*Supplementary for:*

# 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

## 1. Certification performance on different noise strengths (varying $\sigma$)

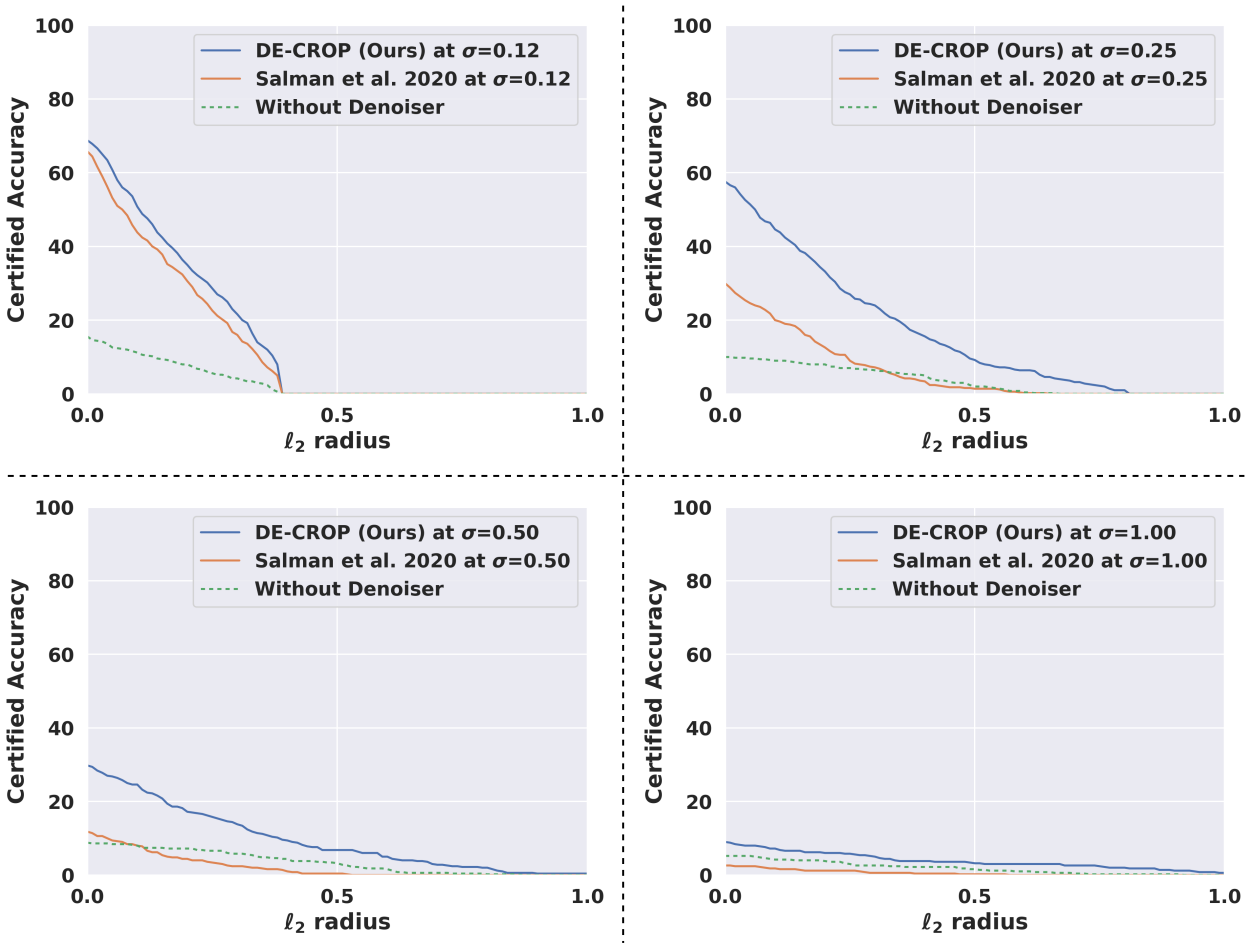


Figure 1. Our method (DE-CROP) consistently provides better certified robustness across different noise strengths with $\sigma \in \{0.12, 0.25, 0.50, 1.00\}$. The results are presented on limited training data $D_{train}^{lim}$ which is $1\%$ of entire training samples $D_{train}$ of CIFAR-10.

*denotes equal contribution.

## 2. Certified guarantees on different quantity of limited training data

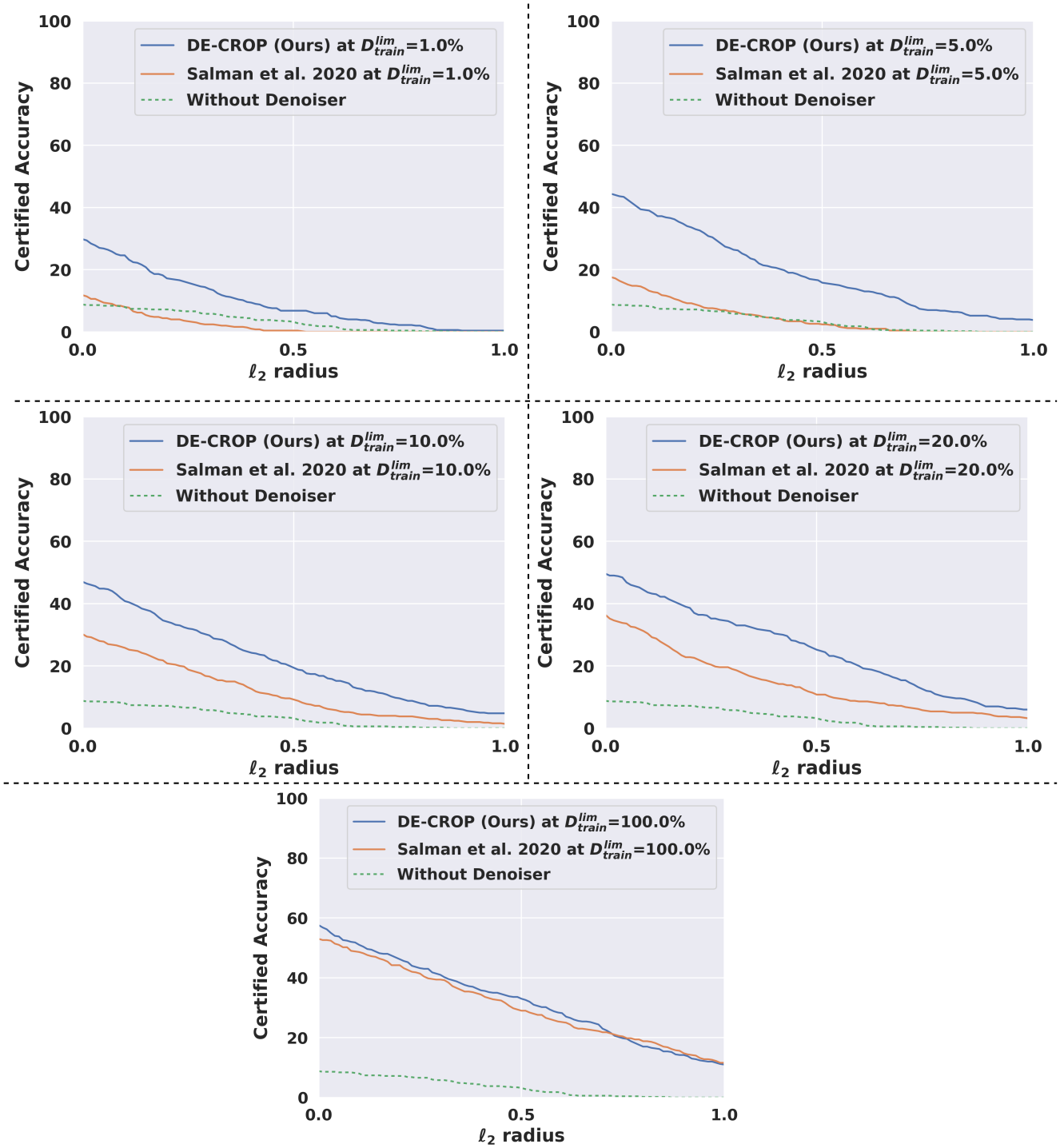


Figure 2. The amount of limited training data ($|D_{train}^{lim}|$) is $k\%$ of $D_{train}$. We perform ablation for different choices of $k$ (i.e. 1, 5, 10, 20 and 100) and report our certified performance. DE-CROP significantly boosts the certified accuracy across the radii when the available training set size is small, showing the efficacy of our method for limited data settings. The benefit is limited when the amount of training samples is high. The experiments are conducted on CIFAR-10 dataset with noise strength $\sigma = 0.50$

## 3. Sensitivity Analysis of mixing coefficient ($\alpha$)

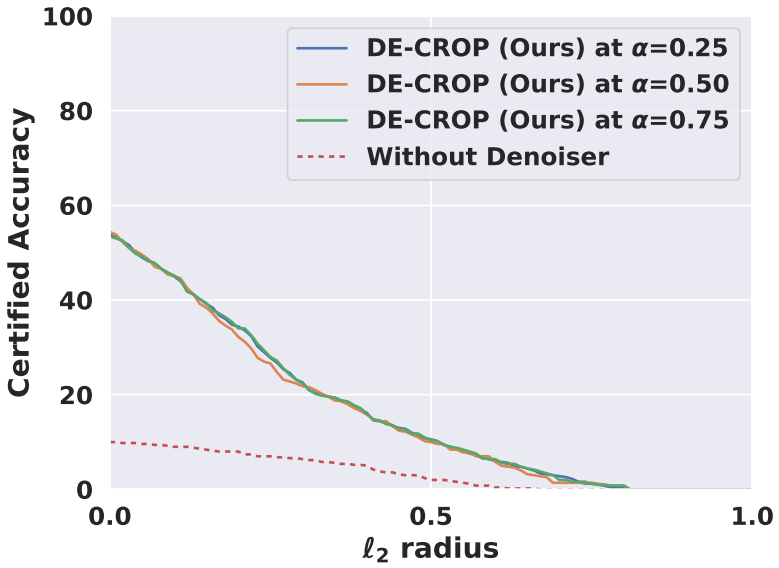


Figure 3. Ablation to determine the sensitivity of our proposed method (DE-CROP) to change in value of mixing coefficient ($\alpha$). We perform experiments with three different values of $\alpha$ (i.e. $0.25$, $0.50$, $0.75$) and observe that the certified performance of DE-CROP is stable across the broad range of $\alpha$ values allowing our technique to be easily adoptable without significant hyper-parameter tuning.

## 4. Additional experimental details

For all our experiments, we use three 1080Ti 12Gb cards. The additional details of specific components of our proposed approach (DE-CROP) are provided below:

**Training of base classifier ($B_c$)**: The base pretrained classifier $B_c$ is trained for 300 epochs with cross-entropy loss and stochastic gradient descent optimizer. The initial learning rate is $0.1$, which reduces by a factor of $10$ every $100$th epoch.

**Training of denoiser network ($D_n$)**: We use the architecture proposed by [4] for our denoiser ($D_n$). The network $D_n$ is a 17 layers deep fully convolutional network that contains multiple blocks of convolutional and batch-norm layers followed by ReLU activation. We train the denoiser network $D_n$ for 600 epochs, adopting the same learning routine as Salman *et al*. [2]. The Adam optimizer is used with a learning rate of $0.001$ for initial 50 epochs, followed by stochastic gradient descent optimizer with a learning rate of $0.001$ that reduces by a factor of $10$ every $200$th epoch.

**Training of domain discriminator ($D_d$)**: We use a vanilla multi-layer perceptron architecture with one hidden layer of size $100$ units as the architecture for domain discriminator ($D_d$). While training the $D_d$, we initialize the $D_n$ by pre-training it using $L_{lc}$, $L_{cs}$ and $L_{mmd}$ losses. Then optimize the parameters of $D_d$ ($\phi$) using Adam optimizer with a learning rate of 1e-5. Moreover, in order to ignore the noisy gradients from $D_d$ during the initial phase of training the domain discriminator we use a scheduler (similar to [1]) to weight that value of negative gradient by $\beta$ that is gradually increased over the course of training. We use the standard parameters for certification described by Salman *et al*. [2].

**Details on augmentations**: In the main draft (Table 2 of Sec. $5.1$), we showed experimental results to demonstrate the effect of various augmentation policies in improving certification performance. In policy 1, we apply the different transformations - 'randomized cropping' followed by 'random horizontal flipping with probability of $0.5$. For policy 2 we randomly select one of the following augmentations: 'random horizontal flip (probability=$0.5$)', 'random vertical flip (with probability=$0.5$)', 'randomized cropping', 'random affine transform to translate (with parameters = ($0.25$, $0.50$) and rotate (degrees=$10$) the image. Whereas for policy 3 all the augmentations of policy 2 are applied in a sequential manner making it a stronger augmentation via composition of different transformations that are applied on a given sample.

## 5. Visualizations of Original and Generated Samples

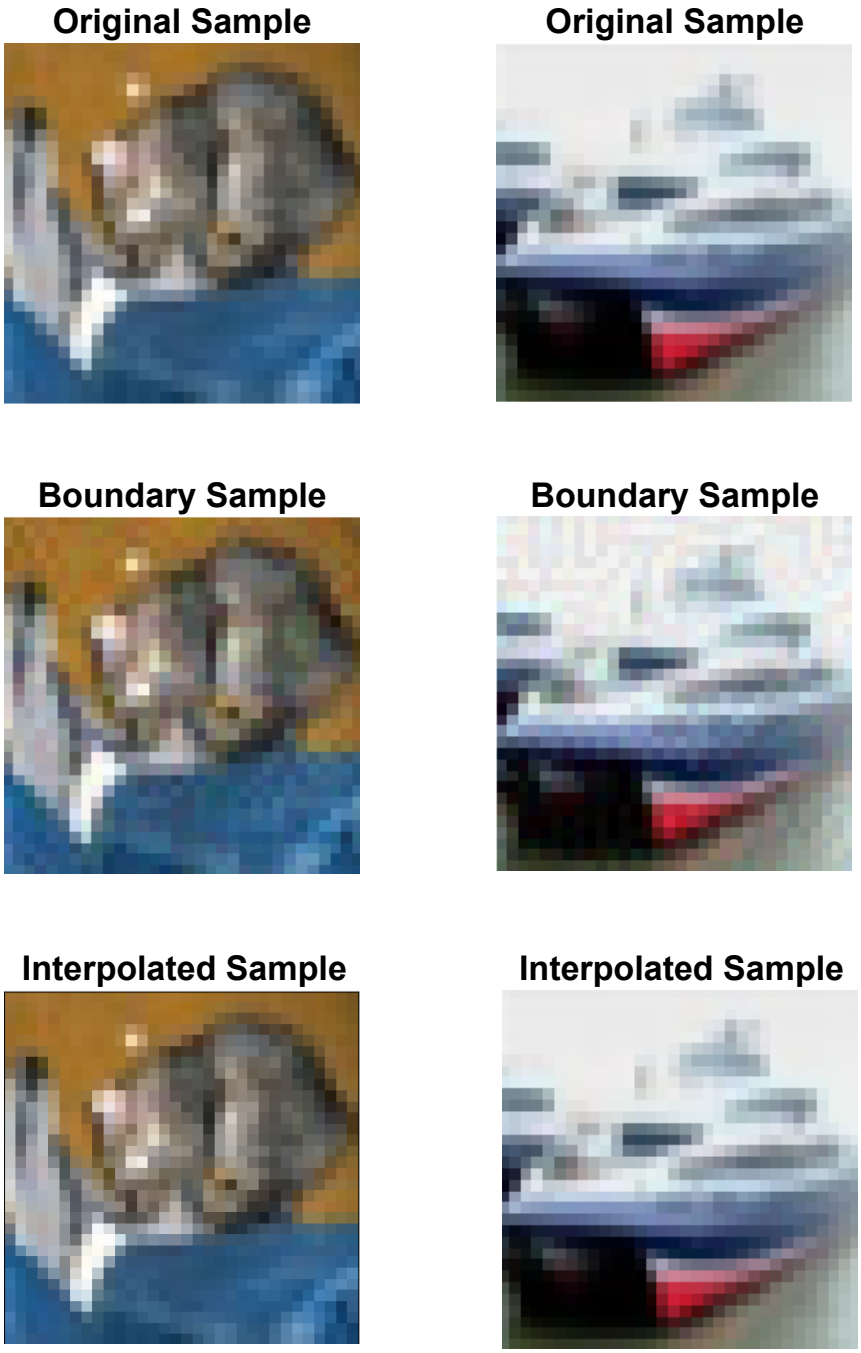


Figure 4. Visualization of the original, generated boundary and generated interpolated samples for CIFAR-10 dataset. The boundary samples are generated by perturbing the original sample with adversarial noise (refer eq. 5 in the main draft) and interpolated samples are generated by minimizing the Mean Squared Error with interpolation of original and boundary samples logits as the ground truth (refer eq. 7 in the main draft). Our simple and intuitive sample generation technique avoids complicated and time consuming methods like generative adversarial networks. Moreover, the generated samples ensure preservation of class semantics while providing diversity in the feature space of pre-trained classifiers (ref Fig. 2 in the main draft)

## 6. Certifying pretrained models of different architecture using limited training data

We adopted the denoiser network proposed by [4] i.e. DnCNN for all our ablations in the main draft. In order to demonstrate the adaptability and effectiveness of our proposed approach (DE-CROP) across diffrent choices of denoiser architecture, we compare our performance against Salman *et al*. [2] on a different denoiser architecture i.e. MemNet [3]. Similar to the results on DnCNN in the main draft (refer Fig. 4 in the main draft), we significantly outperform Salman *et al*. [2] in a limited data setting ($D^{lim}_{train}$ = 1% of $D_{train}$) across all the radii on CIFAR-10 with noise strength $\sigma$ = 0.25 (refer Fig. 5).

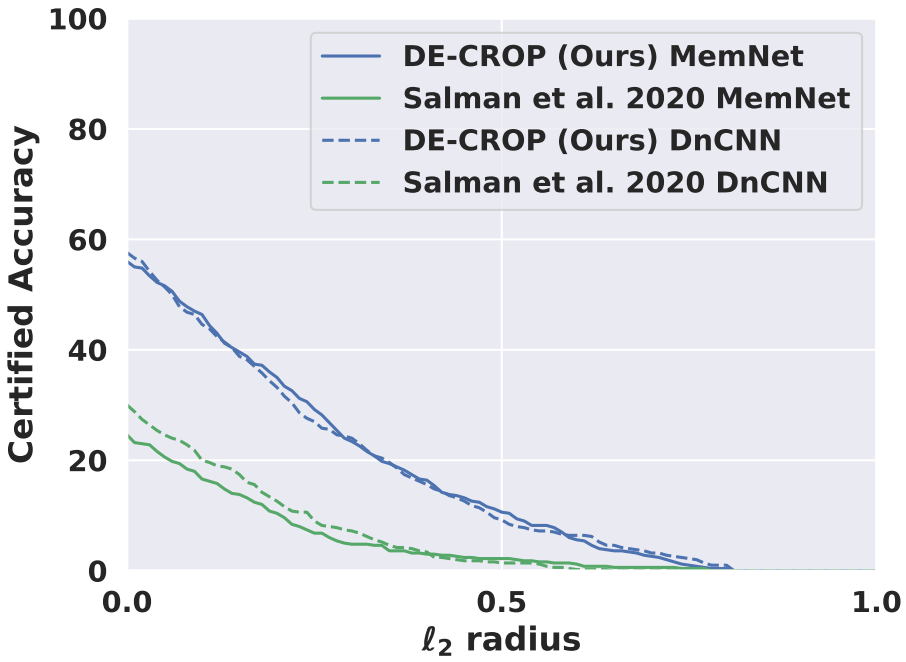


Figure 5. Performance comparison of our approach (DE-CROP) against Salman *et al*. [2] across multiple choices of denoiser architectures i.e. DnCNN and MemNet. DE-CROP achieves significantly better results on both DnCNN and MemNet without any hyperparameter tuning demonstrating its practical usefulness. The results are presented on the CIFAR-10 dataset with noise strength $\sigma$ = 0.25 and training set size equal to 1% of the entire trainset.

## 7. Effect of increasing interpolated samples

As described in Sec. 4.1 of the main draft, we generate one boundary sample ($x^i_b$) and one interpolated sample ($x^i_{int}$) for each data sample ($x^i_o$) belonging to our available limited training set ($D^{lim}_{train}$). In this section, we aim to explore whether increasing the amount of interpolated samples generated per sample leads to proportional benefit in the overall certification performance. In Fig. 6 we observe that increasing the amount of interpolated samples generated (per sample) from 1 (also reported in the main draft) to 2 and 5 does not increase the performance. Thus, we generate 1 interpolated sample only as it provides similar performance with cheaper computational overhead. Note, our performance with any of the three choices discussed (1, 2 or 5 interpolated samples) comfortably outperforms other state-of-the-art techniques.

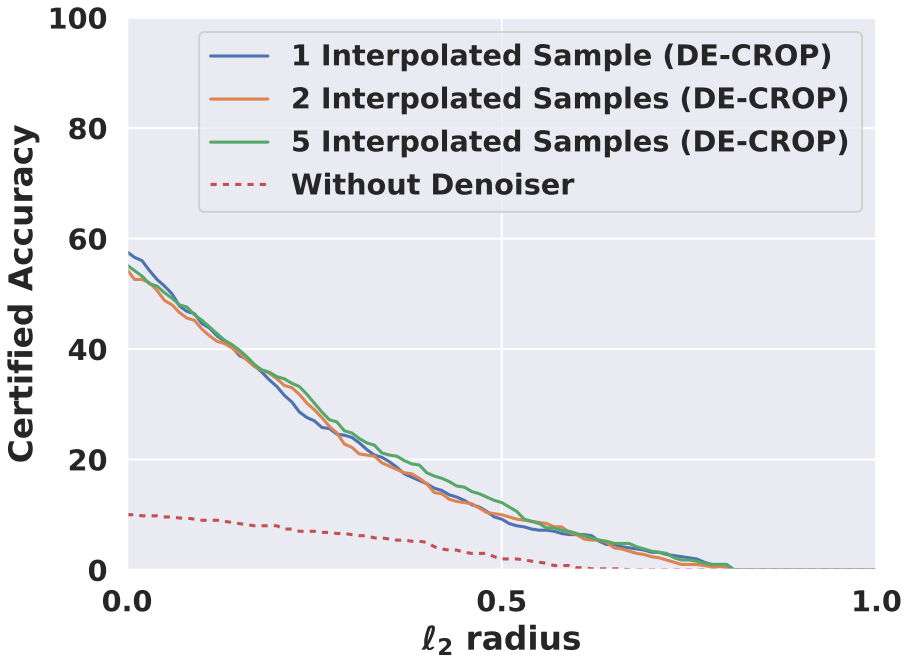


Figure 6. Effect of increasing the number of interpolated samples generated per sample on certification performance for CIFAR-10 dataset with noise strength $\sigma = 0.25$ and available limited training set = $1\%$ of entire training data. We observe that the improvement in performance is not proportional to the number of interpolated samples generated. Hence, we generate only 1 interpolated samples in our proposed approach DE-CROP.

## 8. Performance of DE-CROP when boundary samples crafted via different adversarial attacks

| **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 |
| DE-CROP (PGD) | 57.60 | 27.20 | 9.20 | 2.20 |
| DE-CROP (Auto Attack) | 53.20 | 29.20 | 11.20 | 1.20 |
| DE-CROP (DeepFool) | 54.40 | 26.40 | 10.20 | 1.80 |

Table 1. Investigating the effect of the choice of adversarial attack (for crafting boundary samples) on certified performance using our proposed approach DE-CROP. We significantly outperform the baseline across the different choices of adversarial attacks (i.e., PGD, DeepFool, and Auto Attack). We observe that although all three of them perform similarly, PGD obtains better certified standard accuracy compared to Auto Attack and DeepFool. Thus, we use PGD in the main draft.

## 9. Importance of boundary samples in DE-CROP

| **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 |
| DE-CROP without Boundary Samples | 52.46 | 20.26 | 5.80 | 0.40 |
| DE-CROP with Boundary samples | **57.60** | **27.20** | **9.20** | **2.20** |

Table 2. Demonstrating the utility of crafting boundary samples for our proposed approach: DE-CROP. In absence of boundary samples, interpolated logits are generated by using a pair of random samples from the same class (average results over three runs are reported). In contrast, we observe that the certified performance significantly improves when we utilize adversarial attack to generate boundary samples and use them along with the original samples to obtain interpolated logits. Thus, clearly highlighting the importance of crafting boundary samples in our method (DE-CROP) for certified defense in limited data.

## References


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

[2] 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.

[3] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[4] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.