PROVABLE DEFENSE BY DENOISED SMOOTHING WITH LEARNED SCORE FUNCTION

Kyungmin Lee
Agency for Defense Development
kyungmnlee@gmail.com

ABSTRACT

While randomized smoothing is an efficient method that presents certified robustness, it requires multiple classifiers for each noise type and scale. On the other hand, the denoised smoothing circumvents the multiple training of classifiers by deploying an image denoiser at the front of the classifier. Yet denoised smoothing also requires training multiple denoisers for each noise type and scales, we introduce a unified denoiser that can be applied for various noise types and scales, with one neural network estimator. Our idea is built on a score-based generative model which estimates the score function of data distribution. We show that training only one multi-scale score estimator can enhance the performance of denoised smoothing, and can be applied to various ℓ_p norm adversaries which were not available before. We validate our methods through experiments on ImageNet and CIFAR-10, under various ℓ_p adversaries.

1 INTRODUCTION

The deep neural network base image classifiers are susceptible to deliberate noises as known as *adversarial attacks* (Szegedy et al., 2013; Goodfellow et al., 2014; Carlini & Wagner, 2017). Even though many works proposed heuristics that can annul or mitigate adversarial attacks, most of them were broken by stronger attacks (Athalye et al., 2018; Athalye & Carlini, 2018). The vulnerability of empirical defenses had led the researchers to scrutinize on *certified defenses*, which ensure the models to have constant output within the allowed set around given input. Unfortunately, many provable defenses are not feasible to large-scale neural networks because of their constraints on the architecture.

The randomized smoothing, on the other hand, is a practical method that does not restrain the choice of neural networks. The randomized smoothing converts any base classifier to a smoothed classifier by making predictions over randomly perturbed samples. Then the smoothed classifiers are guaranteed to have a ℓ_p certified radius, which is theoretically derived by the noise type used for smoothing. Since Cohen et al. (2019) derived tight ℓ_2 certified radius for Gaussian randomized smoothing, sequential works studied the certification bounds for various distributions (Teng et al., 2020; Yang et al., 2020).

As the smoothed classifier makes the base classifier to predict on noisy samples, many works suggested methods to train the base classifier to deal with noisy inputs, which results in redundant classifiers trained with different noise types and scales. On the other hand, the denoised smoothing is a method that doesn't require auxiliary training of classifiers. It prepends image denoiser to the pre-trained classifier so that the noisy input is recovered before feeding into the classifier. However, those approach also requires multiple image denoiser for each noise type and scale. In this work, we develop denoised smoothing by introducing score-based image denoising. We exploit multi-scale denoising score matching (Song & Ermon, 2019) for score estimation, and propose an efficient simulated annealing algorithm for image denoising. Remark that we only require one score network to certify various noise distributions and levels. We provide experiments on ImageNet and CIFAR-10 to show the efficacy of our methods. Specifically, our methods perform better than original denoised smoothing, while can be applied to various noise types without any re-training. Furthermore, we compare with the random-ensemble based method, which we refer to *white-box smoothing*, and show that our method works are comparable to them.

2 BACKGROUNDS ON RANDOMIZED SMOOTHING AND DENOISED SMOOTHING

2.1 RANDOMIZED SMOOTHING

Let $f : \mathbb{R}^d \to \mathcal{Y}$ be a classifier and q be a distribution on $\mathbb{R}^d$. Then the *randomized smoothing* with q is a method that converts the base classifier f to the *associated smoothed classifier* g, where $g(\mathbf{x})$ returns the class which is most likely to be predicted by the base classifier f when $\mathbf{x}$ is perturbed by a random noise sampled from q, i.e.,

$$g(\mathbf{x}) = \arg\max_{c \in \mathcal{Y}} \Pr_{\mathbf{u} \sim q(\mathbf{u})} \big[f(\mathbf{x} + \mathbf{u}) = c \big]. \tag{1}$$

Robustness guarantee for smoothed classifiers Suppose an adversary can perturb the input $\mathbf{x}$ inside the allowed set $\mathcal{B}$, which is usually an ℓ_p ball centered at $\mathbf{x}$. For the case when $\mathcal{B}$ is ℓ_2 ball and q is Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$, $g(\mathbf{x})$ is robust within the radius

$$R = \frac{\sigma}{2} \Big(\Phi^{-1}(p_1) - \Phi^{-1}(p_2) \Big) \tag{2}$$

where Φ is inverse cumulative distribution function, and $p_1 = \max_c \Pr[f(\mathbf{x} + \mathbf{u}) = c]$ and $p_2 = \max_{c \neq g(\mathbf{x})} \Pr[f(\mathbf{x} + \mathbf{u}) = c]$. The derivation for Gaussian distribution was first introduced by Cohen et al. (2019), and has been generalized to various distributions (Teng et al., 2020; Yang et al., 2020).

2.2 DENOISED SMOOTHING

While randomized smoothing can theoretically achieve certified robustness, it requires the base classifier to predict over perturbed samples. Many works proposed training of classifiers by noisy data augmentation (Cohen et al., 2019), adversarial training (Salman et al., 2019), or regularizations (Zhai et al., 2019). In contrast to training the classifier for randomized smoothing, Salman et al. (2020) proposed *denoised smoothing* which prepends image denoiser to the pre-trained classifier. By training denoiser $\mathcal{D}_\theta : \mathbb{R}^d \to \mathbb{R}^d$, the smoothed classifier converted from $f \circ \mathcal{D}_\theta$ outperforms 'no-denoiser' baseline. They proposed training denoisers with mean squared error (MSE) loss or classification (CLF) loss, or combining both methods. The CLF loss makes the denoiser constrained to the classifier, so has limited application.

3 SCORE-BASED IMAGE DENOISING

3.1 IMAGE DENOISING WITH SCORE FUNCTION

The image denoising is an example of linear inverse problem, which can be formulated as following: given an observation $\mathbf{y} = \mathbf{x} + \mathbf{u}$ with $\mathbf{u} \sim q(\mathbf{u})$ finds $\hat{\mathbf{x}}(\mathbf{y})$ that is close to original $\mathbf{x}$. Let $\mathbf{x} \sim p(\mathbf{x})$ then the distribution of $\mathbf{y}$ is $p_q(\mathbf{y}) = \int p(\mathbf{y}, \mathbf{x}) d\mathbf{x} = \int p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \int q(\mathbf{y} - \mathbf{x}) p(\mathbf{x}) d\mathbf{x} = (p * q)(\mathbf{y})$.

One-step denoiser Suppose q is a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$ and let the distribution of $\mathbf{y}$ by p_{σ^2}. Let us define the score function of density $p(\mathbf{x})$ by $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, then the optimal denoiser can be obtained by estimating the score of p_{σ^2} (see Appendix for proof). Let $\mathbf{s}_\theta(\cdot; \sigma)$ be score network that estimates score of smoothed density p_{σ^2}. Then the denoiser from $\mathbf{s}_\theta$ is given by

$$\hat{\mathbf{x}}(\mathbf{y}) = \mathbf{y} + \sigma^2 \mathbf{s}_\theta(\mathbf{y}; \sigma). \tag{3}$$

Multi-step denoiser Otherwise, one can consider the maximum a posteriori (MAP) estimator that maximizes the conditional distribution $p(\mathbf{x}|\mathbf{y})$. Formally the MAP loss is given by,

$$\arg\min_{\mathbf{x}} L_{\text{MAP}}(\mathbf{x}; \mathbf{y}) = \arg\min_{\mathbf{x}} - \log p(\mathbf{x}) + \gamma \phi(\mathbf{y} - \mathbf{x}), \tag{4}$$

where ϕ is logarithm of q and γ is a hyperparameter (see Appendix for derivation). We aim to approximate the gradient of L_{MAP} by the score of Gaussian smooth densities. Let the approximate MAP loss with $\tilde{\sigma}$ by

$$L_{\text{MAP},\tilde{\sigma}}(\mathbf{x}; \mathbf{y}) = - \log p_{\tilde{\sigma}^2}(\mathbf{x}) + \phi(\mathbf{y} - \mathbf{x}). \tag{5}$$

ℓ_2 radius (CIFAR-10)	0.25	0.50	0.75	1.00	1.25	1.50
white-box smoothing (Cohen et al., 2019)	59	45	31	21	18	13
denoised smoothing (Query Access) (Salman et al., 2020)	45	20	15	13	11	10
denoised smoothing (Full Access) (Salman et al., 2020)	56	41	28	19	16	13
denoised smoothing (Our method)	60	42	28	19	11	6

Table 1: Certified accuracy of ResNet-110 on CIFAR-10 at various ℓ_2 radii.

ℓ_2 radius (ImageNet)	0.25	0.50	0.75	1.00	1.25	1.50
white-box smoothing (Cohen et al., 2019)	62	52	45	39	34	29
denoised smoothing (Query Access)(Salman et al., 2020)	48	31	19	12	7	4
denoised smoothing (Full Access)(Salman et al., 2020)	50	33	20	14	11	6
denoised smoothing (Our method)	56	41	30	24	17	11

Table 2: Certified accuracy of ResNet-50 on ImageNet at various ℓ_2 radii.

Then we can approximate the gradient of $L_{\text{MAP},\tilde{\sigma}}(\mathbf{x};\mathbf{y})$ by score network and perform gradient descent initialized with $\mathbf{x}_0 = \mathbf{y}$ as following:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \nabla_{\mathbf{x}_t} L_{\text{MAP},\tilde{\sigma}}(\mathbf{x};\mathbf{y}) \approx \mathbf{x}_t + \alpha(\mathbf{s}_\theta(\mathbf{x}_t;\tilde{\sigma}) + \nabla_{\mathbf{x}_t}\phi(\mathbf{y}-\mathbf{x}_t)). \tag{6}$$

Remark that the proposed method can be applied to any log-concave noise distributions. In appendix, we present theorem that shows the recovery guarantee when q is a Gaussian distribution.

3.2 Multiscale Denoising Score Matching and Simulated Annealing

Recently, score network trained by multi-scale denoising score matching objective has shown to be effective for generative modeling (Song & Ermon, 2019). Multi-scale denoising score matching is weighted sum of denoising score matching objectives with various noise magnitudes. Given a sequence of noise levels $\{\sigma_i\}_{i=1}^L$, which is the variance of centered Gaussian distribution, the total loss function with σ_i^2 is given as following:

$$\mathcal{L}(\theta;\{\sigma_i\}_{i=1}^L) = \frac{1}{L}\sum_{i=1}^{L} \frac{\sigma_i^2}{2} \mathbb{E}_{\mathbf{x}\sim p,\mathbf{y}\sim\mathcal{N}(\mathbf{x},\sigma_i^2 I)}\left[\left\|\mathbf{s}_\theta(\mathbf{y};\sigma_i) + \frac{\mathbf{y}-\mathbf{x}}{\sigma_i^2}\right\|_2^2\right]. \tag{7}$$

Given mutli-scale score estimator, we conduct multistep denoising with simulated annealing, which runs update with noise magnitudes that anneals to the smallest. The simulated annealing allows faster optimization, which is suitable for denoised smoothing. The detailed algorithm and empirical analyses on how multi-scale denoising score matching objective helps denoised smoothing are presented in the appendix.

4 Experiments

We study the performance of score based denoised smoothing on ImageNet (Deng et al., 2009) and CIFAR-10 Krizhevsky et al. (2009). We measured the certified accuracy at R, which is the fraction of test set for which the smoothed classifier correctly predicts and certifies robust at an ℓ_p radius bigger than R. Detailed experimental settings can be found in appendix.

First, we experimented the performance of one-step denoiser for Gaussian randomized smoothing. We compare with 1) *white-box smoothing*, which is canonical approach that trains base classifiers with Gaussian data augmentation (Cohen et al., 2019), and 2) the denoised smoothing with denoisers trained by Salman et al. (2020). For all experiments on denoised smoothing, we used same ResNet110 classifier for CIFAR-10 and pytorch pretrained ResNet50 classifier for ImageNet. The results are in Table 1 and Table 2. We found out that for CIFAR-10, our method achieves better performance than original denoised smoothing, while is on par with white-box smoothing. Also, for

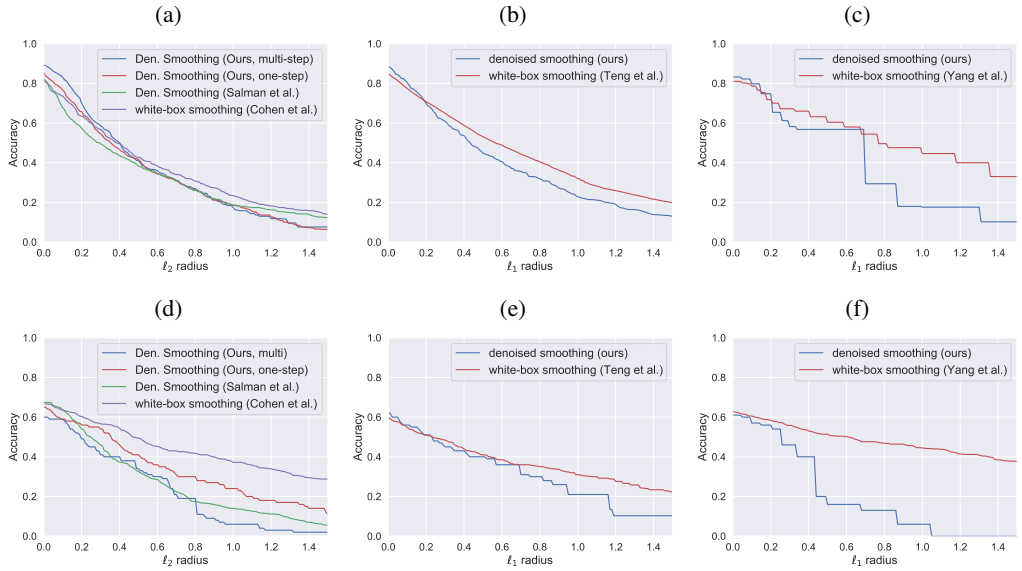

Figure 1: The performance of multi-step denoiser for denoised smoothing. The blue lines are our methods, and red lines are white-box smoothing which are experimented by each authors. (a) ℓ_2 certified accuracy with Gaussian smoothing on CIFAR-10, (b) ℓ_1 certified accuracy with Laplace smoothing on CIFAR-10, (c) ℓ_1 certified accuracy with uniform smoothing on CIFAR-10, (d) ℓ_2 certified accuracy with Gaussian smoothing on ImageNet, (e) ℓ_1 certified accuracy with Laplace smoothing on ImageNet, (f) ℓ_1 certified accuracy with uniform smoothingon ImageNet,.

ImageNet, our method achieves better performance than original denoised smoothing, yet is slightly below than white-box smoothing.

Next, we demonstrate the effectiveness of our multi-step denoiser on denoised smoothing using various noise types. For a baseline, we compare with white-box smoothing which is training classifiers with noisy data augmentation. We experimented on Gaussian noise (Cohen et al., 2019), Laplace noise (Teng et al., 2020), and uniform noise (Yang et al., 2020) for both CIFAR-10 and ImageNet. For all experiments, we used ResNet110 classifiers for CIFAR-10 and ResNet50 classifiers for ImageNet. See Appendix for more details. It is important to claim that all experiments are done with the only **one** score-network for each CIFAR-10 and ImageNet.

5 Conclusion

In this work, we presented a score-based denoised smoothing which exploits score estimation neural network for certified defense of any classifier that can be used regardless of noise types and scales. We empirically found out that our method performs better than original denoised smoothing, while comparable to randomized smoothing with noisy trained base classifiers.

We believe that current randomized smoothing is theoretically well-designed but needs to be scalable to be deployed for real world applications. On that perspective, our approach is a good initial point that can endow robustness to any classifier without any re-training. However, the hardness of estimating score function of high-dimensional data should be compromised. We believe using better architecture or devising faster optimization algorithm might help.

References

Muhammad Asim, Ali Ahmed, and Paul Hand. Invertible generative models for inverse problems: mitigating representation error and dataset bias, 2020. URL `https://openreview.net/forum?id=BJgkbyHKDS`.

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

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pp. 274–283, 2018.

N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.

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

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014.

Durk P Kingma and Yann L. Cun. Regularized estimation of image statistics by score matching. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems 23*, pp. 1126–1134. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/4060-regularized-estimation-of-image-statistics-by-score-matching.pdf.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Mengyin Lu and Matthew Stephens. Empirical bayes estimation of normal means, accounting for uncertainty in estimated standard errors, 2019.

Herbert Robbins. An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pp. 157–163, Berkeley, Calif., 1956. University of California Press. URL https://projecteuclid.org/euclid.bsmsp/1200501653.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pp. 11292–11303, 2019.

Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J. Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers, 2020.

Saeed Saremi and Aapo Hyvarinen. Neural empirical bayes, 2020.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11918–11930, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pp. 574–584. PMLR, 2020.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

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.

Jiaye Teng, Guang-He Lee, and Yang Yuan. ℓ_1 adversarial robustness certificates: a randomized smoothing approach, 2020. URL https://openreview.net/forum?id=H1lQIgrFDS.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Jay Whang, Qi Lei, and Alexandros G Dimakis. Compressed sensing with invertible generative models and dependent noise. *arXiv preprint arXiv:2003.08089*, 2020.

Greg Yang, Tony Duan, Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. *arXiv preprint arXiv:2002.08118*, 2020.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*, 2019.

A DETAILED EXPLANATIONS

A.1 FORMULATION OF IMAGE DENOISING PROBLEM

The image denoising is an example of linear inverse problem, which can be formulated as following: given an observation $\mathbf{y} = \mathbf{x} + \mathbf{u}$ with $\mathbf{u} \sim q(\mathbf{u})$ finds $\hat{\mathbf{x}}(\mathbf{y})$ that is close to original $\mathbf{x}$. Let $\mathbf{x} \sim p(\mathbf{x})$ then the distribution of $\mathbf{y}$ is $p_q(\mathbf{y}) = \int p(\mathbf{y}, \mathbf{x}) d\mathbf{x} = \int p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \int q(\mathbf{y} - \mathbf{x}) p(\mathbf{x}) d\mathbf{x} = (p * q)(\mathbf{y})$.

One-step denoiser Suppose q is a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$ and let the distribution of $\mathbf{y}$ by p_{σ^2}. Then the following proposition (Robbins, 1956; Lu & Stephens, 2019; Saremi & Hyvarinen, 2020) reveals the relationship between the optimal denoiser $\mathcal{D}_{\theta^*}$ and p_{σ^2}.

Proposition A.1. *Assume $\theta* \in \arg\min_\theta L_{MSE}(\theta)$, then the following equation holds:*

$$\mathcal{D}_{\theta^*}(\mathbf{y}) = \mathbf{y} + \sigma^2 \nabla_{\mathbf{y}} \log p_{\sigma^2}(\mathbf{y}) \tag{8}$$

The proof of proposition A.1 is in Appendix B. Let us define the score function of density $p(\mathbf{x})$ by $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, then the optimal DAE can be obtained by estimating the score of p_{σ^2}. Let $\mathbf{s}_\theta(\cdot; \sigma)$ be score network that estimates score of smoothed density p_{σ^2}. Then the denoiser from $\mathbf{s}_\theta$ is given by

$$\hat{\mathbf{x}}(\mathbf{y}) = \mathbf{y} + \sigma^2 \mathbf{s}_\theta(\mathbf{y}; \sigma). \tag{9}$$

Remark that it is only valid when q is Gaussian distribution.

Multi-step denoiser Consider the maximum a posteriori (MAP) estimator that maximizes the conditional distribution $p(\mathbf{x}|\mathbf{y})$. Formally the MAP loss is given by,

$$\begin{aligned}
\arg\min_{\mathbf{x}} L_{\text{MAP}}(\mathbf{x}; \mathbf{y}) &= \arg\min_{\mathbf{x}} -\log p(\mathbf{x}|\mathbf{y}) && (10)\\
&= \arg\min_{\mathbf{x}} -\log p(\mathbf{x}) - \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{y}) && (11)\\
&= \arg\min_{\mathbf{x}} -\log p(\mathbf{x}) - \log q(\mathbf{y} - \mathbf{x}) && (12)\\
&= \arg\min_{\mathbf{x}} -\log p(\mathbf{x}) + \phi(\mathbf{y} - \mathbf{x}). && (13)
\end{aligned}$$

Note that we simply remove density term $p(\mathbf{y})$ and rewrite with q. Lastly, we rewrite q with ϕ. Since the density $p(\mathbf{x})$ is usually intractable for high-dimensional dataset, one may use approximation to make the MAP loss tractable. Many recent works focused on using cutting edge generative models such as generative adversarial network (GAN) or invertible neural networks to approximate $p(\mathbf{x})$ in equation 12 (Ulyanov et al., 2018; Whang et al., 2020; Asim et al., 2020). However, GAN suffer from mode collapse, and invertible neural networks require extremely long steps to reach local minima, which are not sufficient for randomized smoothing.

Instead, we aim to approximate the gradient of L_{MAP} by the score of Gaussian smooth densities. Let the approximate MAP loss with $\tilde{\sigma}$ by

$$L_{\text{MAP},\tilde{\sigma}}(\mathbf{x}; \mathbf{y}) = -\log p_{\tilde{\sigma}^2}(\mathbf{x}) + \phi(\mathbf{y} - \mathbf{x}). \tag{14}$$

Then we can approximate the gradient of $L_{\text{MAP},\tilde{\sigma}}(\mathbf{x}; \mathbf{y})$ by score network and perform gradient descent initialized with $\mathbf{x}_0 = \mathbf{y}$ as following:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \nabla_{\mathbf{x}_t} L_{\text{MAP},\tilde{\sigma}}(\mathbf{x}; \mathbf{y}) \approx \mathbf{x}_t + \alpha(\mathbf{s}_\theta(\mathbf{x}_t; \tilde{\sigma}) + \nabla_{\mathbf{x}_t} \phi(\mathbf{y} - \mathbf{x}_t)). \tag{15}$$

Remark that the proposed method can be applied to any log-concave noise distributions. Following theorem shows the recovery guarantee of our methods when q is a Gaussian distribution.

Theorem A.2. *Let $\mathbf{x}^*$ be local optimum of $p(\mathbf{x})$, and $\mathbf{y} = \mathbf{x}^* + \mathbf{u}$ where $\mathbf{u} \sim \mathcal{N}(0, \sigma^2 I)$. Assume $-\log p$ is μ-strongly convex within the neighborhood $\mathcal{B}_r(\mathbf{x}) = \{\mathbf{z} : \|\mathbf{z} - \mathbf{x}\| \leq r\}$. Then, the gradient descent method on approximate loss $L_{MAP,\tilde{\sigma}^2}(\mathbf{x}; \mathbf{y})$ initialized by $\mathbf{x}_0 = \mathbf{y}$ converges to its local minima $\hat{\mathbf{x}}(\mathbf{y}; \tilde{\sigma}) \in \arg\min L_{MAP,\tilde{\sigma}^2}(\mathbf{x}; \mathbf{y})$ that satisfies:*

$$\mathbb{E}\|\hat{\mathbf{x}}(\mathbf{y}; \tilde{\sigma}) - \mathbf{x}^*\|_2 \leq \frac{\sigma\sqrt{d}(1 + \mu\tilde{\sigma}^2)}{1 + \mu\tilde{\sigma}^2 + \mu\sigma^2} + \tilde{\sigma}\sqrt{d} \tag{16}$$

The proof of theorem A.2 is in Appendix B. Remark that the upper bound in equation 16 increases as σ increases, which shows that the recovery becomes harder as σ becomes larger. Also the upper bound is strictly increasing function of $\tilde{\sigma}$, and has the minimum when $\tilde{\sigma} = 0$.

A.2 EFFICIENT IMAGE DENOISING WITH SIMULATED ANNEALING

From theorem 3.2, for small $\tilde{\sigma}$ the error bound is tight but the approximation is inaccurate at nascent steps. Otherwise, when $\tilde{\sigma}$ is large, the error bound is too large. To arbiter the tradeoff, and to make the method scalable, we propose simulated annealing for score-based image denoising. Let $\{\sigma_i\}_{i=1}^L$ be a decreasing sequence of noise levels, then simulated annealing runs T steps of approximate gradient descent for each σ_i. The algorithm for simulated annealing for image denoising is in Algorithm 1.

Algorithm 1 Simulated Annealing for denoising

Require: $\mathbf{y}, \{\sigma_i\}_{i=1}^L, \alpha, T$
1: initialize $\mathbf{x}_0 = \mathbf{y}$
2: **for** $i \leftarrow 1 : L$ **do**
3: $\quad \alpha_i \leftarrow \alpha \cdot \sigma_i^2/\tilde{\sigma}^2$
4: $\quad$ **for** $t \leftarrow 1 : T$ **do**
5: $\quad\quad \mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \alpha_i\big(\mathbf{s}_{\theta,\sigma_i}(\mathbf{x}_t) + \nabla_{\mathbf{x}_t}\phi(\mathbf{x}_t - \mathbf{y})\big)$
6: $\quad$ **end for**
7: $\quad \mathbf{x}_0 \leftarrow \mathbf{x}_T$
8: **end for**
9: **return** $\mathbf{x}_T$

Note that Song & Ermon (2019; 2020) used annealed Langevin dynamics for generative modeling. Our approach is similar to them, but we consider the image denoising problem instead. Also, note that Kingma & Cun (2010) trained score network for image denoising, but they used primitive neural networks where exact score-matching was possible.

A.3 SCORE ESTIMATION VIA SCORE MATCHING

Score estimation has been studied through various topics such as generative modeling (Song et al., 2020; Song & Ermon, 2019) and reinforcement learning (Sutton et al., 2000). Score matching is a method that trains a score network $\mathbf{s}_\theta(\mathbf{x})$ to estimate score. The original score matching objective is given by

$$\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\text{tr}(\nabla_{\mathbf{x}}\mathbf{s}_\theta(\mathbf{x})) + \frac{1}{2}\|\mathbf{s}_\theta(\mathbf{x})\|_2^2\right]. \tag{17}$$

However, due to heavy computation of $\text{tr}(\nabla\mathbf{s}_\theta(\mathbf{x}))$, and since we are only interested in score of smoothed densities, we use different approach.

Denoising Score Matching Denoising score matching is a method that learns the score of smooth densities. More concretely, the score network $\mathbf{s}_\theta$ estimates the score of density $p_{\sigma^2}(\mathbf{y}) = \int \mathcal{N}(\mathbf{x}, \sigma^2 I)p(\mathbf{x})d\mathbf{x}$. The objective was proved to be equivalent to the following (Vincent, 2011):

$$\mathbb{E}_{\mathbf{y}\sim q_{\sigma^2}(\mathbf{y}|\mathbf{x}),\mathbf{x}\sim p(\mathbf{x})}[\|\mathbf{s}_\theta(\mathbf{y};\sigma) - \nabla_y \log q_{\sigma^2}(\mathbf{y}|\mathbf{x})\|_2^2]. \tag{18}$$

Remark that the optimal score network satisfies $\mathbf{s}_{\theta^*}(\mathbf{x};\sigma) = \nabla \log p_{\sigma^2}(\mathbf{x})$ for each σ, and as $\sigma \to 0$, $\mathbf{s}_{\theta^*,\sigma}(\mathbf{x}) \to \nabla \log p(\mathbf{x})$.

Multi-Scale Denoising Score Matching Recently, training score network with multi-scale denoising score matching has been proposed (Song & Ermon, 2019). Multi-scale denoising score matching trains one score network with various noise magnitudes. Given a sequence of noise levels $\{\sigma_i\}_{i=1}^L$, which is the variance of centered Gaussian distribution, by rewriting the denoising score matching objective for each σ_i, we have

$$\mathcal{L}(\theta;\sigma_i) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p,\mathbf{y}\sim\mathcal{N}(\mathbf{x},\sigma_i^2 I)}\left[\left\|\mathbf{s}_\theta(\mathbf{y};\sigma_i) + \frac{\mathbf{y}-\mathbf{x}}{\sigma_i^2}\right\|_2^2\right]. \tag{19}$$

Then the total loss is

$$\mathcal{L}(\theta;\{\sigma_i\}_{i=1}^L) = \frac{1}{L}\sum_{i=1}^L \sigma_i^2\mathcal{L}(\theta;\sigma_i), \tag{20}$$

note that each loss is weighted by σ_i which allows the loss of each noise level has the same order of magnitude. It is worth to notify that our method is unsupervised, and classifier-free.

Here we demonstrate some advantages of multi-scale denoising score matching. First, through learning various noise magnitudes at once, it suffices to train only one neural network to apply image denoising. Therefore, we can do randomized smoothing regardless of the noise level. Second, the noise makes the support of the score function to be whole space, making score estimation more consistent. Moreover, a large amount of noise fills the low-density region, which helps to estimate the score of the non-Gaussian or off-the-manifold samples. Empirically, we found out that multi-scale learning helps the denoising performance. See Appendix C for details.

B THEORETICAL ANALYSIS

Proposition B.1. *Let θ^* be optimal, i.e. $\theta^* = \arg\min_\theta L_{MSE}$. Then it satisfies*

$$D_{\theta^*}(\tilde{x}) = \tilde{x} + \sigma^2 \nabla_{\tilde{x}} \log p(\tilde{x}), \tag{21}$$

where $\tilde{x} \sim \mathcal{N}(x, \sigma^2 I)$.

Proof. Assume data density $p(x)$ be differentiable, then the optimal denoisier, i.e.

$$D^* \in \arg\min_D \mathbb{E}_{x\sim p(x), \tilde{x}\sim p(\tilde{x}|x)}[\|D(\tilde{x}) - x\|_2^2]$$

is given by

$$D^*(\tilde{x}) = \mathbb{E}_{x\sim p(x|\tilde{x})}[x]. \tag{22}$$

First note that the smooth density $p_{\sigma^2}(x)$ is given by

$$p_{\sigma^2}(\tilde{x}) = \int p(x, \tilde{x}) dx \tag{23}$$

$$= \int p(\tilde{x}|x) p(x) dx \tag{24}$$

where $p(\tilde{x}|x) = \mathcal{N}(x, \sigma^2 I)$. Then the gradient of smooth density is

$$\nabla p_{\sigma^2}(\tilde{x}) = \int \nabla p(\tilde{x}|x) p(x) dx \tag{25}$$

$$= \int \frac{(x - \tilde{x})}{\sigma^2} p(\tilde{x}|x) p(x) dx \tag{26}$$

$$= \frac{1}{\sigma^2} \int (x - \tilde{x}) p(x|\tilde{x}) p_{\sigma^2}(\tilde{x}) dx \tag{27}$$

$$= \frac{p_{\sigma^2}(\tilde{x})}{\sigma^2} \left(\int x p(x|\tilde{x}) dx - \tilde{x} \int p(x|\tilde{x}) dx \right) \tag{28}$$

$$= \frac{p_{\sigma^2}(\tilde{x})}{\sigma^2} \big(\mathbb{E}_{p(x|\tilde{x})}[x] - \tilde{x} \big) \tag{29}$$

$$= \frac{p_{\sigma^2}(\tilde{x})}{\sigma^2} \big(D^*(\tilde{x}) - \tilde{x} \big) \tag{30}$$

which results in

$$D(\tilde{x}) = \tilde{x} + \frac{\sigma^2}{p_{\sigma^2}(\tilde{x})} \nabla p_{\sigma^2}(\tilde{x}) = \tilde{x} + \sigma^2 \nabla \log p_{\sigma^2}(\tilde{x}) \tag{31}$$

□

Before we prove theorem 3.2, we introduce several lemmas.

Definition B.1. (Strong convexity). A function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if, for all $\mathbf{x}_1, \mathbf{x}_2$, the following inequality holds for some $\mu > 0$:

$$f(\mathbf{x}_2) \geq f(\mathbf{x}_1) + \nabla f(\mathbf{x}_1)^\top (\mathbf{x}_2 - \mathbf{x}_1) + \frac{\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|^2 \tag{32}$$

Definition B.2. (Strong log-concavity). A distribution $p : \mathbb{R}^d \to [0, 1]$ is Σ-strongly log-concave if, p of the form

$$p(\mathbf{x}) = g(\mathbf{x})\mathcal{N}(0, \Sigma) \tag{33}$$

for some log-concave function g and a positive definite matrix Σ. If $\Sigma = \sigma^2 I$, p is σ^2-strongly log-concave shortly.

Following lemma shows the relationship between strong log-concavity and strong convexity.

Lemma B.2. *Assume p be σ^2-strongly log-concave, then $p \propto \exp(-f)$ for some $\frac{1}{\sigma^2}$-strongly convex f.*

The proof can be found in . Next lemma states the preservation of strong log-concavity under convolution.

Lemma B.3. *If p_1 is σ_1^2-strongly log-concave, and p_2 is σ_2^2-strongly concave, then the distribution $p_1 * p_2$ is $(\sigma_1^2 + \sigma_2^2)$-strongly log-concave.*

The proof can be found in . Finally, we have following lemma for the bounds on Wasserstein distance between p and its smoothed density p_{σ^2}.

Lemma B.4. *Let p be any distribution and p_{σ^2} be smoothed density obtained by $p_{\sigma^2} = p * \mathcal{N}(0, \sigma^2 I)$, then the 2-Wasserstein distance between p and p_{σ^2} satisfies*

$$\mathcal{W}_2(p, p_{\sigma^2}) \leq \sigma\sqrt{d} \tag{34}$$

Now we're ready to proof our theorem.

Proof. Let $\tilde{\mathbf{x}}$ be local optimum of $p_{\tilde{\sigma}^2}$. By lemma A.4, as $-\log p$ is μ-strongly convex, p is $\frac{1}{\mu}$-strongly log-concave and as Gaussian distribution $\mathcal{N}(0, \tilde{\sigma}^2)$ is $\tilde{\sigma}^2$-strongly log-concave, $p_{\tilde{\sigma}^2}$ is $(\frac{1}{\mu} + \tilde{\sigma}^2)$-strongly log-concave, and equivalently $-\log p_{\tilde{\sigma}^2}$ is $\frac{\mu}{1+\mu\tilde{\sigma}^2}$-strongly convex. Then as $\hat{\mathbf{x}} \in \arg\min L_{\text{MAP},\tilde{\sigma}}$, we have

$$\nabla L_{\text{MAP},\tilde{\sigma}}(\hat{\mathbf{x}}) = 0 \iff -\nabla p_{\tilde{\sigma}^2}(\hat{\mathbf{x}}) + \frac{1}{\sigma^2}(\hat{\mathbf{x}} - \mathbf{y}) = 0 \tag{35}$$

$$\iff -\nabla p_{\tilde{\sigma}^2}(\hat{\mathbf{x}}) + \nabla p_{\tilde{\sigma}^2}(\tilde{\mathbf{x}}) = \frac{1}{\sigma^2}(\mathbf{y} - \hat{\mathbf{x}}) \tag{36}$$

$$\iff \langle -\nabla p_{\tilde{\sigma}^2}(\hat{\mathbf{x}}) + \nabla p_{\tilde{\sigma}^2}(\tilde{\mathbf{x}}), \hat{\mathbf{x}} - \tilde{\mathbf{x}}\rangle = \frac{1}{\sigma^2}\langle \mathbf{y} - \hat{\mathbf{x}}, \hat{\mathbf{x}} - \tilde{\mathbf{x}}\rangle \tag{37}$$

Then we have

$$\frac{1}{\sigma^2}\langle \mathbf{y} - \tilde{\mathbf{x}}, \hat{\mathbf{x}} - \tilde{\mathbf{x}}\rangle = \frac{1}{\sigma^2}\langle (\mathbf{y} - \hat{\mathbf{x}}) + (\hat{\mathbf{x}} - \tilde{\mathbf{x}}), \hat{\mathbf{x}} - \tilde{\mathbf{x}}\rangle \tag{38}$$

$$= \frac{1}{\sigma^2}\langle \mathbf{y} - \hat{\mathbf{x}}, \hat{\mathbf{x}} - \tilde{\mathbf{x}}\rangle + \frac{1}{\sigma^2}\langle \hat{\mathbf{x}} - \tilde{\mathbf{x}}, \hat{\mathbf{x}} - \tilde{\mathbf{x}}\rangle \tag{39}$$

$$= \langle -\nabla p_{\tilde{\sigma}^2}(\hat{\mathbf{x}}) + \nabla p_{\tilde{\sigma}^2}(\tilde{\mathbf{x}}), \hat{\mathbf{x}} - \tilde{\mathbf{x}}\rangle + \frac{1}{\sigma^2}\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|_2^2 \tag{40}$$

$$\geq \frac{\mu}{1 + \mu\tilde{\sigma}^2}\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|_2^2 + \frac{1}{\sigma^2}\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|_2^2 \tag{41}$$

$$= \frac{1 + \mu\tilde{\sigma}^2 + \mu\sigma^2}{\sigma^2(1 + \mu\tilde{\sigma}^2)}\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|_2^2 \tag{42}$$

Then by Cauchy-Schwarz inequality, we have

$$\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|_2 \leq \frac{1 + \mu\tilde{\sigma}^2}{1 + \mu\tilde{\sigma}^2 + \mu\sigma^2}\|\hat{\mathbf{x}} - \mathbf{y}\|_2 = \frac{1 + \mu\tilde{\sigma}^2}{1 + \mu\tilde{\sigma}^2 + \mu\sigma^2}\|\mathbf{u}\|_2 \tag{43}$$

Finally, by lemma A.5,

$$\mathbb{E}\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \mathbb{E}\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|_2 + \mathbb{E}\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \tag{44}$$

$$\leq \frac{1 + \mu\tilde{\sigma}^2}{1 + \mu\tilde{\sigma}^2 + \mu\sigma^2}\mathbb{E}_{\mathbf{u}\sim\mathcal{N}(0,\sigma^2 I)}[\|\mathbf{u}\|_2] + \mathcal{W}_2(p, p_{\tilde{\sigma}^2}) \tag{45}$$

$$= \frac{\sigma\sqrt{d}(1 + \mu\tilde{\sigma}^2)}{1 + \mu\tilde{\sigma}^2 + \mu\sigma^2} + \tilde{\sigma}\sqrt{d} \tag{46}$$

□

C EXPERIMENT DETAILS

C.1 TRAINING SCORE NETWORKS

We used NCSNv2 from Song & Ermon (2020). For CIFAR-10 we used original NCSNv2, and for ImageNet we used the deepest version of NCSNv2. Note that the author first released NCSN (Song & Ermon, 2019), then proposed improved version (Song & Ermon, 2020). NCSN and NCSN v2 are based on RefineNet, and some major changes in normalization, pooling layer, and convolution layer lead to successful score-based modeling. The original NCSN was developed for generative modeling, and choosing noise level is crucial for generative modeling. Even though we are doing image denoising, choosing noise level also seems important. We experimented with two types of noise sequences: uniform sequence and geometric sequence. We used uniform noise sequence for one-step denoiser. We set $\sigma_1 = 1.0$ and $\sigma_L = 0.05$ with $L = 20$. We used geometric sequence for multi-step denoiser. For geometric noise sequences, we set $\sigma_1 = 1.0$, and $\sigma_L = 0.01$ with $L = 32$. Note that combining both sequences doesn't change the overall results.

For all experiments, we trained with Adam optimizer with learning rate 1e-5, and ran 300,000 iterations. We will soon release the code for details.

C.2 MULTI-STEP DENOISERS

For each Gaussian and uniform distribution, we ran annealed gradient descent with learning rate $\alpha = 2e-5$, and for laplace distribution we ran with learning rate $\alpha = 3e-5$. For each noise levels, we ran with $T = 1$ for fast denoising.

C.3 TRAINING CLASSIFIERS

For pretrained classifiers, we used CIFAR-10 classifiers publicly released from Salman et al. (2020), and pytorch pretrained ResNet50 for ImageNet. Also, for white-box smoothing baseline, we used Gaussian randomized smoothing baseline from Cohen et al. (2019) and ImageNet uniform and laplace ResNet50 baseline from Yang et al. (2020). Otherwise, we trained ResNet110 with laplace and uniform noise data augmentation on CIFAR-10 to reproduce the results. For training, we tested with $\sigma = \{0.15, 0.25, 0.50, 1.00\}$, with the training hyperparameter same as Cohen et al. (2019).

C.4 CERTIFICATION

We use the CERTIFY of randomized smoothing (Cohen et al., 2019) to do our experiments. We conducted all experiments with $n = 10,000$, $n0 = 100$ and $\alpha = 0.001$. Note that if we certify with larger n all results can be improved, however we stick with $n = 10,000$ due to computational constraints.

D ADDITIONAL EXPERIMENTS

D.1 HOW MULTI-SCALE METHODS HELP

In this section, we show how training with multi-scale DSM differs from training with each noise levels. To comparse, we trained score networks with one noise level each, and otherwise we trained with multi-scale DSM. We trained with noise levels $\sigma = 0.12, 0.25, 0.50, 1.00$, and plot certified accuracy for denoised smoothing with one-step denoiser from each score networks (Figure). that the multi-scale DSM achieves better performance, which is explained in section 2.

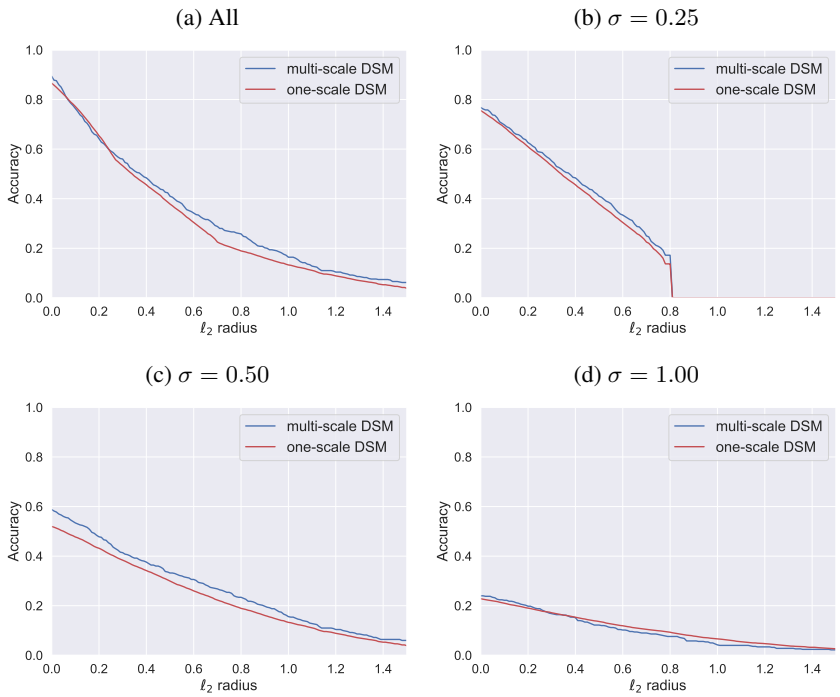

Figure 2: Denoised smoothing with multi-scale DSM helps.

D.2 CERTIFICATION WITH OTHER CLASSIFIERS

Here we show that using stronger classifier, i.e. the classifier with high test accuracy, achieves better performance. We used 4 pretrained classifiers ResNet110, ResNet18, WideResNet40-10, WideResNet28-10 from Salman et al. (2020), where each classifier is trained with 300 epochs. We found out that using stronger classifier achieves better certified accuracy, and it is because we aren't fitting the denoiser into specific classifier (See Figure 3).

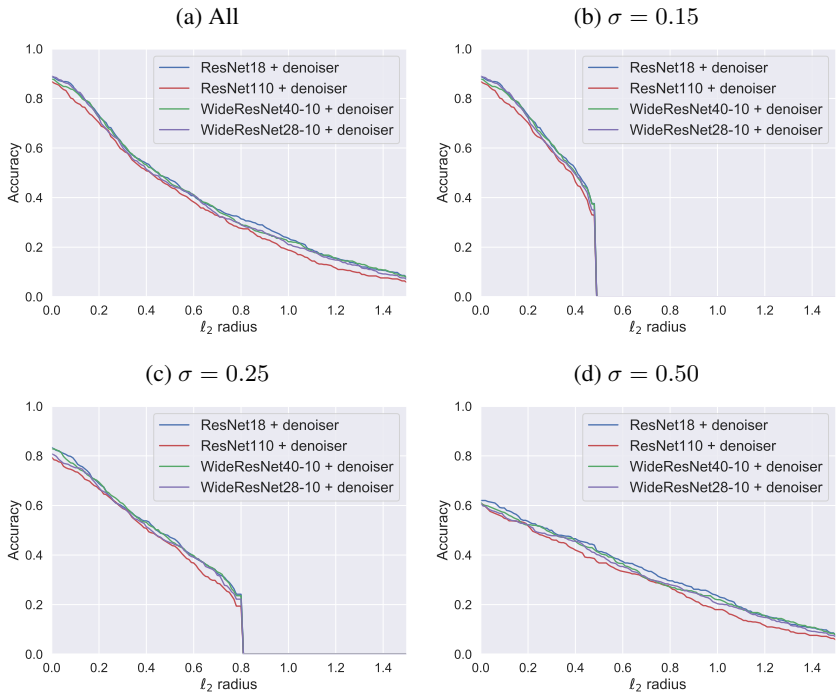

Figure 3: Robust accuracy under various ℓ_p radius with various classifiers.