Supplementary Materials for Adversarial Purification with Score-based Generative Models

A. Experimental details

A.1. Software and Hardware Configurations

We implemented our code on Python version 3.8.5 and PyTorch version 1.7.1 with Ubuntu 18.04 operating system. We run each of our experiments on a single Titan X GPU with Intel Xeon CPU E5-2640 v4 @ 2.40GHz. Our implementation is available at `https://github.com/jmyoon1/adp`.

A.2. Dataset details

MNIST is the dataset that consists of handwritten digits. It consists of a training set of 60,000 examples, and a test set of 10,000 examples. MNIST is a grayscaled dataset with 28×28 size at a total of 784 dimensions, and its label consists of 10 digits.

FashionMNIST is the dataset that consists of clothes. It consists of a training set of 60,000 examples, and a test set of 10,000 examples. Like MNIST, FashionMNIST is a grayscaled dataset with 28×28 size, where its label is included in one of 10 classes of clothes. The full list of classes is as follows: {T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot}.

CIFAR-10 is the dataset that consists of colored images. It consists of a training set of 45,000 examples, a validation set of 5,000 examples, and a test set of 10,000 examples. CIFAR-10 is an RGB-colored dataset with 32×32 size, at a total of 3,072 dimensions each data, where its label belongs to one of the following ten classes. The full list of classes is as follows: {airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, trucks}.

CIFAR-100 is also the dataset that consists of colored images. It consists of a training set of 45,000 examples, a validation set of 5,000 examples, and a test set of 10,000 examples. Like CIFAR-10, CIFAR-100 is an RGB-colored dataset with 32×32 size.

CIFAR-10-C is the dataset that consists of corrupted CIFAR-10 examples. It consists of 15 types of adversaries, denoted to *common corruption*, with 5 severities each. We introduce some samples from common corruption examples at Appendix B.

A.3. Training hyperparameters

Table 5. Hyperparameters for training score networks.

Dataset	σ_1	σ_L	L	Training iterations	Batch size
MNIST	15	0.005253	110	200,000	128
FashionMNIST	15	0.005253	64	200,000	128
CIFAR-10	50	0.008454	232	300,000	128
CIFAR-100	50	0.008454	232	300,000	128
CIFAR-10, DCT Augmented	50	0.08454	232	200,000	128
CIFAR-10, AugMix Augmented	50	0.08454	232	200,000	128

We present the hyperparameters that are used for training our purifier networks having NCSNv2 architecture in Table 5. Here, σ_1 and σ_L stands for the largest and smallest standard deviation of the isotropic Gaussian noise for training NCSNv2, L is the number of steps of noise standard deviations. We follow Song & Ermon (2020) to get appropriate hyperparameters.

When we train NCSN with DCT- or AugMix-augmented perturbations to enhance robustness in CIFAR-10-C evaluation, the smallest noise level σ_L is adjusted since out-of-distribution examples might be over-represented for training with small noise levels, because the distance between the original and perturbed inputs will become farther compared to NCSN trained with Gaussian perturbations.

For training all the classifier and purifier networks, we use Adam optimizer with learning rate 0.001 and $(\beta_1, \beta_2) = (0.9, 0.999)$, and no weight decay. We disabled horizontal flip at MNIST and FashionMNIST datasets, and enabled it at CIFAR-10, CIFAR-100 datasets.

A.4. Neural Network Descriptions

For CIFAR-10, CIFAR-10-C and CIFAR-100 datasets, we use WideResNet-28-10 (Zagoruyko & Komodakis, 2016) for classification and NCSNv2 (Song & Ermon, 2020) which is a modified version of RefineNet (Lin et al., 2017) for purification. The overall structures are depicted in Fig. 6. In CNN architecture for FashionMNIST classifier, we use filter size 5×5, stride 1, and padding 2 in our pytorch implementation. For WideResNet-28-10 architecture for classifier for larger datasets (CIFAR-10, CIFAR-100), we use filter size 3×3, stride 1, and padding 1 in our Pytorch implementation.

We also describe the detailed NCSN structure in Fig. 7. Here, N denotes the number of channels. The RefineNet (Lin et al., 2017) structure is used as the decoder part of NCSN.

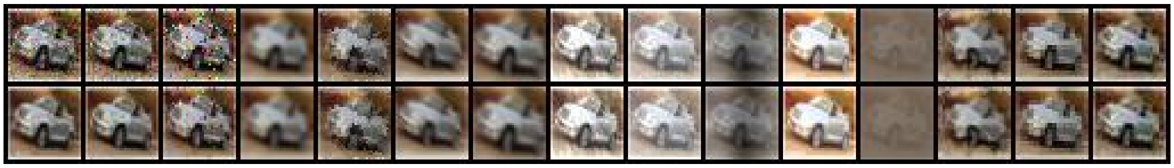


Figure 5. Examples of corrupted and purified images. From left: {Gaussian, shot, impulse} noise, {Defocus, glass, motion, zoom} blur, {snow, frost, fog, brightness} weather, {contrast, elastic, pixelate, JPEG} digital corruptions.

B. Common corruption and purified examples

Fig. 5 shows the examples of images corrupted with severity level 5 and their corresponding purified counterparts. The order of corrupted images is the same as indicated in Table 11.

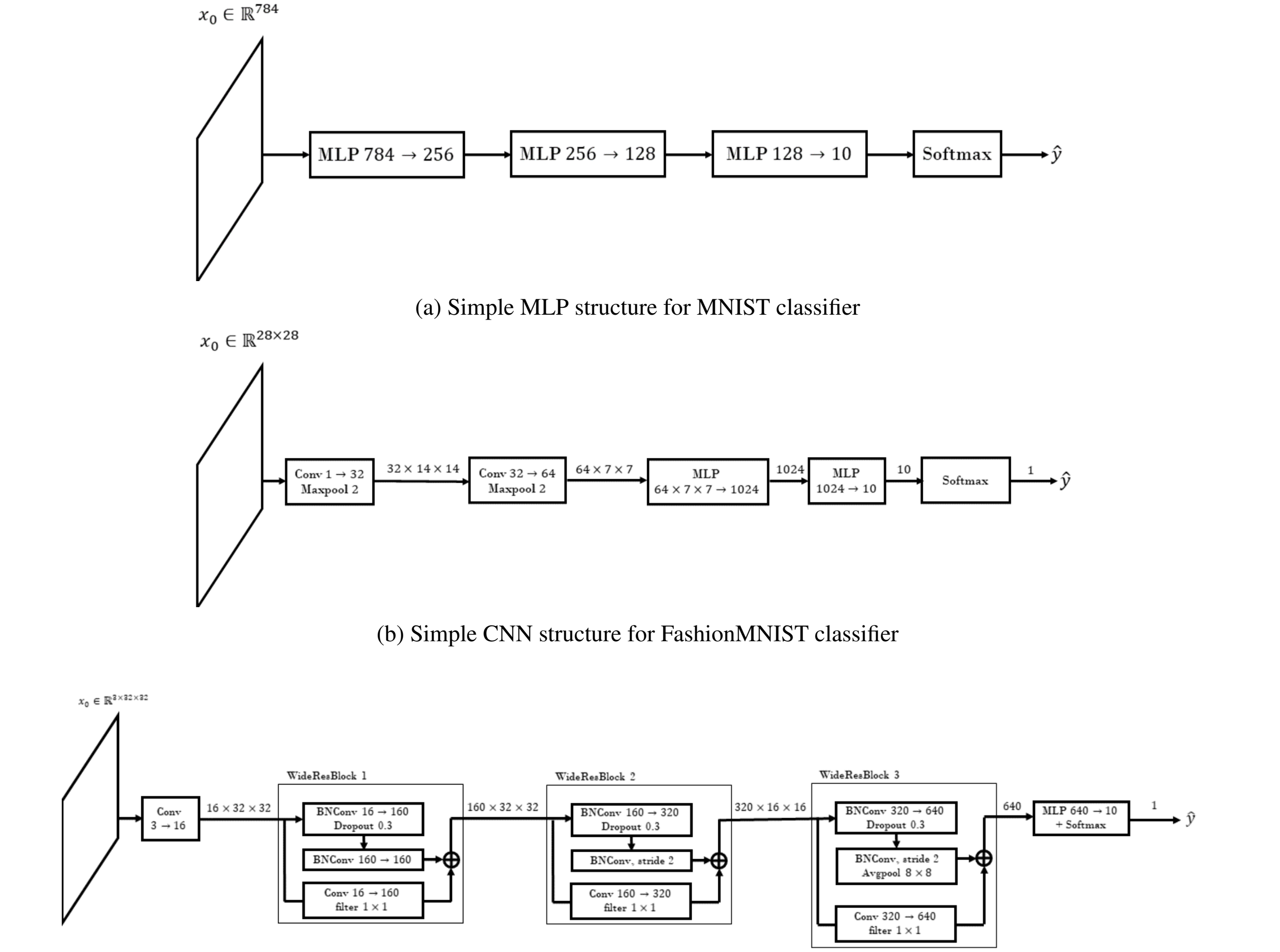
C. Full CIFAR-10-C performance

In this section, we report our full common corruption performances. For implementation of adversarial training methods, we import from the `RobustBench` (Croce et al., 2020) benchmark. We present both the full and average performances for every 15 kind of corruptions in Table 11. We used adaptive step size $\alpha = 0.05$ for ADP trained with Gaussian and DCT perturbation, and exponentially decreasing deterministic step size from $\sigma_1 = 0.08$ to $\sigma_{10} = 8.0 \times 10^{-4}$ for ADP trained with AugMix augmented perturbation. For (DCT+AugMix) case, we iteratively purify the input with those two purifier models 10 times in rotation. For comparison, the results at the second category include recent adversarial training cases, and the third category include recent data augmentation and domain adaptation methods.

When training NCSN with DCT or AugMix augmentations, we slightly modified the DSM objective. The idea is, as described in the main text, to modify the perturbation distribution from Gaussian distributions centered at original images to Gaussian distributions centered at augmented images.

For DCT training of the purifier network, we first take DCT to original images, and drop the frequency components with smallest eigenvalues, until the sum of dropped coefficients reach 5% of the sum of their eigenvalues. Then the smallest noise level σ_L in the NCSN objective is multiplied by 10, since DCT-transformed images are more deviated from original images than conventional noisy images and norm-based attacked images in terms of l_2-distance, and too small noise level may over-represent the deviation by DCT transformation. Then, we replace the perturbation distribution by $q(\tilde{x}|x) = \mathcal{N}(\tilde{x}|F(x), \sigma^2 I)$ where $F(x)$ is a DCT-transformed image from x.

AugMix (Hendrycks et al., 2020) has even larger deviation than DCT augmentation, so it is more difficult to train the purifier network with it. Instead of directly targeting the original image, we first generate the auxiliary image that locates



(a) Simple MLP structure for MNIST classifier

(b) Simple CNN structure for FashionMNIST classifier

(c) Simple CNN structure for classifying CIFAR-10 and CIFAR-100. For TinyImageNet, all image sizes and avgpool size are doubled.

Figure 6. Neural network architecture for classifier networks

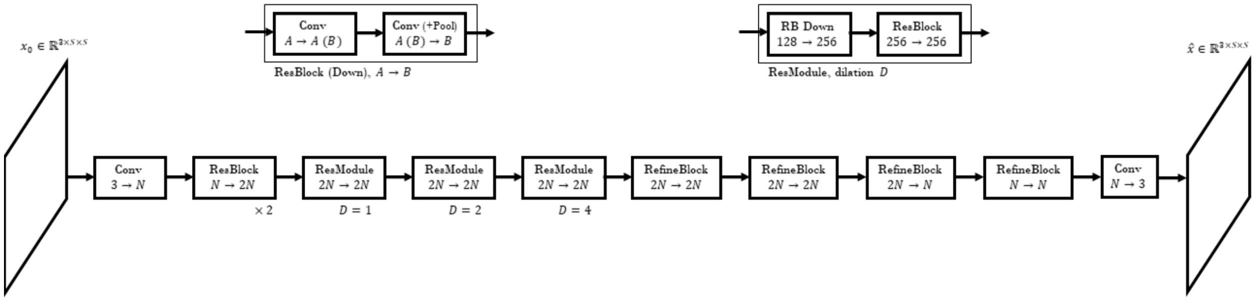


Figure 7. Neural network architecture for the NCSN purifier network

Table 6. Evaluation results for adaptive attacks. Threat model: l_∞ ε-ball with $\varepsilon = 8/255$, CIFAR-10 dataset. The white-box attack results for adversarial training methods are also referred for comparison.

	Natural	Robust	Preprocessor	Classifier	Attack method	Threat blindness
ADP (Adaptive LR)	86.14					
($\sigma = 0.25$), BPDA step 40		70.01	NCSNv2	WRN-28-10	BPDA+EOT	Unseen
($\sigma = 0.25$), BPDA step 100		69.71	NCSNv2	WRN-28-10	BPDA+EOT	Unseen
($\sigma = 0.25$)		70.61	NCSNv2	WRN-28-10	Joint (score)+EOT	Unseen
($\sigma = 0.25$)		78.39	NCSNv2	WRN-28-10	Joint (full)+EOT	Unseen
($\sigma = 0.25$)		80.80	NCSNv2	WRN-28-10	SPSA	Unseen
($\sigma = 0.25$) with detection	95.74	69.85	NCSNv2	WRN-28-10	BPDA+EOT	Unseen
(Hill et al., 2021) (1500 iterations)	84.12	54.90	IGEBM	WRN-28-10	BPDA+EOT	Unseen
(Yang et al., 2019) (Natural)	94.8	40.8	Masking+Recon.	ResNet-18	BPDA	Unseen
(Yang et al., 2019) (AT)	-	52.8	Masking+Recon.	ResNet-18	BPDA	Unseen
(Yang et al., 2019) (AT)	88.7	55.1	Masking+Recon.	WRN-28-10	BPDA	Seen
(Yang et al., 2019) (Natural)	89.4	41.5	Masking+Recon.	ResNet-18	Approx. Input	Unseen
(Yang et al., 2019) (AT)	88.7	62.5	Masking+Recon.	ResNet-18	Approx. Input	Seen
(Song et al., 2018)	95	5	PixelCNN	ResNet-62	BPDA	Seen
(Madry et al., 2018)	87.3	45.8	Robust Classifier	ResNet-18	Full PGD	Seen
(Zhang et al., 2019)	84.90	56.43	Robust Classifier	ResNet-18	Full PGD	Seen
(Carmon et al., 2019)	89.70	62.50	Robust Classifier	WRN-28-10	Full PGD	Seen

comparatively near to the perturbed point, then target to the auxiliary image. We first replace the perturbation distribution by $q'(\tilde{x}|x) = \mathcal{N}(\tilde{x}|F(x), \sigma^2 I)$ where $F(x)$ is an AugMix-transformed image from x. Then we replace the DSM objective Eq. (8) with

$$\ell(\theta, \sigma) = \mathbb{E}_{q'(\tilde{x}|x)p_{\text{data}}(x)} \left[\frac{1}{2\sigma^4} \left\| \tilde{x} + \sigma^2 s_\theta(x') - x' \right\|^2 \right] \tag{20}$$

where $x' = \frac{x+F(x)}{2}$ is the midpoint of x and $F(x)$. That is, to ease the reconstruction from highly corrupted images $F(x)$, we choose to learn $s_\theta(x)$ to recover from the midpoint x'.

D. Detailed results for strong adaptive attacks

D.1. Full list of defense results for adaptive attacks

In this section, we present the full list of defense results for strong adaptive attacks contained in Table 3 of the main paper in Table 6, including the preprocessor and classifier architectures, attack method, and blindness against the threat model. The results at the first, second, and third category includes our work, recently proposed preprocessor-based defense methods, and existing adversarial training-based defense methods, respectively. *Approx. Input* (Yang et al., 2019) first iteratively updates inputs by classifier PGD followed by purification, and thus add classifier gradients to purified images instead of clean images. The term *with detection* denotes our method with the procedure of detecting adversarial examples before the purification, as described in Appendix F. Our method with detection can increase clean accuracy because it can filter out natural images and prevent the need of unnecessary purification.

D.2. Performance with various noise injection levels

We present the standard and robust accuracy of ADP for the strong adaptive BPDA+EOT attack as well as the preprocessor-blind classifier PGD attacks in CIFAR-10 dataset, from $\sigma = 0.05$ to $\sigma = 0.4$ in Table 7. As the noise level increases, both the standard accuracy and the gap between standard and robust accuracy decrease, as the standard accuracy falls much faster than the robust accuracy as the injected noise becomes stronger. Although both attacks are held in the same threat models, the best robust accuracy at the classifier PGD attack is achieved at much less injected noise than the BPDA+EOT attack, implying that the classifier PGD attack actually needs less noise injection than BPDA+EOT attack for optimal purification.

D.3. Effect of number of EOT for BPDA attacks

We present the robust accuracy of ADP over BPDA+EOT attacks with different number of EOT in Table 8 in CIFAR-10 dataset.

Table 7. CIFAR-10 results for different levels of noise injections on attacked images, from $\sigma = 0.05$ to $\sigma = 0.4$ with preprocessor-blind classifier PGD attacks.

Method	Accuracy (%)			Architecture	Blindness
	Standard	BPDA+EOT	Clf PGD		
ADP ($\sigma = 0.05$)	93.35	6.08	66.94	WRN-28-10	Unseen
ADP ($\sigma = 0.10$)	93.09	41.06	**87.13**	WRN-28-10	Unseen
ADP ($\sigma = 0.15$)	90.36	57.73	86.34	WRN-28-10	Unseen
ADP ($\sigma = 0.20$)	86.80	67.36	85.74	WRN-28-10	Unseen
ADP ($\sigma = 0.25$)	86.14	**70.01**	83.93	WRN-28-10	Unseen
ADP ($\sigma = 0.30$)	80.98	69.06	78.89	WRN-28-10	Unseen
ADP ($\sigma = 0.35$)	79.44	69.70	77.54	WRN-28-10	Unseen
ADP ($\sigma = 0.40$)	77.41	69.67	75.80	WRN-28-10	Unseen

Table 8. BPDA+EOT attack results for different number of EOT in CIFAR-10 dataset. The input is attacked after (#EOT) different random noise injections with $\sigma = 0.25$ via BPDA attack.

Number of EOT	1	3	5	10	15	30	50
Accuracy (%)	76.46	73.89	72.90	69.86	70.01	68.75	67.60

D.4. Effect of BPDA iterations

The experiments of the main paper are all performed with 40 iterations of BPDA attacks. To measure the effect of the number of BPDA iterations, we also run experiments with 100 iterations of BPDA attacks. As contained in Table 6, increasing the number of iterations from 40 to 100 slightly decrease the robust accuracy by 0.30%.

D.5. Effect of purification runs

In the main paper, we fixed the maximum number of purification runs to 10. In this section, we present robust accuracy of ADP on different number of purification runs in CIFAR-10 dataset, with BPDA+EOT attack with 40 BPDA iterations and 15 EOT attacks. As described in Fig. 8, we observed that the robust accuracy is improved until 10 runs, and stay stable for more runs.

D.6. Full list of defense results for more datasets

We present the full list of defense results for various datasets, including MNIST, FashionMNIST, and CIFAR-100 in Table 9.

E. Robust accuracy of Randomized Smoothing Classifiers

We present the standard accuracy of randomized smoothing classifiers of ADP on CIFAR-10 dataset in Table 10. We see that on low noise levels up to $\sigma = 0.25$, the robust accuracy of the randomized smoothing classifier performing ADP surpasses those of the existing randomized smoothing classifiers.

F. Detecting Adversarial Examples before purification

While random noise injection before purification improves the robust accuracy, this degrades the standard accuracy because the features helpful for natural image classification can also be screened out. To prevent this, we propose a detection and noise injection scheme where we first classify an image into attacked or natural image and apply different noise injection policies according to the classification result. We draw the histogram of the score norms $\|s_\theta(x)\|$ for natural, adversarial and purified images for various attacks in Fig. 9. Except for joint attacks, attacked images usually have higher score norms than natural images, showing the promises of our method for detecting adversarial examples before purifications.

The detection of the attacked is based on the score norms. We choose the threshold τ, and classify an image whose Euclidean norm of the estimate score $\|s_\theta(x)\|_2$ below the threshold as a natural image, and an image whose score norm above the threshold as an attacked image. Figure 3 shows the histograms of score norms for natural and attacked images. As shown in

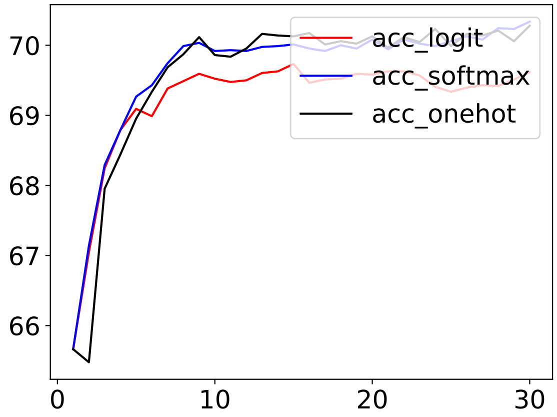


Figure 8. Robust accuracy of CIFAR-10 dataset under BPDA+EOT attack on different purification runs. The x-axis stands for the number of purification runs and y-axis stands for accuracy (%). The red, , and black line stand for expectation over pre-softmax outputs, post-softmax outputs, and argmax outputs, respectively.

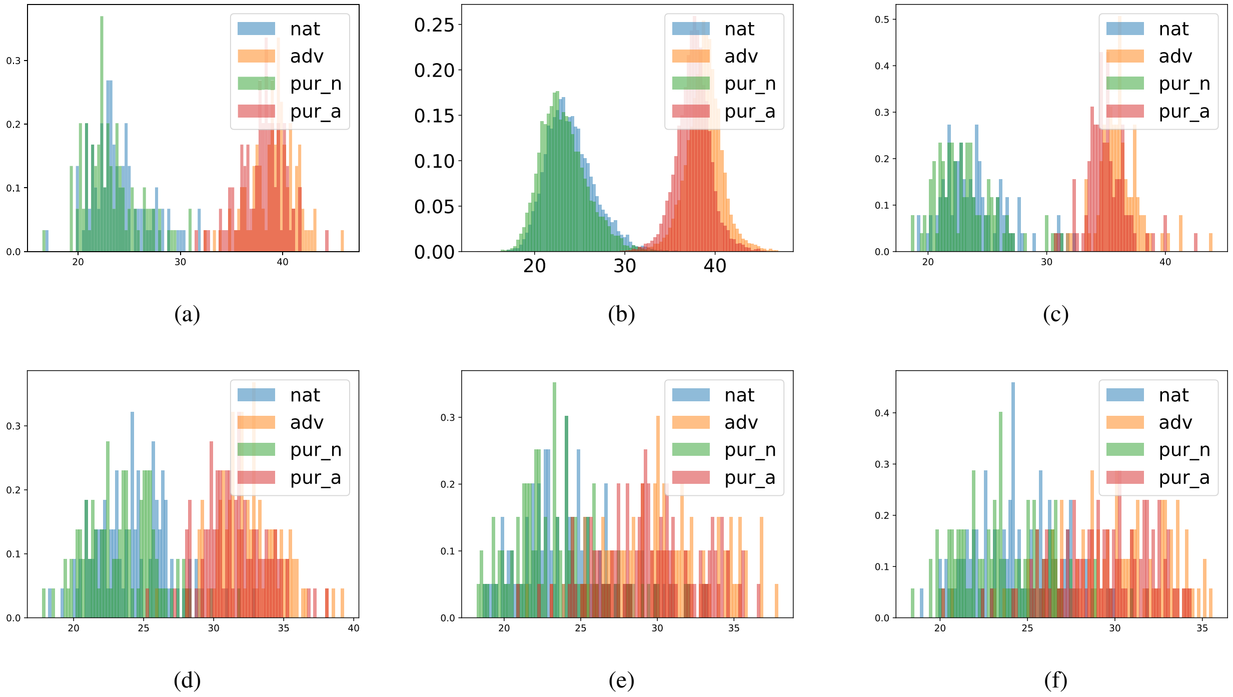


Figure 9. Histogram of score function norms $\|s_\theta(x)\|$ for natural, adversarial and purified images. Pur_a and Pur_n denotes scores of one-step purified adversarial and natural images, respectively. The x-axis and y-axis stand for the score norm $\|s_\theta(x)\|_2$ and the probability density, respectively. From upper left to lower right: (a) Classifier PGD (b) BPDA (c) Approximate input (d) One-step unrolling (e) Joint (full) (f) Joint (score). The x-axis and y-axis represent $\|s_\theta(x)\|$ and the probability density, respectively. One step unrolling attack is an adaptive attack where the PGD attack is performed under the composition of the classifier and one-step forward propagation of the purifier network.

Table 9. Evaluation results for more datasets.

Dataset Defense methods	ε	Attack type	Accuracy (%) Standard	 Robust
MNIST	0.3	Clf PGD	98.07	96.41
FashionMNIST	$8/255$	Clf PGD	93.19	86.62
CIFAR-100				
Raw WideResNet			79.86	
$\sigma = 0.0$, det 0.08	$8/255$	Clf PGD	77.83	43.21
$\sigma = 0.25$, $\alpha = 0.05$	$8/255$	BPDA+EOT	60.66	39.72
(Hill et al., 2021)	$8/255$	BPDA+EOT	51.66	26.10
AT (Madry et al., 2018)	$8/255$	PGD	59.58	25.47
(Li et al., 2020)	$8/255$	PGD	61.01	28.88

Table 10. Robust accuracy of randomized smoothing classifiers.

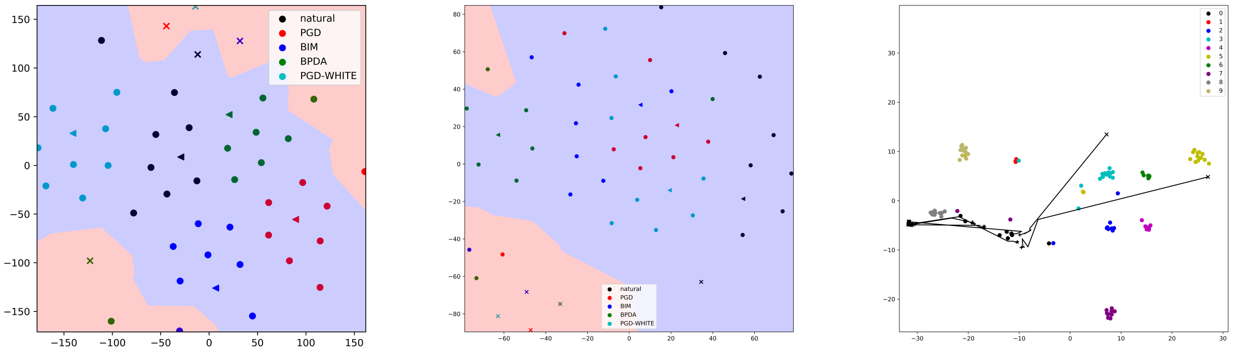
Models	Noise level σ 0.12	 0.25	 0.5	 1.0
ADP	93	86	62	27
(Cohen et al., 2019)*	81	75	65	47
(Salman et al., 2019)*	84	77	68	50

the figure, the score norm is a good criterion for detecting adversarial examples.

Having decided that an image is an attacked image, we inject higher noise level $\sigma_{\text{high}} = \sigma$ (the one obtained with the heuristic described in Section 3). Otherwise, we apply the low noise level $\sigma_{\text{low}} = \beta\sigma$ with β fixed to 0.2. For all experiments on CIFAR-10 and CIFAR-100 datasets, we fixed $\tau = 25.0$.

G. Decision Boundary Plot with t-SNE

Fig. 10 shows the decision boundaries and trajectories over purification steps for existing attacks. We draw t-SNE (van der Maaten & Hinton, 2008) diagrams for attacked and purified images and their corresponding features, and draw Voronoi diagrams to discriminate between correctly classified images and failed ones. Moreover, we display a trajectory of purifying image drawn on the t-SNE diagrams, starting from the attacked images and ends with the purified images. We show that the features of attacked images locate far from the natural images in the feature domain, and approaches to those of the natural images via the purification process.



(a)

(b)

(c)

Figure 10. t-SNE diagram of decision boundaries on (a) Image space and (b) Feature space, with respect to various attacks on a single data. The blue and red regions represent the region whose predictions are equal to and different to the ground truth labels, respectively. (c) shows the trajectory of features of attacked images to predicted ones.

Table 11. Performance for CIFAR-10-C dataset

Models		Noise			Blur				Weather				Digital			
	Average	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixelate	JPEG
Raw WideResNet	71.89	32.94	47.78	46.37	82.67	50.41	78.51	79.38	83.86	78.71	87.94	94.43	75.21	84.84	77.52	77.76
ADP																
$\sigma = 0.0$	80.49	91.43	91.07	71.26	82.55	52.36	78.36	79.20	83.99	79.26	87.85	94.44	75.07	84.96	77.76	77.74
$\sigma = 0.25$	77.45	84.87	85.09	84.08	81.54	66.19	76.21	78.13	83.13	79.04	62.99	86.76	48.04	79.90	82.53	83.21
$\sigma = 0.25$+Detection	78.96	84.86	85.09	83.75	78.58	58.68	74.13	75.64	84.88	83.26	76.47	92.75	53.06	83.45	83.87	86.00
$\sigma = 0.1$	76.25	88.80	88.52	83.32	74.82	62.56	67.18	69.64	80.70	81.48	64.08	89.26	52.20	75.72	81.86	83.64
(+DCT Augmentation)	67.67	82.04	83.40	79.20	63.76	56.60	53.32	59.40	76.8	75.44	50.88	81.64	40.24	63.00	74.28	75.04
$\sigma = 0.0$, deterministic LR	80.09	90.28	89.55	68.81	82.53	51.36	78.39	79.20	83.95	79.13	88.09	94.49	75.40	84.83	77.58	77.79
(+DCT Augmentation)	80.74	84.94	86.79	81.24	80.29	58.67	74.53	76.64	86.13	85.55	87.58	94.25	73.87	82.72	78.86	82.49
($\times 10$ training var)	82.63	88.60	90.32	83.64	82.36	62.00	76.64	79.80	87.68	87.00	87.64	92.96	76.40	80.68	78.68	85.04
(DCT+AugMix)	82.40	87.00	89.48	78.68	85.92	55.84	80.04	81.40	84.64	84.80	89.72	93.28	78.48	83.24	80.44	83.04
TRADES (Zhang et al., 2019)	75.63	79.17	80.45	73.85	80.05	77.96	76.50	78.97	80.42	76.58	60.30	82.63	43.11	78.87	82.73	82.81
RST (Carmon et al., 2019)	80.40	82.49	84.14	76.98	85.47	81.71	81.92	84.65	84.57	82.70	65.90	87.59	49.01	84.05	87.68	87.20
(Cohen et al., 2019)	73.70	82.69	82.95	78.81	74.81	74.37	69.12	72.09	76.90	74.90	56.89	80.09	45.13	73.89	80.66	82.14
AugMix (Hendrycks et al., 2020)	88.78	81.68	86.52	85.78	94.21	79.35	92.23	92.94	89.69	89.37	91.73	94.24	90.14	90.31	86.06	87.40
TENT (Wang et al., 2021)	89.52	85.22	87.82	83.78	93.84	80.04	91.82	93.02	89.58	89.84	93.56	94.54	94.10	89.54	91.50	84.66
DCT (Hossain et al., 2020)	89.17	85.10	88.90	86.40	94.60	78.60	90.20	91.60	89.30	90.40	91.20	94.10	80.70	90.50	91.70	94.20