training data, we provide robustness guarantees for a non-robust pretrained classifier against l_2 perturbations on both white-box and black-box setups. To the best of our knowledge, we are the first to provide certified adversarial defense using only the few training samples.

- To mitigate overfitting on limited training data, we propose a novel sample-generation strategy that synthesize 'boundary' and 'interpolated' samples (Sec. 4.1) to augment the limited training data, leading to improved feature-diversity on the pretrained classifier.

- The denoiser network trained with regular cross entropy loss provides limited benefit. To enhance the performance further, we proposed additional losses (Sec. 4.2) that align the feature representations of original and denoised gaussian perturbed generated/original samples at multiple granularities (both instance and distribution levels).

- We show benefit of our generated samples (Sec. 5.2) along with contributions from each of the proposed losses (Sec. 5.3), by reporting significant improvements observed across diverse sample budgets and noise levels in both white-box and black-box settings.

2. Related Works

We broadly categorize the relevant works that provides adversarial robustness and briefly discuss them below:

Empirical Robustness: Empirically motivated adversarial robustness defenses can be broadly classified into: a) adversarial training (AT) and b) non-adversarial training regularizations. AT [22, 13, 30] improves robustness by augmenting the training data with adversarial samples generated by a particular threat model. Although AT is widely regarded as the best empirical defense, it suffers from high-computational costs due to generation of adversarial samples at training time. Non-AT based approaches [5] attempt to reduce the computational burden (usually at the cost of drop in adversarial robustness) by mimicking properties observed in robust networks explicitly. AT is also highly dependent on the quantity of training data. Aditi *et al.* [4] demonstrated that using additional unlabeled data in a pseudo-labelling setup led to a significant increase in adversarial robustness. However, since performance of pseudo-labelling itself depends on the amount of labeled data, the performance of their technique drops considerably as labeled data is reduced. To alleviate this problem, Sehwag *et al.* [28] illustrated the benefit of using additional data generated from generative models for improving adversarial robustness.

Since the empirical defenses are designed based on heuristics, they can be easily fooled as stronger adversarial attacks are developed in the future. In contrast, we attempt to provably robustify pretrained classifier against l_2 adversarial attacks under the challenging constraint of limited training samples.

Certified Robustness: Unlike empirical defense, here the model predictions on the neighborhood region lying within ball of small radius around the input sample, is guaranteed to not vary and remain constant. The methods that provide certification are either 'exact' [10, 9, 17, 31] or 'conservative' [34, 37]. The former is not scalable to large architectures, is compute-intensive, and often uses less expressive networks but can verify with guarantees about the existence of any adversarial samples if lying within the radius of input. The latter is more scalable and requires less computation but can incorrectly decline the certification even if no adversarial sample is present. Both techniques require either customized or specific architectures, hence are not suitable for modern deep architectures.

Randomized smoothing is a popular technique that does not have any architectural dependency and was initially used as heuristic defense [3, 21]. It was first shown to provide certified guarantees by Lecuyer *et al.* [19] where techniques from 'differential privacy' were used for certification. It was later improved by Li *et al.* [20] who provided better guarantees using ideas from 'information theory'. Both these methods have lower guarantees on the smoothed classifier. Cohen *et al.* [6] provided a tight certified guarantee against l_2 norm adversarial perturbations. After that, the certified accuracy was further improved by using adversarial training techniques in the randomized smoothing framework [26]. However, all these techniques trained the classifier from scratch while providing certified robustness. Recently, Salman *et al.* [27] provided provable robustness to pretrained models by appending a custom trained denoiser before it. We also train the custom denoiser but only using few training samples. Unlike [27], our limited data setup is more challenging and directly using their method yields poor certification results. Our generated samples with the added domain discriminator optimized using our proposed losses, handles the overfitting on the denoiser quite well and gives significant improvements on certified accuracy.

We now discuss the necessary preliminaries to give required background before explaining our approach.

3. Preliminaries

Notations: The complete original training dataset is denoted by $D_o = \{D_{train}, D_{test}\}$, where D_{train} and D_{test} are the training set and the test set respectively. The base classifier B_c is pretrained on the entire D_{train} which consists of N training samples. The API provider has granted public access to the trained B_c which can be used by clients to obtain predictions. However, only a limited amount of D_{train} (denoted by D^{lim}_{train}) is shared to the clients. D^{lim}_{train} is only $k\%$ of D_{train} containing N^k training samples such that $N^k \ll N$). The same relationship holds for each class of the classifier B_c i.e. for any class c: $N^k_c \ll N_c$ and N^k_c is $k\%$ of N_c. An i^{th} sample of D^{lim}_{train} (i.e. x^i_o) with label y^i_o is perturbed by gaussian noise which is denoted by $\bar{x}^i_o = x^i_o + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, constituting the collection of perturbed samples as $\bar{D}^{lim}_{train}$.

The penultimate features, logits and label predictions from the pretrained classifier B_c on x^i_o are denoted by $F_{B_c}(x^i_o)$, $label(B_c(x^i_o))$ and $B_c(x^i_o)$ respectively. Similarly, $label(S_c(x^i_o))$ is the label predicted by smoothed classifier S_c on input x^i_o. The classifier B_c is converted to smoothed classifier S_c, which is used for certified defense via randomized smoothing. The certified radius of an i^{th} test sample is R^i_c. The evaluations are performed on l_2 perturbations at a radius r denoted by l^r_2. The denoiser network D_n and domain discriminator D_d, are parameterized by θ and ϕ respectively. The boundary sample and interpolated sample generated using an i^{th} training sample of D^{lim}_{train} (i.e. x^i_o) are represented by x^i_b and x^i_{int} respectively.

Randomized Smoothing (RS): This technique is used to build a new smoothed classifier S_c from the given base classifier B_c. For any i^{th} sample of D^{lim}_{train} (i.e. x^i_o), the output of the classifier S_c corresponding to input x^i_o is the most likely class that has the highest probability to be get predicted by B_c on $\bar{x}^i_o$.

$$label(S_c(x^i_o)) = \underset{c \in C}{\operatorname{argmax}} \quad Prob(label(B_c(x^i_o + \epsilon)) = c) \\ \text{where} \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{1}$$

Here, σ is a hyperparameter which controls the noise level and C is the set of unique target labels in D^{lim}_{train}. The process of RS does not assume B_c's architecture and thus permits the B_c to be any arbitrary large deep neural network.

Certified Robustness using RS: Lecuyer *et al.* [19] and Li *et al.* [20] gave robustness guarantees for the smoothed classifier S_c using RS, but they were loose as S_c was provably more robust than the obtained guarantees. Cohen *et al.* [6] gave a tight bound on l_2 robust guarantees using RS.

If the prediction on the base classifier B_c for the gaussian perturbed copies of x_o^i i.e. $\mathcal{N}(x_o^i, \sigma^2 I)$ is c_1 as the 'most probable class' with probability p_1 and c_2 as "runner-up" class with p_2, then the smoothed classifier S_c is provably robust around the input x_o^i within the radius $R_c = \sigma/2(\phi^{-1}(p_1) - \phi^{-1}(p_2))$. Here R_c is the certified radius as the predictions are guaranteed to remain constant inside the radius and ϕ^{-1} denotes the inverse of standard gaussian CDF. It is impossible to calculate p_1 and p_2 exactly when B_c is a deep neural network. Hence, Cohen *et al.*, estimates lower bound on p_1 ($\underline{p_1}$) and upper bound on p_2 ($\overline{p_2}$) using Monte Carlo technique.

Theorem [Cohen *et al.* [6]]: *Let B_c be any function that maps the input to one of the output class labels. Let S_c be defined as in eq. 1. The noise ϵ is sampled from the normal distribution i.e. $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. If $c_1 \in C$ and $\underline{p_1}$, $\overline{p_2} \in [0, 1]$ holds the below inequality*:

$$Prob(label(B_c(x_o^i + \epsilon)) = c_1) \geq \underline{p_1} \geq \overline{p_2} >= \max_{c \neq c_1} Prob(label(B_c(x_o^i + \epsilon) = c) \tag{2}$$

Then $label(S_c(x_o^i + \epsilon) = c_1 \forall \left\lVert \epsilon \right\rVert_2 < R_c$*, where*

$$R_c = \sigma/2(\phi^{-1}(\underline{p_1}) - \phi^{-1}(\overline{p_2})) \tag{3}$$

In practice, Cohen *et al.* used $R_c = \sigma\phi^{-1}(\underline{p_1})$ for $\underline{p_1} > 1/2$, assuming $\overline{p_2} = 1 - \underline{p_1}$, otherwise abstained with $R_c = 0$. These above expressions can be derived using the "Neyman-Pearson" lemma for which we refer the reader to [6]. Now, we discuss our proposed approach in detail in the next section.

4. Proposed Approach

We aim to provide certified robustness to the given pretrained base classifier B_c. However, obtaining certification using randomized smoothing expects the model B_c to be robust against the input perturbations with the random Gaussian noise, which may not be the case with the model B_c supplied by the API provider. In order to make the base model B_c appropriate for randomized smoothing based certification without modifying/retraining B_c, a denoiser network D_n is prepended to B_c. Thus, $B_c \circ D_n$ is the new base classifier using which the prediction on smoothed classifier S_c on an i^{th} training sample is defined as follows:

$$label(S_c(x_o^i)) = \underset{c \in C}{\operatorname{argmax}} Prob(label(B_c(D_n(x_o^i + \epsilon))) = c)$$
$$\text{where} \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{4}$$

The above smoothed classifier S_c is provably robust against the l_2 perturbations with certified radius R_c (refer to Sec. 3). For high performance on the certified defense, high R_c is required, which is directly proportional to $\underline{p_1}$ (eq. 3). The probability ($\underline{p_1}$) of predicting the most probable class (c_1) i.e. confidence depends on the performance of the denoiser network D_n. However, D_n trained on the given limited training data D_{train}^{lim} using the existing technique [27] yields poor certification (shown in Fig. 1). Even when we attempt to minimize the overfitting of D_n on D_{train}^{lim} ($\because |D_{train}^{lim}| \ll |D_{train}|$) using different traditional approaches such as weight decay, augmentation, and mix-up strategies, we observe only a minor improvement (refer to Tables 1, 2) in certified accuracy. Hence, we propose our method ('DE-CROP') that overcomes this limitation by crafting boundary and interpolated samples using D_{train}^{lim}, followed by using them to train the denoiser with appropriate losses.

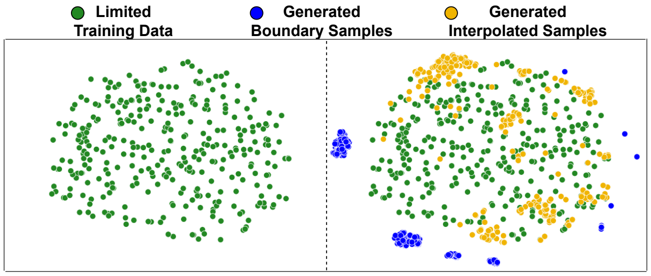

Figure 2. t-SNE visualization of the pretrained base classifier's features for training samples of a particular class of CIFAR-10. Our generated samples (interpolated and boundary) increases the feature diversity of limited-original training data.

4.1. Generation of Boundary & Interpolated sample

In Fig. 2, we show the visualization of the logit layer features of the pretrained base classifier B_c on a particular class via t-SNE plot. The class-features corresponding to available limited class samples of D_{train}^{lim} is shown in green color (on left). We focus on improving the feature diversity of the limited training samples. For this, we first estimate class-boundary samples that would have respective features in the boundary region of the t-SNE.

Adversarial attacks [22] synthesize samples via an optimization procedure that carefully perturbs the input sample with a small human-imperceptible noise (i.e., adversarial noise). The model gets fooled on such samples (i.e., adversarial samples) as the prediction gets flipped to some other class. As these samples cross the decision boundary, which are constructed by adding a small noise to the input original sample, they often lie very close to the decision boundary. Moreover, they are human-imperceptible and preserve the class semantics of the input class sample. Hence, adversarial samples serve as a good candidate for a proxy of class-boundary samples. For any i^{th} training sample of D_{train}^{lim} (i.e., x_o^i), we obtain boundary sample (x_b^i) by computing

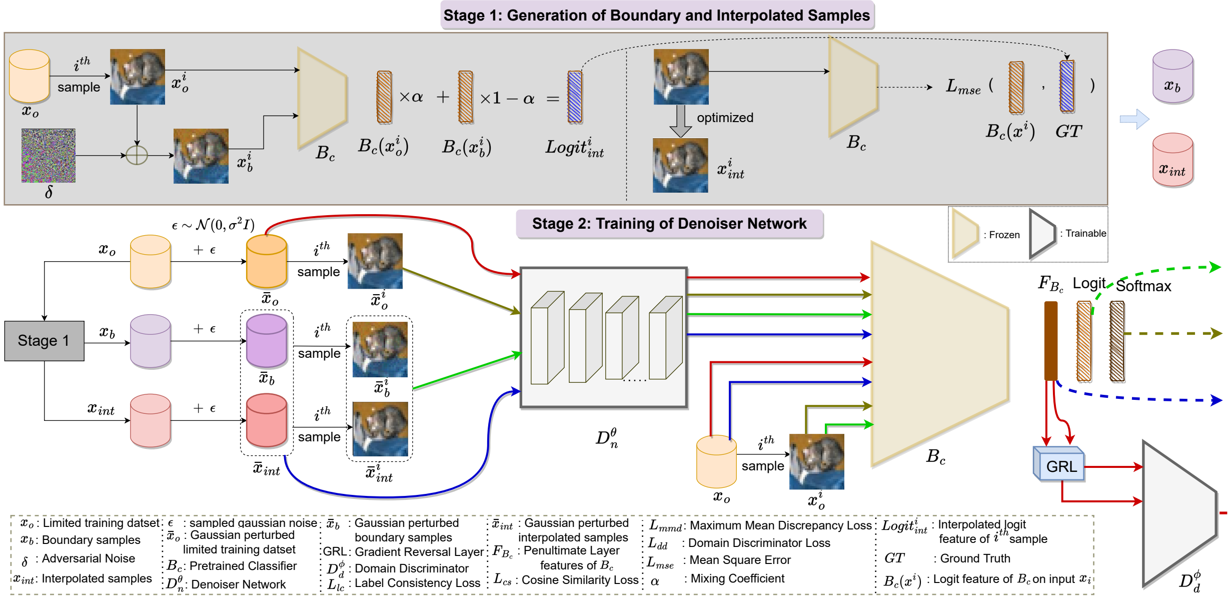
Stage 1: Generation of Boundary and Interpolated Samples
i^{th} sample
x_o
x_o^i
δ
x_b^i
B_c
$B_c(x_o^i)$
$\times\alpha$ + $\times 1-\alpha$ =
$B_c(x_b^i)$
$Logit_{int}^i$
optimized
x_{int}^i
L_{mse} (,)
$B_c(x^i)$
GT
x_b
x_{int}
Stage 2: Training of Denoiser Network
: Frozen
: Trainable
$\epsilon \sim \mathcal{N}(0, \sigma^2 I)$
Stage 1
x_b
x_{int}
+ ϵ
$\bar{x}_o$
$\bar{x}_b$
$\bar{x}_{int}$
$\bar{x}_o^i$
$\bar{x}_b^i$
$\bar{x}_{int}^i$
D_n^θ
F_{B_c}
Logit
Softmax
L_{lc}
L_{cs}
L_{mmd}
GRL
L_{dd}
D_d^ϕ
x_o : Limited training datset
x_b : Boundary samples
δ : Adversarial Noise
x_{int} : Interpolated samples
ϵ : sampled gaussian noise
$\bar{x}_o$: Gaussian perturbed limited training datset
B_c : Pretrained Classifier
D_n^θ : Denoiser Network
$\bar{x}_b$: Gaussian perturbed boundary samples
GRL : Gradient Reversal Layer
D_d^ϕ : Domain Discriminator
L_{lc} : Label Consistency Loss
$\bar{x}_{int}$: Gaussian perturbed interpolated samples
F_{B_c} : Penultimate Layer features of B_c
L_{cs} : Cosine Similarity Loss
L_{mmd} : Maximum Mean Discrepancy Loss
L_{dd} : Domain Discriminator Loss
L_{mse} : Mean Square Error
α : Mixing Coefficient
$Logit_{int}^i$: Interpolated logit feature of i^{th} sample
GT : Ground Truth
$B_c(x^i)$: Logit feature of B_c on input x_i

Figure 3. Different stages involved in the proposed method (*DE-CROP*) that provides robustness guarantees on pretrained classifier (B_c) against l_2 perturbations using only limited training samples (x_o). Samples crafted via adversarial attack act as proxy *boundary* samples (x_b). Using x_o and x_b, *interpolated* samples (x_{int}) are generated in stage 1. The generated samples along with limited training data are used in stage 2 to train the denoiser network (D_n^θ) by aligning the feature representation at instance and distribution levels using L_{lc}, L_{cs}, L_{mmd} and L_{dd} losses. The forward pass to compute these losses are shown by different colors.

adversarial noise δ for which the following relation holds:

$$x_b^i = x_o^i + \delta, \\ \|\delta\|_\infty < \epsilon, \quad \epsilon > 0, \quad label(B_c(x_b^i)) \neq label(B_c(x_o^i)) \tag{5}$$

Next, we obtain interpolated features by performing a mixup between the features of generated boundary sample x_b^i and original training sample x_o^i as follows:

$$Logit_{int}^i = \alpha \times B_c(x_o^i) + (1-\alpha) \times B_c(x_b^i) \tag{6}$$

Here α is the mixing coefficient. The interpolated features, the class-boundary features, and the original limited class-sample features are highlighted in yellow, blue, and green colors respectively, in the t-SNE plot (Fig. 2). These interpolated features being label preserving help to improve the feature-diversity leading to improved certified accuracy (refer to Sec. 5.2). We craft the input sample corresponding to the interpolated feature $Logit_{int}^i$ (i.e., x_{int}^i) by perturbing the original sample x_o^i to match its feature response to $Logit_{int}^i$ using mean square error loss (L_{mse}). Mathematically, we obtain interpolated sample x_{int}^i as follows:

$$x_{int}^i \leftarrow \min_x \quad L_{mse}(B_c(x), Logit_{int}^i) \tag{7}$$

where x is initialized with x_o^i and kept trainable with ground truth as $Logit_{int}^i$. The model B_c is non-trainable, but the gradients are allowed to backpropagate from the model to update the input x. Thus, we obtain the boundary and the interpolated samples corresponding to each training sample.

Refer supplementary for visualization of both the boundary and interpolated samples in the input space. All of them preserve the class semantics. Our generated boundary samples x_b and interpolated samples x_{int} are used along with limited training samples x_o to train the denoiser network D_n, which we discuss in the subsequent subsection.

4.2. Training of the denoiser network (D_n)

The denoiser network D_n is attached before the pretrained base classifier B_c to make it suitable for randomized smoothing. Apart from this, we also add a domain discriminator network D_d whose input is the normalized features of the penultimate layer of B_c. The discriminator D_d learns to distinguish between distribution of clean samples and distribution of denoised outputs of gaussian perturbed input samples. Motivated from domain adaptation literature [12], we use gradient reversal layer (GRL) before feeding the normalized penultimate layer features to the discriminator that allows normal forward pass but reverses the direction of gradient in the backward pass. As a consequence, this negative gradients backpropagates to the denoiser network D_n that helps it to produce denoised output which yields indistinguishable domain-invariant features on the pretrained B_c classifier.

Refer to supplementary for architectural details of the networks D_n and D_d. The overall steps involved in the proposed framework ('DE-CROP') are also shown in Fig. 3. The network is trained with our generated data and limited training data D_{train}^{lim} by using different losses aimed at different objectives - label consistency to ensure correct predictions on B_c and matching of high-level feature similarity obtained on B_c for denoised output of gaussian perturbed input and clean original training samples both at the sample level and distribution level. The respective losses are described below:

Ensuring label consistency: Similar to [27], we use cross entropy loss (L_{ce}) to ensure that the label predicted by the pretrained network B_c on original clean data and denoised output of its gaussian perturbed counterpart are same.

$$L_{lc} = 1/N^k \sum_{i=1}^{N^k} L_{ce}(softmax(B_c(D_n(\bar{x}_o^i))), \quad label(B_c(x_o^i))) \tag{8}$$

Enforcing feature similarity at sample level: The pretrained base classifier B_c trained on complete training data D_{train} is highly discriminative. To leverage this on our crafted data, we use cosine similarity loss at the logits of the B_c network to encourage logit features on the denoised output of our generated data to be as discriminative as the features of the limited original training samples D_{train}^{lim} by maximizing this loss.

$$L_{cs} = 1/N^k \sum_{i=1}^{N^k} (\ CS(B_c(D_n(\bar{x}_b^i)), B_c(x_o^i)) + CS(B_c(D_n(\bar{x}_{int}^i)), B_c(x_o^i))\ ), \ \text{s.t. } CS(w,z) = \frac{w^T z}{\|w\| \|z\|} \tag{9}$$

Enforcing feature similarity at distribution level: Unlike L_{cs} which is applied at sample level, here we enforce distribution level matching between the set of our denoised generated data and the set of limited original training data by using maximum mean discrepancy (MMD) [14] loss on the normalized pre-logit layer of B_c network.

$$L_{mmd} = MMD(F_{B_c}(D_n(\bar{x}_b)), F_{B_c}(x_o)) + MMD(F_{B_c}(D_n(\bar{x}_{int})), F_{B_c}(x_o)) \tag{10}$$

Moreover, we also train the domain discriminator network D_d (parameterized by ϕ) using binary cross entropy loss (L_{bce}) to distinguish the distribution of gaussian perturbed samples and clean samples.

$$L_{dd} = \sum_{x^i \in D_{train}^{lim} \cup D_n(\bar{D}_{train}^{lim})} L_{bce}(D_d(F_{B_c}(x^i)), d^i) \tag{11}$$

Here, if $x^i \in D_{train}^{lim}$ then $d^i = 1$ and if $x^i \in D_n(\bar{D}_{train}^{lim})$ then $d^i = 0$. The negative gradients are backpropagated via GRL [11] (multiplies calculated gradient by -1), to update the parameters θ of denoiser network D_n such that features of limited training data D_{train}^{lim} and its corresponding denoised output of gaussian corrupted data ($D_n(\bar{D}_{train}^{lim})$) on network B_c are domain invariant.

Hence, the total loss can be written as follows:

$$L(D_n^{\theta}, D_d^{\phi}) = \beta_1 L_{lc} - \beta_2 L_{cs} + \beta_3 L_{mmd} + \beta_4 L_{dd} \tag{12}$$

At test time, the trained denoiser network D_n with optimal parameters θ^* prepended with base classifier B_c is used for evaluation.

5. Experiments

We demonstrate the effectiveness of our proposed approach (DE-CROP) on two widely-popular image-classification datasets, namely, CIFAR-10 [18] and Mini-ImageNet [33]. We limit the training set size of the datasets mentioned above by randomly selecting 1% and 10% samples respectively (ensuring class balance) from the D_{train}. Our baseline corresponds to training the denoiser using the L_{lc} loss (similar to [27]). We fix the selected samples and use a ResNet-110 and ResNet-12 [16] network (for CIFAR-10 and Mini-ImageNet respectively) as our pre-trained classifiers (B_c) with a value of σ as 0.25 for all our ablations and state-of-the-art comparisons, unless otherwise specified. Refer to supplementary for additional ablations on different values of noise strength σ (0.12, 0.50, 1.00), quantity of limited training data D_{train}^{lim} (5%, 10%, 20%, 100%) and choice of architecture for the pretrained classifier B_c. We set the weights for the final loss-equation (refer eq. 12) i.e. β_1, β_2, β_3, β_4 as 1, 4, 4, 1 respectively. In the subsequent subsections, we first show limited benefit of conventional techniques, followed by benefits of each component in DE-CROP and a comparison with state-of-the-art techniques.

5.1. Improving Certification on Limited Training Data via Conventional Techniques

D_n trained with the L_{lc} objective proposed by [27] severely overfits in the presence of limited data resulting in poor certification accuracy (refer Fig. 1). In this section, we explore whether conventional supervised-learning techniques such as explicit regularization and data-augmentation are equipped to meaningfully improve certification accuracy when dealing with limited data.

Method	**Standard Certified**	**Robust Certified**		
	(r=0.00)	(r=0.25)	(r=0.50)	(r=0.75)
Without Reg.	20.60	4.60	0.80	0.00
L_1 Reg.	22.60	5.80	0.60	0.00
L_2 Reg.	**27.80**	**7.80**	**0.80**	0.00

Table 1. Effect of adding weight decay regularizer (L_1 and L_2) on limited data certification against adversarial attacks. L_2 regularization obtains better certified standard and robust accuracies.

In Table 1, we regularize the D_n by applying L_1 and L_2 reg. (regularization) [15]. We observe that L_2 reg. improves certified performance, whereas L_1 reg. only results in a marginal improvement. Although L_2 reg. limits the overfitting effect in the presence of limited data, it can't help in improving the inherent data diversity. Thus, in Table 2, we explore whether traditional affine and specialized augmentation methods (such as mixup [36] and cutmix [35]), when combined with L_2 reg. further boost performance.

Rows 2-4 in Table 2 correspond to affine-transformations where the intensity of the augmentation is increased progressively (i.e. policy1<policy2<policy3). We also experiment with widely-popular augmentation techniques: 'mixup' and 'cutmix'. Surprisingly, we observe that policy 1(row-2; the lightest augmentation) performs the best, followed closely by 'no-aug.'. We hypothesize that if the augmentation-policy is too aggressive, B_c can make incorrect predictions leading to noisy gradients for the D_n to learn from, resulting in poor certification accuracy.

Method	Standard Certified	Robust Certified		
	(r=0.00)	(r=0.25)	(r=0.50)	(r=0.75)
No Aug.	27.80	7.80	0.80	0.00
Aug. (policy 1)	**29.80**	**9.20**	**1.40**	0.00
Aug. (policy 2)	26.40	7.40	1.00	0.00
Aug (policy 3)	21.00	3.00	0.20	0.00
Mixup	24.20	5.40	0.60	0.00
Cutmix	19.80	3.20	0.00	0.00

Table 2. Investigating the effect of augmentation at different intensity levels (policies), mixup and cutmix strategies in minimizing overfitting on limited training data. The augmentation with light intesnity (policy 1) yields marginal improvement against l2 perturbations of different radii compared to no-augmentation strategy.

Thus, a combination of policy 1 and L2-reg. improves the certified standard and robust accuracies. We use this combination as a baseline for further experiments upon which we make orthogonal improvements.

5.2. Effectiveness of our Generated Data

One of the critical reasons for the drop in certified accuracy is the lack of diversity in the limited data. We address this problem by generating synthetic samples that provide diversity in the feature space. As elaborated in the proposed section (refer Sec. 4.1): adversarial samples (termed as boundary-samples i.e. x_b) serve as a good candidate for this task since they flip the classifier prediction with minimum possible perturbations, thus allowing the D_n to train on samples from less-dense boundary regions. Moreover, we also generate samples whose features are an interpolation between the original and boundary samples, crafted by minimizing the L_{mse} loss between the interpolated logits and the generated sample logits (as described by eq. 7).

We empirically validate our motivation in Table 3, where we observe that using L_{cs} on both x_b^i and corresponding x_{int}^i leads to a massive improvement in performance over the baseline. Interestingly, the gain in performance when using only x_b^i is comparatively modest. This observation further reinforces our intuition regarding the complementary nature of the information provided by the x_b^i and x_{int}^i. Moreover, contrary to adversarial training that generates adversarial samples at every iteration during the training time, we only need to generate the boundary and interpolated samples once (amounting to negligible increase in training time), as pre-trained classifier B_c is fixed and not trainable.

Method	Standard Certified	Robust Certified		
	(r=0.00)	(r=0.25)	(r=0.50)	(r=0.75)
Baseline	29.80	9.20	1.40	0.00
Ours (with boundary samples)	31.60	7.80	1.00	0.20
Ours (with boundary + interpolated samples)	**48.80**	**22.00**	**6.00**	**0.80**

Table 3. Benefit of our generated samples in improving certification. Using boundary and interpolated samples, we obtain significant boost in certified accuracy on original and l2 perturbed data.

5.3. Distribution Alignment: Enhancing Certification in Limited Data Setup

In Table 4, we observe that using domain-discriminator (D_d) along with the previously introduced L_{cs} loss works quite well as the standard certified accuracy improves by 6% and certified robust accuracy improves by 4% (at r=0.25) compared to only using L_{cs}.

We also explore whether equipping the D_n with explicit distribution discrepancy losses such as L_{mmd} would also work well in combination with the D_d. Intuitively, applying L_{mmd} along with L_{cs} should make the job of D_d harder, leading to a better D_d. Consequently, improving B_c's feature representation via the negative gradients of domain-discriminator loss (L_{dd}).

We indeed observe this in Table 4, where using L_{mmd} in combination with the L_{cs} and the L_{dd} setup leads to an improvement in performance across both standard and robust accuracies (across radii). Thus, using L_{mmd} and L_{dd} in conjunction with the previously discussed L_{cs} and L_{lc} constitutes our final approach: DE-CROP

5.4. Comparison with state-of-the-art

In this section, we compare the effectiveness of our approach DE-CROP against state-of-the-art robustness certification techniques, namely, denoised-smoothing by Salman *et al.* [27] and gaussian-augmentation by Cohen *et al.* [6].

Since Salman *et al.* [27] don't report performance on limited data scenarios in their paper, we use the code from their official implementation[2] for evaluating their proposed method (D_n with L_{lc}) in the presence of only 1% (for CIFAR-10) and 10% (for Mini-ImageNet) D_{train}. Similarly, for Cohen *et al.* [6], we re-train the classifier with

[2] https://github.com/microsoft/denoised-smoothing

Method	Standard Certified	Robust Certified		
	(r=0.00)	(r=0.25)	(r=0.50)	(r=0.75)
Baseline	29.80	9.20	1.40	0.00
Ours (instance level)	48.80	22.00	6.00	0.80
Ours (instance + distribution level via discriminator)	54.00	26.00	6.80	1.80
Ours (instance + distribution level via discriminator and MMD)	**57.60**	**27.20**	**9.20**	**2.20**

Table 4. Besides performance gains with instance-level feature matching, we observe further improvements in certified standard and robust accuracy when distributions of denoised and clean data are aligned in feature space via domain-discriminator and MMD.

gaussian augmentation only on the available limited training data. Although Cohen *et al.* [6]'s technique is unfeasible for our problem setup as the API provider may not prefer re-training and replacing the deployed model, we still compare our performance as gaussian-augmentation often outperforms previous denoiser-based approaches in presence of full training data.

As shown in Fig. 4, our proposed approach DE-CROP comfortably outperforms Salman *et al.* [27] on CIFAR-10, improving the certified standard accuracy by 27% and consistently improving robust accuracy across radii. The performance of Cohen *et al.* [6] drops to 10% across all radii, indicating that the network behaves like a random baseline (i.e. predicting each class with equal probability irrespective of the input). We observe similar trends for Mini-ImageNet, where we comfortably outperform both Salman *et al.* [27] and Cohen *et al.* [6] further demonstrating the wide applicability of our approach.

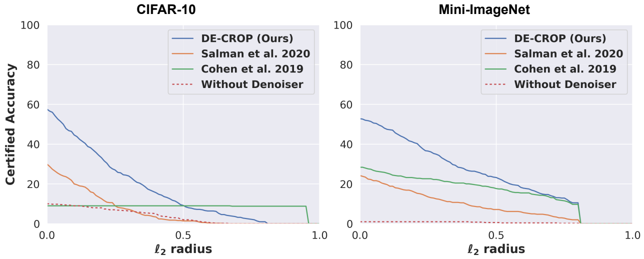

Figure 4. Performance comparison of our approach (*DE-CROP*) with other methods. We comfortably outperform on both the datasets. We also compared with Cohen *et al.* where the classifier is trained from scratch unlike ours where retraining is not feasible.

5.5. Certification of Black Box Classifiers with Limited Training Data

In previous sections we assumed white-box access to the pre-trained base classifier B_c i.e. we can backpropagate the gradients through B_c to optimize the denoiser (D_n). However, this may not always be the case as the API provider can limit access to only B_c's predictions (i.e. black-box) due to proprietary reasons. Since the black-box setup restricts the gradient information of B_c, we first use a black-box model stealing technique [2] to train a surrogate model: S_m. We use S_m which allows gradient backpropagation, to train D_n using our proposed approach DE-CROP (refer Fig. 3). Finally, for evaluation we use the denoiser (trained via S_m) to certify robustness of the black-box classifier B_c.

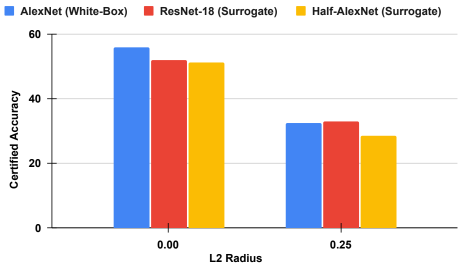

Figure 5. Investigating the efficacy of our method in black-box scenario (no access to pertrained model weights). We observe a minor drop in performance as compared to white-box setup for both certified standard accuracy (l_2^r=0.00) and robust accuracy (l_2^r=0.25).

In Fig. 5, we compare the certification performance (of B_c) using denoisers trained via S_m ('black-box access') to the denoiser trained directly on B_c ('white-box access'). We take B_c as Alexnet and two different choices for S_m, namely ResNet-18 and Half-Alexnet. Our method DE-CROP yields very similar performance in the black box setting across different architectures of S_m. Also, the performance drop compared to white box setting is marginal, highlighting the suitability of our technique even when the pretrained classifier weights are not shared.

6. Conclusion

We presented our approach (DE-CROP), which for the first time, solves the problem of providing provable robustness guarantees to a pretrained classifier in the challenging limited training data settings. Our method comprises a two-step process - a) generation of boundary and interpolated samples ensuring feature diversity and b) effectively utilizing the generated samples along with limited training samples for training the denoiser using the proposed losses to ensure feature similarity between the denoised output and clean data at two different granularities (instance level and distribution level). We validate the efficacy of the generated data as well as the contribution of individual losses by extensive ablations and experiments across CIFAR-10 and Mini-ImageNet datasets. Moreover, our method works quite well in the black-box setting as it provides similar certification performance compared to the white-box setup.

Acknowledgements This work is partially supported by a Young Scientist Research Award (Sanction no. 59/20/11/2020-BRNS) from DAE-BRNS, India.

References

[1] Anish Athalye and Nicholas Carlini. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*, 2018.

[2] Antonio Barbalau, Adrian Cosma, Radu Tudor Ionescu, and Marius Popescu. Black-box ripper: Copying black-box models using generative evolutionary algorithms. *Advances in Neural Information Processing Systems*, 33:20120–20129, 2020.

[3] Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287, 2017.

[4] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Alvin Chan, Yi Tay, Yew Soon Ong, and Jie Fu. Jacobian adversarially regularized networks for robustness. *arXiv preprint arXiv:1912.10185*, 2019.

[6] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[7] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[10] Matteo Fischetti and Jason Jo. Deep neural networks as 0-1 mixed integer linear programs: A feasibility study. *arXiv preprint arXiv:1712.06174*, 2017.

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[14] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[15] Stephen Hanson and Lorien Pratt. Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems*, 1, 1988.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pages 97–117. Springer, 2017.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[20] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. *Advances in neural information processing systems*, 32, 2019.

[21] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[23] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586. PMLR, 2018.

[24] Vedant Nanda, Samuel Dooley, Sahil Singla, Soheil Feizi, and John P Dickerson. Fairness through robustness: Investigating robustness disparity in deep learning. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 466–477, 2021.

[25] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[26] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

[27] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems*, 33:21945–21957, 2020.

[28] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? *arXiv preprint arXiv:2104.09425*, 2021.

[29] D. Steinkraus, I. Buck, and P.Y. Simard. Using gpus for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1115–1120 Vol. 2, 2005.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[31] V Tjeng, K Xiao, and R Tedrake. Evaluating robustness of neural networks with mixed integer programming. preprint. *arXiv preprint arXiv:1711.07356*, 2019.

[32] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

[33] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[34] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

[35] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[36] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[37] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018.