Supplementary Materials for NCIS: Neural Contextual Iterative Smoothing for Purifying Adversarial Perturbations

Sungmin Cha[1], Naeun Ko[3], Heewoong Choi[2], Youngjoon Yoo[3, 4], and Taesup Moon [2]
[1] New York University [2] ASRI / INMC / Seoul National University [3] NAVER Cloud [4] NAVER AI Lab
sungmin.cha@nyu.edu, naeun.ko@navercorp.com, chw0501@snu.ac.kr,
youngjoon.yoo@navercorp.com, tsmoon@snu.ac.kr

1. Experimental Settings

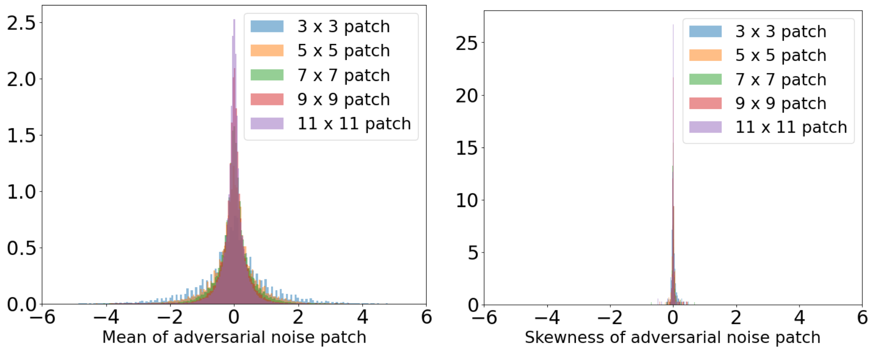
Table 1. Details on experimental settings.

	Access to purifier		Access to classifier	Other name
	weights	aware	weights	
Full white box attack	✓	✓	✓	White-box attack
Pre-processor-aware white-box attack	✗	✓	✓	[19]: Purifier-aware attack
Pre-processor blind white-box attack	✗	✗	✓	[9]: Gray-box attack
Black-box attack	✗	✗	✗	Generally known as black-box attack

To clarify the attack settings we focus on, we summarized them in Table 1. Note that our main target is pre-processor blind white-box attack (equivalent to NRP [15]) and pre-processor-aware white-box attack (shown in Supplementary Materials), not the *full* white-box attack which we believe is unrealistic for practical use, as in the API experiment of the manuscript.

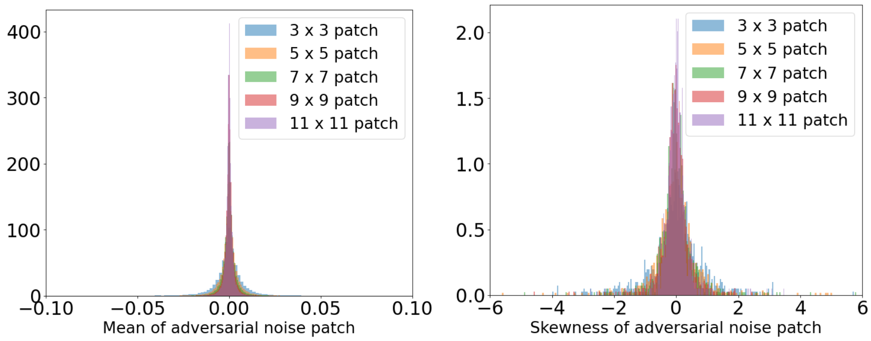
2. Additional Analysis for Adversarial Noise

Figure 1, 2, and 3 show additional experimental results of adversarial noise. We conducted the experiments on the same dataset used in Analysis on Adversarial Noise section of the manuscript, but with different attack methods. Figure 1 is the result of targeted L_∞ PGD [14] attack with the same ϵ, α and attack iterations proposed in the manuscript. Figure 2 shows analysis of untargeted L_2 PGD attack with $\epsilon = \{1, 2, 3, 4, 5\}$ and Figure 3 is about L_2 CW [5] attack with 10 attack iterations. From all the experimental results, we observe that the patches of adversarial noises generated from untargeted/targeted and L_2/L_∞ optimization-based adversarial attacks consistently show more or less zero mean and have symmetric distribution. We believe that these results support how our proposed methods could achieve such strong purification results against various types of attacks, as shown in Table 1 of the manuscript.



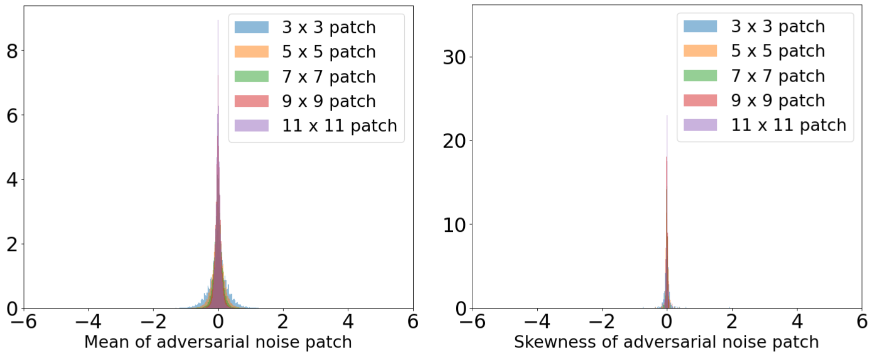
(a) Distribution of mean (b) Skewness

Figure 1. Experimental analysis for adversarial examples generated from targeted L_∞ PGD [14] (α = 1.6 / 255, where α is a step size) attack with 10 attack iterations.



(a) Distribution of mean (b) Skewness

Figure 2. Experimental analysis for adversarial examples generated from targeted L_2 PGD [14] (α = 1 / 255, where α is a step size) attack with 10 attack iterations.



(a) Distribution of mean (b) Skewness

Figure 3. Experimental analysis for adversarial examples generated from targeted L_2 CW [5] attack with 10 attack iterations. All other hyperparameters are set to the default hyperparameters of [12].

3. Architectural Details and Experimental Results for NCIS

Architectural details on FBI-Net We implemented FBI-Net by slightly modifying FBI-Denoiser [3]'s official code. In the original paper, they composed FBI-Net with 17 layers, 64 convolutional filters for each layer. For our method, we changed the number of layers to 8 for all experiments including FBI-Net, FBI-E and NCIS.

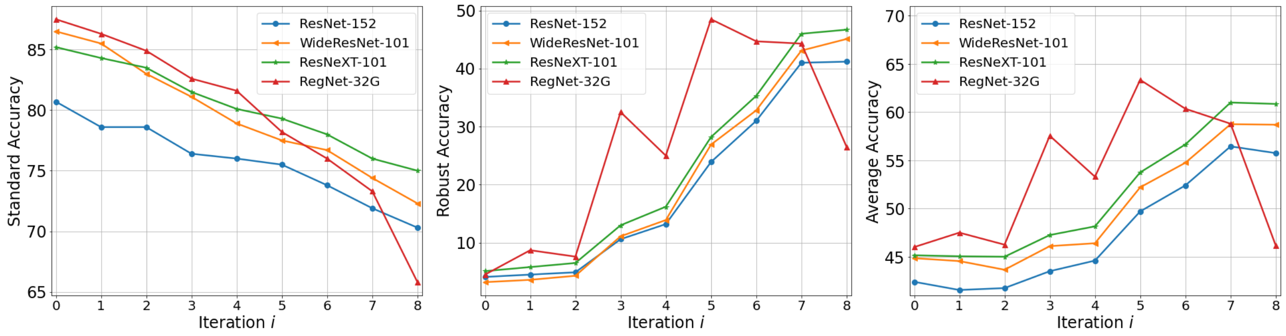
The number of training data for training NCIS For the self-supervised training of NCIS, we randomly selected and used only 5% images of the ImageNet training dataset since there was no significant difference even when more images were used, as shown in Table 2.

Table 2. Experimental results of NCIS ($i = 7$, $K = 11$, $m = 2$) trained by different number of images. For all experiments, we used ResNet-152 as the classification model and evaluate each case with the ImageNet validation dataset. For generating adversarial examples, we attacked each image using untargeted L_∞ PGD ($\epsilon = 16/255, \alpha = 1.6/255$) attack with 10 attack iterations. We only experimented with a single seed.

ResNet-152	Standard Accuracy	Robust Accuracy
NCIS (5%)	**69.07**	**46.49**
NCIS (10%)	68.92	49.14
NCIS (15%)	68.84	46.93
NCIS (20%)	68.93	47.84
NCIS (30%)	68.99	46.84

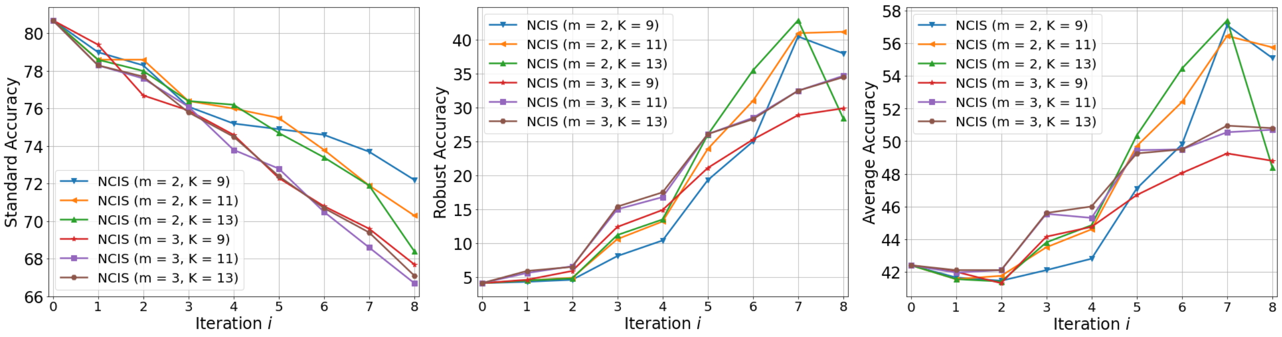
Selecting the number of iterations i for NCIS ($K = 11, m = 2$) Also, we conducted experiments, as in Figure 4, to select i (number of iterations for iterative smoothing) of NCIS for each classification model. Considering average of standard and robust accuracy, $i = 7$ is the best iteration number for ResNet-152 [11], WideResNet-101 [23] and ResNeXT-101 [21] and $i = 5$ is best for RegNet-32G [16]. Note that, for all experiments, the selected i for each classification model was used fixedly.

Finding the best configuration of NCIS Figure 5 shows the experimental results of various types of NCIS. Note that all NCIS are trained with 5% images of ImageNet training dataset as already proposed in the previous section. First, both robust and standard accuracy of NCIS with $m = 3$ is clearly lower than NCIS with $m = 2$ because reconstruction difficulty increases as the reshape size of FBI-E becomes bigger. Second, among the results of NCIS with $m = 2$ at $i = 7$, NCIS ($m = 2$, $K = 13$) achieves slightly better performance compared to NCIS ($m = 2$, $K = 11$). However, robust accuracy of both NCIS ($m = 2$, $K = 13$) and NCIS ($m = 2$, $K = 9$) significantly decrease at $i = 8$ where NCIS ($m = 2$, $K = 11$) does not. In this regard, we are concerned that NCIS ($m = 2$, $K = 13$) and NCIS ($m = 2$, $K = 9$) might be sensitive to the number of iterations even though they are slightly ahead in performance. Therefore, we selected NCIS ($m = 2$, $K = 11$) as the representative model of NCIS and conducted all experiments with it.



(a) Standard accuracy (b) Robust accuracy (c) Average of standard and robust accuracy

Figure 4. Experimental results of selecting the number of iterations i of NCIS for each classification model. We used randomly sampled 1,000 images from ImageNet training dataset and adversarial examples generated by L_∞ PGD ($\epsilon = 16/255$, $\alpha = 1.6/255$) attack with 10 attack iterations.



(a) Standard accuracy

(b) Robust accuracy

(c) Average of standard and robust accuracy

Figure 5. Experimental results of variants of NCIS. Experiments are conducted with ImageNet pretrained ResNet-152. We randomly sampled 1,000 images from ImageNet training dataset and generated adversarial examples using L_∞ PGD ($\epsilon = 16/255$, $\alpha = 1.6/255$) attack with 10 attack iterations.

4. Additional Experimental Results

Ablation study To verify each proposed module, we conducted ablation study and the results are shown in Table 3. For experiments, we used the ImageNet validation dataset and adversarial examples of it generated by L_∞ PGD ($\epsilon = 16/255, \alpha = 1.6/255$) attack with 10 attack iterations. The first row shows the result of NCIS with complete components. The second row is excluding GS ($K = 11$) from NCIS, which is named as FBI-E ($m = 2$). It shows that both standard and robust accuracy slightly drop because it becomes difficult to reconstruct given images. The third and fourth row show the result of FBI + GS ($K = 11$) and FBI respectively. We clearly observe that not only inference time and GPU memory requirement significantly increase but also both standard and robust accuracy decrease after removing the extension operations. The fifth and sixth row is the result of both GS cases and, as already checked in previous experiments, GS ($K = 5$) achieves remarkable performance but GS ($K = 11$) alone doesn't.

Table 3. Experimental results for ablation studies.

GS ($K = 5$)	GS ($K = 11$)	FBI	Extension ($m = 2$)	Standard Accuracy	Robust Accuracy	Inference Time	GPU Memory
✗	✓	✓	✓	**69.07**	**48.06**	**0.0779**	**0.60G**
✗	✗	✓	✓	65.08	46.07	0.0669	0.60G
✗	✓	✓	✗	66.51	21.37	0.1743	2.13G
✗	✗	✓	✗	67.62	39.93	0.1636	2.13G
✓	✗	✗	✗	63.32	44.92	4×10^{-5}	0.002G
✗	✓	✗	✗	21.35	19.98	4×10^{-5}	0.002G

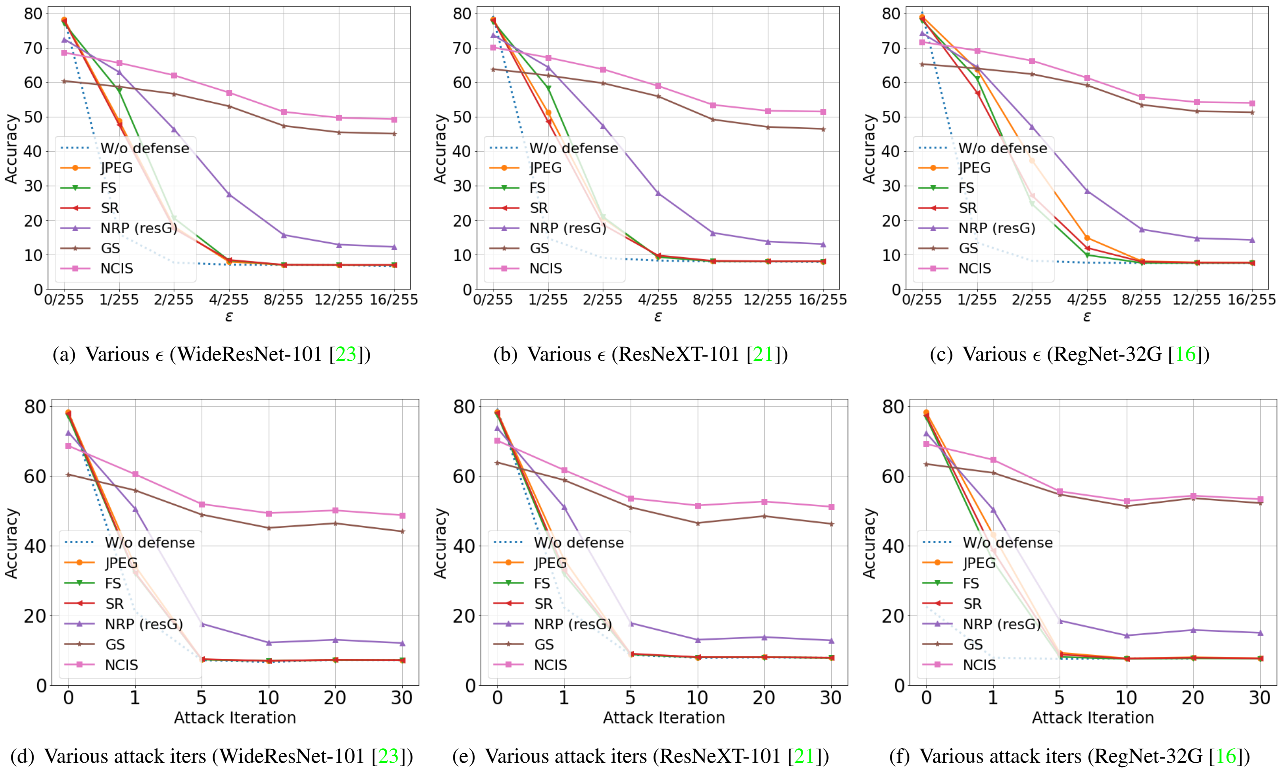
Comparison of inference time, GPU memory, and the number of parameters Table 4 shows the comparison of inference time, GPU memory requirement, and the number of parameters for purifying a single image. GPU memory was measured on an image of size 224x224. First, we can check that traditional input transformation-based methods consistently show fast inference time, except for TVM [17], with no GPU memory requirement. However, as already proposed in the manuscript and [10], these methods are easily broken by strong white-box attacks. Second, the original NRP has a large number of parameters and requires a huge GPU memory for purifying a single image. We think this is a fatal weakness from a practical point of view. To overcome this limitation, the author of NRP newly proposed a lightweight version of NRP, denoted as NRP (resG), at their official code. NRP (resG) significantly reduces inference time, GPU memory requirement, and the number of parameters. However, both NRP and NRP (resG) have the generalization issue and cannot purify several types of adversarial examples well, as proposed in the manuscript. In addition, NCIS is slower than NRP (resG), but shows faster inference time and memory requirements than NRP. Note that the number of parameters of NCIS is significantly lower than NRP variants. GS is the most computationally efficient compared to other baselines. Even though our NCIS is slow and has high computational cost than GS and NRP (resG), we would like to emphasize that NCIS generally achieved superior results against various attacks than GS, and also much better results than both NRP and NRP (resG) when considering both standard and robust accuracy, as already shown in the manuscript.

Table 4. Comparison of computational efficiency.

	JPEG	FS	TVM	SR	NRP (resG)	NRP	**GS** ($i = 7$)	**NCIS** ($i = 7$)
Inference Time (s)	0.0070	0.0007	1.1259	0.0084	0.0007	0.0892	**0.0004**	**0.0779**
GPU Memory	-	-	-	-	0.43G	9.86G	**0.002G**	**0.60G**
# of Parameters	-	-	-	-	1.70M	16.6M	**-**	**0.40M**

Pre-processor blind white-box PGD attack on other classification models Figure 6 shows additional experimental results against pre-processor blind white-box attacks on other classification models. We clearly observe that our proposed methods surpass other baselines on all classification models. Notably, we see that the performance gap between NCIS and GS is slightly wider than the gap on ResNet-152 which was shown in the manuscript. We believe these results show our methods generally purify overall adversarial examples generated from various types of the classification model.

Transfer-based black-box attack on other classification models Following the manuscript, Figure 7 shows the experimental results against transfer-based black-box attacks on other classification models. We set a substitute model for each classification model and generated adversarial examples by attacking it. The experimental results demonstrate that GS and NCIS achieve superior results than NRP (resG) in most cases. Also, the same result is shown once again that the performance gap between NCIS and GS is slightly wider than the gap on WideResNet-101. Note that transfer-based black-box attacks on RegNet-32G were not as successful as attacks on other classification models so that there is not much difference in robust accuracy between defense methods.



(a) Various ϵ (WideResNet-101 [23])

(b) Various ϵ (ResNeXT-101 [21])

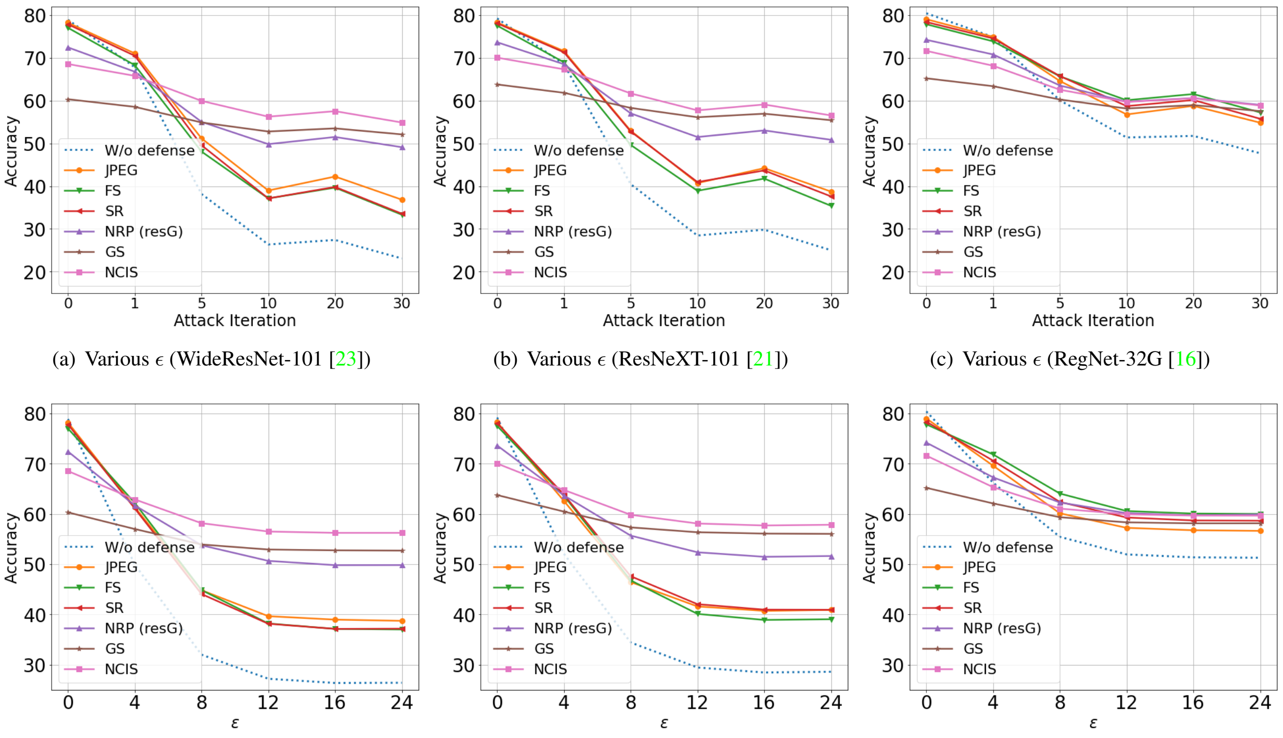
(c) Various ϵ (RegNet-32G [16])

(d) Various attack iters (WideResNet-101 [23])

(e) Various attack iters (ResNeXT-101 [21])

(f) Various attack iters (RegNet-32G [16])

Figure 6. Experimental results against L_∞ pre-processor blind white-box PGD attacks on WideResNet-101/ResNeXT-101/RegNet-32G. For the experiments on various ϵ, we set step size $\alpha = 1.6/255$ and attack iterations = 10. For various attack iterations experiments, we equally set $\epsilon = 16/255$, and $\alpha = 1.6/255$ if the number of attack iteration is lower than 10, and set $\alpha = 1/255$ otherwise.



(a) Various ϵ (WideResNet-101 [23])

(b) Various ϵ (ResNeXT-101 [21])

(c) Various ϵ (RegNet-32G [16])

(d) Various attack iters (WideResNet-101 [23])

(e) Various attack iters (ResNeXT-101 [21])

(f) Various attack iters (RegNet-32G [16])

Figure 7. Experimental results of transfer-based black-box attack with L_∞ PGD attack on WideResNet-101 (substitute model: ResNet-152)/ResNeXT-101 (substitute model: ResNet-152)/RegNet-32G (substitute model: WideResNet-101). For the experiments on various ϵ, we set step size $\alpha = 1.6/255$ and attack iterations = 10. In the case of experiments on attack iterations, we equally set $\epsilon = 16/255$, and $\alpha = 1.6/255$ if the number of attack iteration is lower than 10, and set $\alpha = 1/255$ otherwise.

Experimental results for RegNet-32G [16] Figure 5 presents the experimental results for RegNet-32G which is a recent state-of-the-art architecture for classification. Similar to the experimental results in the manuscript, we again observe that our NCIS achieve superior purification performance than other baselines against various types of adversarial attack.

Table 5. Experimental results of *untargeted* white-box adversarial attack. For L_∞ attacks, we set $\epsilon = 16/255$, $\alpha = 1.6/255$ and attack iterations = 10. For L_2 PGD attack, we used $\epsilon = 5$ and $\alpha = 0.1$. For L_2 CW attack, other than setting attack iterations as 10, we applied default hyperparameters proposed in [12]. **Boldface** denotes our proposed methods, and red and blue denotes the highest and second highest results respectively. We set $i = 6$ of GS and NCIS.

Model / Defense		Standard Accuracy	CW (L_2)	MIFGSM (L_∞)	DIFGSM (L_∞)	PGD (L_∞)	PGD (L_2)
RegNet-32G	W/o defense	80.43	9.19	7.77	0.39	7.46	11.88
	JPEG	79.04	52.56	7.82	0.38	7.65	47.83
	FS	77.86	51.62	7.77	0.40	7.49	47.85
	TVM	67.06	59.37	16.72	14.02	28.66	60.21
	SR	78.38	48.16	7.77	0.33	7.54	40.43
	NRP (resG)	74.22	58.48	20.55	4.35	14.20	58.18
	NRP	76.03	57.00	16.05	4.72	14.06	56.77
	GS ($i = 5$)	**65.22**	**62.71**	**30.83**	**28.04**	**51.24**	**63.01**
	NCIS ($i = 5$)	**71.50**	**67.37**	**41.84**	**31.13**	**53.97**	**67.28**

Experimental results against AutoAttack [6] We conducted the experiments against AutoAttack (implemented in [12]) with various purification methods including NCIS, following the settings of attacking ImageNet pretrained model proposed in [6]. The results in Table 6 show that our NCIS and GS again outperform other baselines in both L_∞ and L_2 attacks. We believe this result again shows a better generalizability of NCIS for purification against various types of attack.

Table 6. Experimental results for AutoAttack with ImageNet pretrained ResNet-152.

	W/o defense	JPEG	FS	SR	NRP (resG)	GS	**NCIS**
AutoAttack (L_∞, ϵ =2/255)	0.0	24.18	10.74	13.73	46.81	59.74	**63.10**
AutoAttack (L_∞, ϵ =4/255)	0.0	3.90	2.47	4.27	30.61	57.38	**59.46**
AutoAttack (L_∞, ϵ =8/255)	0.0	1.63	2.08	2.08	24.07	51.88	**52.98**
AutoAttack (L_2, ϵ =510/255)	0.0	11.47	8.74	10.65	40.68	59.75	**62.78**
AutoAttack (L_2, ϵ =765/255)	0.0	3.50	2.63	5.29	27.40	58.25	**60.36**

Experimental results of Denoised Smoothing (DS) [18] Denoised Smoothing (DS) [18] proposed an approach that applies a pre-trained Gaussian denoiser for adversarial robustness. Although there is similarity in the use of a denoising-based model, there are several critical differences between our method and DS. Firstly, while DS utilizes an existing Gaussian denoiser, we develop a learnable neural network-based smoothing function that is based on both BSN and novel findings for adversarial noise. Secondly, DS requires multiple samplings of Gaussian noise to certify a classifier and the certified accuracy is only valid for a small radius of L2 perturbations, whereas our method is more efficient and achieves strong robust accuracy against a wide range of adversarial attacks.

In order to evaluate the adversarial robustness of the Denoised Smoothing (DS) approach against PGD attack(L_∞, $\alpha = 1.6/255$, $\epsilon = 16/255$), we conducted experiments using the experimental settings outlined in Table 1 of our paper, specifically using the RegNet-152 architecture. We used the official code and pre-trained weights of the denoiser ($\sigma = 0.25$) provided by the authors of the DS method. To ensure fair comparison, we varied the number of random noise samplings, a key hyperparameter of the DS method, and present the results in Table 7. Our findings indicate that while a single iteration of noise sampling ($n = 1$) of the DS method yields improved robust accuracy, it also leads to a significant drop in standard accuracy. Additionally, increasing the number of noise samplings ($n > 1$) improves both robust and standard accuracy, however, the improvement is relatively small and the inference time increases significantly. In comparison to our NCIS method proposed Table 1 of the manuscript, these results suggest that the DS method may not be as effective against PGD attacks.

Score-based black-box attack Following the suggestion of [4], we evaluate both baselines and our proposed method against score-based black-box attack. We select Square [1], which is the state-of-the-art and query efficient black-box attack method,

Table 7. Experimental results for Denoised Smoothing ($\sigma = 0.25$) with ImageNet pretrained ResNet-152.

PGD (L_∞)	$n = 0$	$n = 1$	$n = 10$	$n = 100$	$n = 1000$
Standard Accuracy	78.25	14.52	15.38	15.35	15.41
Robust Accuracy	6.20	13.07	13.69	13.77	13.80
Inference Time (s)	-	0.16	1.6	16.2	162.7

for the experiment. For L_2 Square attack, we set $\epsilon = 5, p = 0.1$ with 300 queries and, for L_∞ Square attack, we set $\epsilon = 8/255, p = 0.1$ with 300 queries. Other hyperparameters are set to default values as [12].

Table 8 shows the experimental results against L_∞ and L_2 Square attacks. First, we observe that attack success rate of Square is still low compared to transfer-based black-box attacks and it requires more than 300 queries to successfully fool the classification model. Second, as already shown in [8], the traditional input transformation-based methods (*e.g.* JPEG, FS, TVM and SR) show robust performance compared to others, such as white-box and transfer-based black-box attacks. Third, our proposed methods (NCIS and GS) and NRP (resG) accomplish competitive performance, and NCIS consistently surpasses GS. When comparing NCIS with NRP (resG), we observe that NCIS obtain higher robust accuracy than NRP (resG) in Square (L_∞) but not in Square (L_2).

Table 8. Experimental results for Square attack with ImageNet pretrained ResNet-152.

Model / Defense		Square (L_∞)	Square (L_2)
ResNet-152	W/o defense	38.00	37.68
	JPEG	58.81	63.20
	FS	62.57	63.68
	TVM	51.23	58.05
	SR	58.99	60.04
	NRP (resG)	53.10	59.25
	GS	**55.31**	**51.57**
	NCIS	**58.52**	**55.95**

As an analysis, we visualized adversarial examples and noises generated by Square attack in Figure 8. We numerically checked that the mean of the adversarial noise of an entire image is almost zero. However, the figure clearly shows that adversarial noise generated by Square has a structured noise different from adversarial noise generated by the optimization-based white-box attack (*e.g* PGD). We believe that this type of adversarial noise is not a form considered in both our proposed method and NRP (resG) and that's why input transformation-based methods show better accuracy.

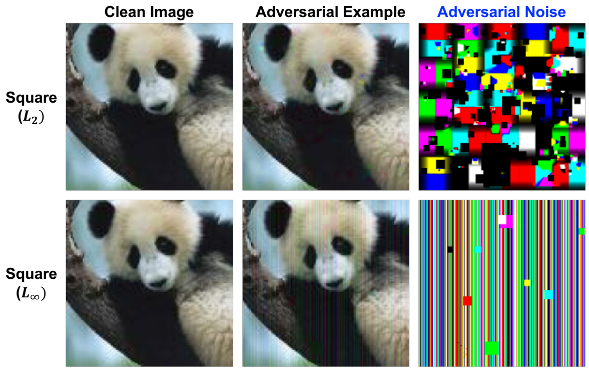


Figure 8. Visualizations of adversarial example and adversarial noise generated by Square [1] attack.

Experimental results of dynamic inference against full and purifier-aware white-box attacks As we aforementioned in the introduction, our paper focused on the real-world situation where the purifier is inaccessible. However, according to the suggestion of [20] that the robustness of the defense should be reported under as many conditions as possible, we conducted additional experiments for various attack scenarios including both *full* and *purifier-aware* white-box attack (in other words, strong adaptive attacks [2, 20]). To counter both strong attacks, we applied dynamic inference, which injects noises into an input image, to our NCIS and NRP (resG), as proposed in [22] and the Supplementary Material of NRP [15]. We implemented it by referring their code and we denote it as noise injection. We report the averaged experimental result for a single seed.

We evaluated each purifier using ImageNet validation set (50k images) with the ImageNet pre-trained ResNet-152 classifier, and untargeted L_∞ PGD attack ($\epsilon = 16/255, \alpha = 1.6/255$) with 10 attack iterations. In the case of NRP (resG) with the noise injection, we injected Gaussian noise $N(0, \sigma^2)$ into the input images before passing through the purifier. Similarly, for NCIS with the noise injection, we injected the same Gaussian noise $N(0, \sigma^2)$ *repeatedly* at each iteration of purifying process. Note that both methods without the noise injection denote its original method, respectively, and i denotes the number of iterations. We report the results for two cases of the noise injection, $\sigma = 0.03, 0.04$. For the full white-box attack, we generate adversarial examples with gradients through both the purifier and classifier together. For the purifier-aware attack, we consider the case where only the forward-pass outputs of the purifier are exposed to the attacker, not all the weights of the purifier. In this case, we used BPDA [2] which can attack the non-differentiable pre-processor based-method by approximating its gradients as the identity function. We used the implementation of BPDA in [7].

Table 9. Experimental results of dynamic inference for NRP(resG) and NCIS

ResNet-152	Noise Injection σ	Standard Accuracy	Purifier-blind PGD attack	Full white-box PGD attack	Purifier-aware white-box PGD attack (BPDA)
NRP (resG)	×	74.04	9.68	6.29	7.12
NRP (resG)	0.03	66.69	31.40	39.33	22.55
NRP (resG)	0.04	61.46	38.04	47.83	31.56
NCIS ($i = 7$)	×	68.93	48.06	1.29	8.72
NCIS ($i = 4$)	0.03	62.55	51.00	24.40	41.25
NCIS ($i = 3$)	0.04	63.26	51.60	37.64	40.49

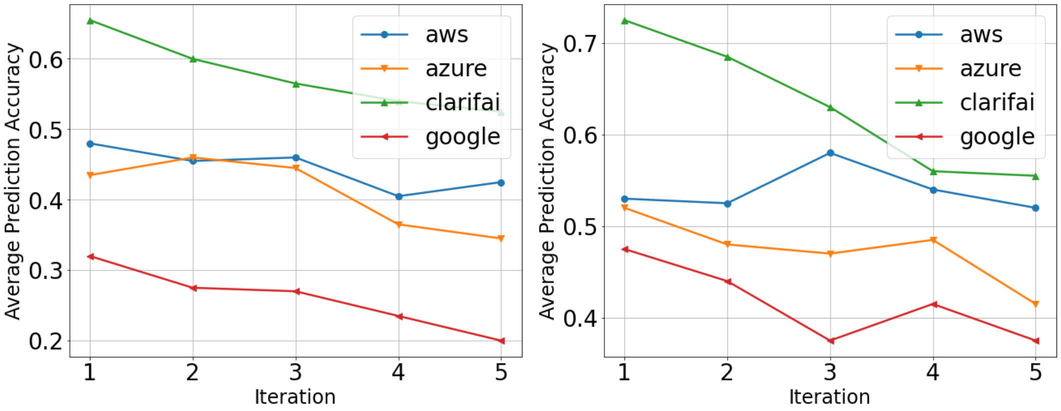
The first and fourth row in the Table 9 correspond to the original NRP (resG) [15] and NCIS, respectively. Unlike NCIS, NRP (resG) could not defend against purifier-blind PGD attack, as we already observed in the manuscript. In addition, Table 9 presents both NCIS and NRP (resG) are easily broken by both *full* and *purifier-aware* white-box attacks. The reason is that the purifier and classifier models are simply concatenated, and each layer of the neural network is differentiable, so the backward pass gradient can be easily calculated. Nevertheless, we confirm that applying the noise injection brings three advantages in both strong attack cases: First, robust accuracy against the purifier-blind attack slightly increases than the case without the noise injection. Although the level of increase of NRP (resG)' robust accuracy is larger than NCIS, the performance is not competitive with NCIS. Second, the optimal iteration number for the original NCIS is seven, but higher robust accuracy is obtained at shorter iterations. Since each optimal iteration number at $\sigma = 0.03$ and 0.04 is reduced to four and three from seven, it is beneficial for lowering both computational cost and inference time. Finally, adding random noise makes each purifier defend against both the full and purifier-ware white-box attacks. In particular, on average, $\sigma = 0.04$ is the best for both methods, and, in this variance, NCIS beats NRP (resG) in all cases except for the full white-box PGD attack. We believe that both purifier-blind and purifier-aware attacks are more likely to occur in the real world than the full white-box attack, and our method reveals more efficiency in these realistic scenarios. However, the noise injection has a common shortcomings that slightly decreases standard accuracy but NCIS more successfully maintains it in $\sigma = 0.04$.

In conclusion, even though we mainly consider the purifier-blind white-box attack, our simple variance of NCIS can be robust to not only the purifier-blind attack but also both the full and purifier-aware white-box attack, by easily applying the dynamic inference.

5. Additional Details on API Experiments

Dataset generation For generating benchmark datasets, we sampled images from ImageNet training dataset and generated adversarial examples of them using transfer-based black-box attack using ensemble of five classification models, ResNet-152, VGG-19, GoogleNet, ResNeXT-101, WideResNet-101, based on [13]. We attacked each image with targeted L_∞ PGD $(\epsilon = 16/255, \alpha = 1.6/255)$ attack with 10 attack iterations, and then made a pair of a clean and an adversarial example. Then, we queried each pair and stored them only when all the top five predicted labels of the clean and adversarial example were totally different. The above process was performed on all the four APIs, and we respectively sampled 100 pairs for test dataset and 20 pairs for validation dataset. In other words, we sampled 100 pairs of test dataset and 20 pairs of validation dataset for each API respectively. We will open all generated datasets publicly.

Evaluation metrics First, *Prediction Accuracy* is the measure for the same number of labels among top five labels between the predicted label of the purified image and the predicted label of the clean image. Second, *Top-1 Accuracy* is the measure for whether the Top-1 label of the purified image is same as the Top-1 label of the clean image. Finally, *Top-5 Accuracy* is the measure for whether the Top-1 label of the clean image exists within the Top-5 predicted labels of the purified image.



(a) Experimental results of GS

(b) Experimental results of NCIS

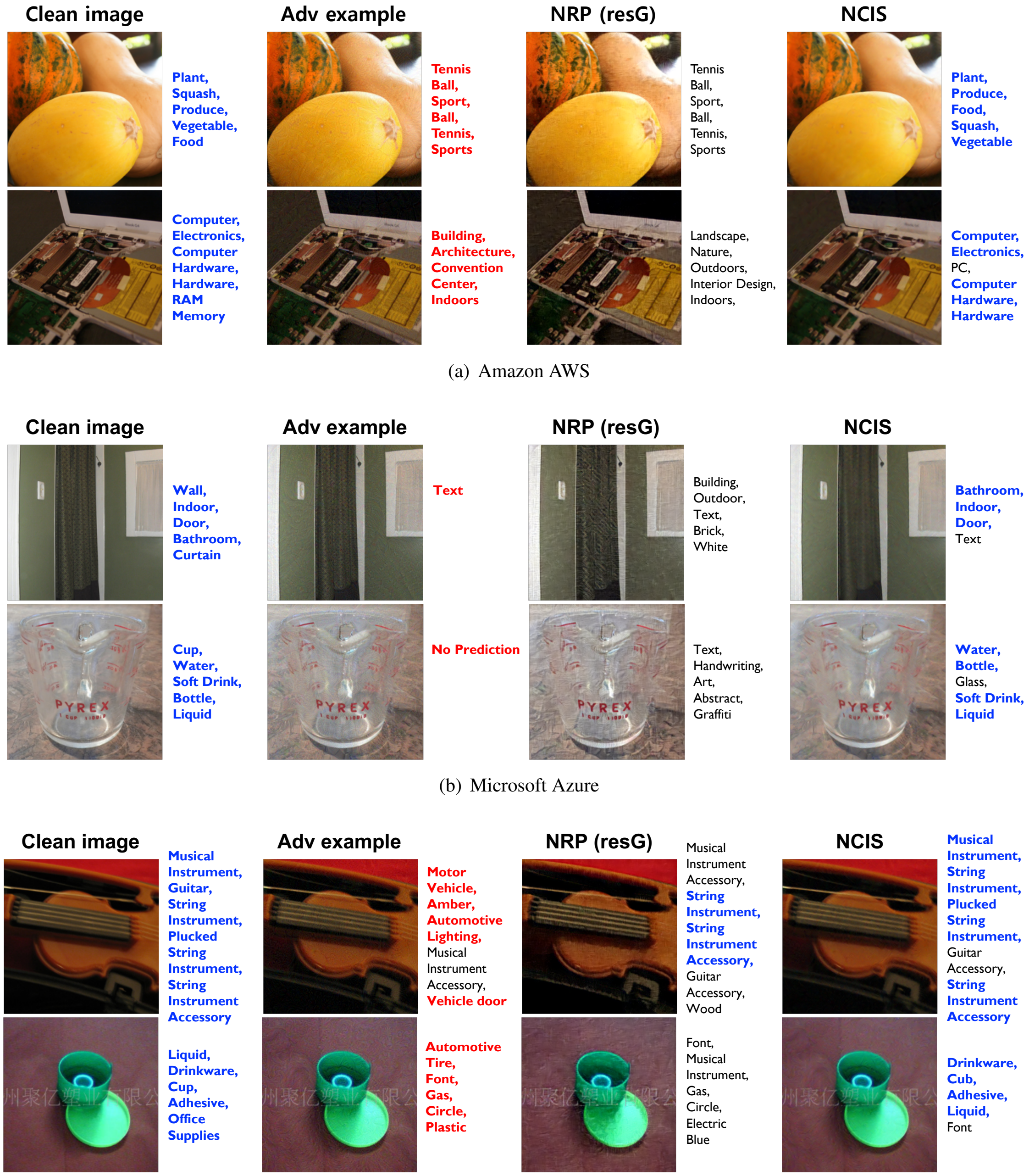
Figure 9. Experiments for hyperparameter selection of GS and NCIS.

Hyperparameter selection For GS and NCIS, we found the number of iterations i for each API by using the generated validation dataset. As a criterion, we only consider highest average Prediction Accuracy of a clean and adversarial example to select best i. Figure 9 shows the experimental results and Table 10 selected best i for each dataset. Note that all selected i are used for the experiments for APIs in the manuscript.

Visualization examples Figure 10 presents additional visualization examples of defending commercial vision APIs.

Table 10. Experimental results of hyperparameter selection.

Prediction Accuracy	AWS	Azure	Clarifai	Google
GS	0.48 $(i = 1)$	0.46 $(i = 2)$	0.65 $(i = 1)$	0.32 $(i = 1)$
NCIS	0.58 $(i = 3)$	0.51 $(i = 1)$	0.73 $(i = 1)$	0.48 $(i = 1)$


Clean image
Adv example
NRP (resG)
NCIS
Plant, Squash, Produce, Vegetable, Food
Tennis Ball, Sport, Ball, Tennis, Sports
Tennis Ball, Sport, Ball, Tennis, Sports
Plant, Produce, Food, Squash, Vegetable
Computer, Electronics, Computer Hardware, Hardware, RAM Memory
Building, Architecture, Convention Center, Indoors
Landscape, Nature, Outdoors, Interior Design, Indoors,
Computer, Electronics, PC, Computer Hardware, Hardware
Clean image
Adv example
NRP (resG)
NCIS
Wall, Indoor, Door, Bathroom, Curtain
Text
Building, Outdoor, Text, Brick, White
Bathroom, Indoor, Door, Text
Cup, Water, Soft Drink, Bottle, Liquid
No Prediction
Text, Handwriting, Art, Abstract, Graffiti
Water, Bottle, Glass, Soft Drink, Liquid
Clean image
Adv example
NRP (resG)
NCIS
Musical Instrument, Guitar, String Instrument, Plucked String Instrument, String Instrument Accessory
Motor Vehicle, Amber, Automotive Lighting, Musical Instrument Accessory, Vehicle door
Musical Instrument Accessory, String Instrument, String Instrument Accessory, Guitar Accessory, Wood
Musical Instrument, String Instrument, Plucked String Instrument, Guitar Accessory, String Instrument Accessory
Liquid, Drinkware, Cup, Adhesive, Office Supplies
Automotive Tire, Font, Gas, Circle, Plastic
Font, Musical Instrument, Gas, Circle, Electric Blue
Drinkware, Cub, Adhesive, Liquid, Font

(a) Amazon AWS

(b) Microsoft Azure

(c) Google

Figure 10. Visualization examples of defending commercial vision APIs. The APIs predict correct top-5 predictions of the original clean images (first column), and when completely fooled by the adversarial examples (second column). The right two columns show the prediction results when two purifiers, NRP (resG) [15] and our NCIS, are applied to the adversarial examples.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In European Conference on Computer Vision, pages 484–501. Springer, 2020. 7, 8

[2] 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, pages 274–283. PMLR, 2018. 9

[3] Jaeseok Byun, Sungmin Cha, and Taesup Moon. Fbi-denoiser: Fast blind image denoiser for poisson-gaussian noise. arXiv preprint arXiv:2105.10967, 2021. 3

[4] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705, 2019. 7

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pages 39–57. IEEE, 2017. 1, 2

[6] 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. 7

[7] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. arXiv preprint arXiv:1902.07623, 2019. 9

[8] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 321–331, 2020. 8

[9] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117, 2017. 1

[10] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117, 2017. 4

[11] 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. 3

[12] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. arXiv preprint arXiv:2010.01950, 2020. 2, 7, 8

[13] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770, 2016. 10

[14] 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. 1, 2

[15] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 262–271, 2020. 1, 9, 11

[16] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10428–10436, 2020. 3, 5, 6, 7

[17] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena, 60(1-4):259–268, 1992. 4

[18] 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, 2020. 7

[19] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervised learning. In International Conference on Learning Representations, 2020. 1

[20] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. arXiv preprint arXiv:2002.08347, 2020. 9

[21] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1492–1500, 2017. 3, 5, 6

[22] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. arXiv preprint arXiv:2106.06041, 2021. 9

[23] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016. 3, 5, 6