APPENDIX

A DETAILED DESCRIPTIONS OF THE ALGORITHM FOR COMPUTING DUAL-PERTURBATION EXAMPLES

We use the following steps to solve the optimization problem of dual-perturbation attacks:

1. *Initialization*. Start with a random initial starting point $\boldsymbol{\delta}^{(0)}$. To do this, randomly sample a data point $\boldsymbol{\delta}_F^{(0)}$ in ℓ_p ball $\Delta(\epsilon_F)$ and $\boldsymbol{\delta}_B^{(0)}$ in $\Delta(\epsilon_B)$. Then, $\boldsymbol{\delta}^{(0)}$ can be obtained by using $\boldsymbol{\delta}^{(0)} = \boldsymbol{\delta}_F^{(0)} \circ \mathcal{F}(\boldsymbol{x}) + \boldsymbol{\delta}_B^{(0)} \circ \mathcal{B}(\boldsymbol{x})$. This ensures that the initial perturbation is feasible in both foreground and background.

2. *Split*. At the k-th iteration, split the perturbation $\boldsymbol{\delta}^{(k)}$ into $\boldsymbol{\delta}_F^{(k)}$ for foreground and $\boldsymbol{\delta}_B^{(k)}$ for background:
$$\begin{cases} \boldsymbol{\delta}_F^{(k)} = \boldsymbol{\delta}^{(k)} \circ \mathcal{F}(\boldsymbol{x}) \\ \boldsymbol{\delta}_B^{(k)} = \boldsymbol{\delta}^{(k)} \circ \mathcal{B}(\boldsymbol{x}) \end{cases}. \tag{1}$$
Then update the foreground and background perturbations seperately using the following rules:
$$\begin{cases} \boldsymbol{\delta}_F^{(k+1)} = \mathcal{P}_\epsilon(\boldsymbol{\delta}_F^{(k)} + \alpha_F \cdot g_F) \\ \boldsymbol{\delta}_B^{(k+1)} = \mathcal{P}_\epsilon(\boldsymbol{\delta}_B^{(k)} + \alpha_B \cdot g_B) \end{cases} \tag{2}$$
where g_F is the update that corresponds to the *normalized steepest descent* constrained in the foreground, and g_B for the background. Specifically,
$$\begin{cases} g_F = \mathcal{G}(\mathcal{F}(\boldsymbol{x}) \circ \nabla_{\boldsymbol{\delta}^{(k)}} \{\mathcal{L}(h_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}^{(k)}), y)) + \lambda \cdot \mathcal{S}\left(\boldsymbol{x} + \boldsymbol{\delta}^{(k)}\right)\} \\ g_B = \mathcal{G}(\mathcal{B}(\boldsymbol{x}) \circ \nabla_{\boldsymbol{\delta}^{(k)}} \{\mathcal{L}(h_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}^{(k)}), y)) + \lambda \cdot \mathcal{S}\left(\boldsymbol{x} + \boldsymbol{\delta}^{(k)}\right)\} \end{cases} \tag{3}$$
where α_F is the stepsize for foreground, and α_B is the stepsize for background.

3. *Merge*. At the end of the k-th iteration, merge the perturbations obtained in the last step by using
$$\boldsymbol{\delta}^{(k+1)} = \boldsymbol{\delta}_F^{(k+1)} + \boldsymbol{\delta}_B^{(k+1)}. \tag{4}$$
$\boldsymbol{\delta}^{(k+1)}$ is further used to derive the update for the normalized steepest descent at the next iteration.

4. Return to step 2 or terminate after either a fixed number of iterations.

B DESCRIPTIONS OF DATASETS

B.1 SEGMENT-6

The statistics of the Segment-6 dataset are displayed in Table 1.

Class	**Number of samples**	
	Training	**Test**
Train	3,000	200
Bird	3,000	200
Cat	3,000	200
Dog	3,000	200
Toilet	3,000	200
Clock	3,000	200
Total	18,000	1,200

Table 1: Number of samples in each class of the Segment-6 dataset.

Class	Number of samples	
	Training	Test
Airplane	500	10
Bird	500	10
Car	500	10
Cat	500	10
Deer	500	10
Dog	500	10
Horse	500	10
Monkey	500	10
Ship	500	10
Truck	500	10
Total	5,000	100

Table 2: Number of samples in each class of the STL-10 dataset.

B.2 STL-10

The statistics of the STL-10 dataset are displayed in Table 2.

B.3 IMAGENET-10

The labels and number of images per class in the ImageNet-10 dataset are listed in Table 3.

Class	Number of samples	
	Training	Test
Airplane	500	10
Car	500	10
Cat	500	10
Dog	500	10
Truck	500	10
Elephant	500	10
Zebra	500	10
Bus	500	10
Bear	500	10
Bicycle	500	10
Total	5,000	100

Table 3: Number of samples in each class of the ImageNet-10 dataset.

C IMPLEMENTATIONS

We implemented all the attack model, as well as the defense approaches in PyTorch[1], an open-source library for neural network learning. We used the ResNet34 model (He et al., 2016) and standard transfer learning, as the datasets employed in our experiments do not have a sufficient amount of data to achieve high accuracy. Specifically, we initialized the network with the model pre-trained on ImageNet, reset the final fully connected layer, and added a *normalization layer* in front of the ResNet34 model, which performs a channel-wise transformation of an input by subtracting $(0.485, 0.456, 0.406)$ (the mean of ImageNet) and then being divided by $(0.229, 0.224, 0.225)$ (the standard deviation of ImageNet); [2] then, we train the neural networks as usual.

[1] Available at `https://pytorch.org/`.

[2] To fit the Segment-6 dataset which contains much smaller images compared to ImageNet, we also reset the first convolutional layer of the pre-trained ResNet34 model by reducing the kernel size from 7×7 to 3×3, stride from 2 to 1, and pad from 3 to 1.

Unless otherwise specified, we used 60 epochs with training batch size 128 for Segment-6. For STL-10 and ImageNet-10. we trained the classifiers for 20 epochs by using a batch size of 64. We used Adam Optimizer (Kingma & Ba, 2014) with initial learning rate of 10^{-4} for *Clean*, and 10^{-3} for *AT-PGD* and *AT-Dual*, respectively. We dropped the learning rate by 0.1 every 20 epochs on Segment-6, and similarly at the 8th and 15th epochs on STL-10 and ImageNet-10.

As mentioned above, we implemented *PGD* and *dual-perturbation* attacks, bounded by both ℓ_∞ and ℓ_2 norms, to evaluate robustness of a classification model, as well as to build robust classifiers. For ℓ_∞ attacks, when they were used for evaluation, they are performed with 20 steps; for training robust classifiers, these attacks were performed with 10 steps at each epoch of adversarial training. Similarly, for ℓ_2 attacks, they were performed with 100 steps for evaluation, and 50 steps for adversarial training. We used the semantic segmentation masks on the Segment-6 dataset and used fixation prediction to identify foreground and backround on STL-10 and ImageNet-10.

D ADVERSARIAL TRAINING USING ℓ_2 NORM ATTACKS ON IMAGENET-10

Transferability of Adversarial Examples. Here, we measure the *transferability* of adversarial examples among different classification models. To do this, we first produced adversarial examples by using ℓ_2 PGD attack or dual-perturbation attack on a source model. Then, we used these examples to evaluate the performance of an independent target model, where a higher prediction accuracy means weaker transferability. The results are presented in Figure 1. The first observation is that dual-perturbation attacks exhibit significantly better transferability than the conventional PGD attacks (transferability is up to 40% better for dual-perturbation attacks). Second, we can observe that when *AT-Dual* is used as the target (i.e., defending by adversarial training with dual-perturbation examples), these are typically resistant to adversarial examples generated against either the clean model, or against *AT-PGD*. This observation obtains even when we use PGD to generate adversarial examples.

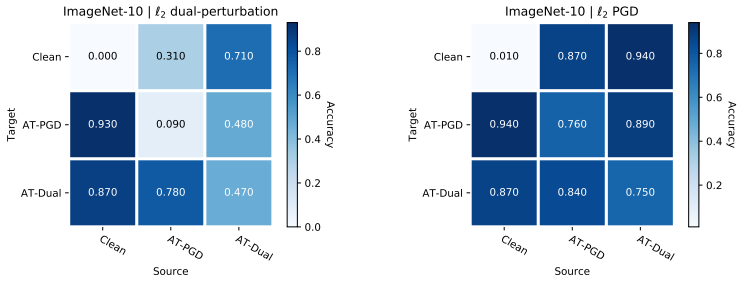

Figure 1: Robustness against adversarial examples transferred from other models on ImageNet-10. Left: ℓ_2 dual-perturbation attacks performed by using $\{\epsilon_F, \epsilon_B, \lambda\} = \{2.0, 20.0, 1.0\}$ on different source models. Right: ℓ_2 PGD attacks with $\epsilon = 2.0$ on different source models.

E ADVERSARILA TRAINING USING ℓ_2 NORM ATTACKS ON STL-10

Here, we present experimental results of the robustness of classifiers that use adversarial training with ℓ_2 norm attacks on STL-10. Specifically, we trained AT-PGD using ℓ_2 PGD attack with $\epsilon = 1.0$, and AT-Dual by using ℓ_2 dual-perturbation attack with $\{\epsilon_F, \epsilon_B, \lambda\} = \{1.0, 5.0, 0.0\}$. The results are shown in Figure 2, 3, 4, and 5.

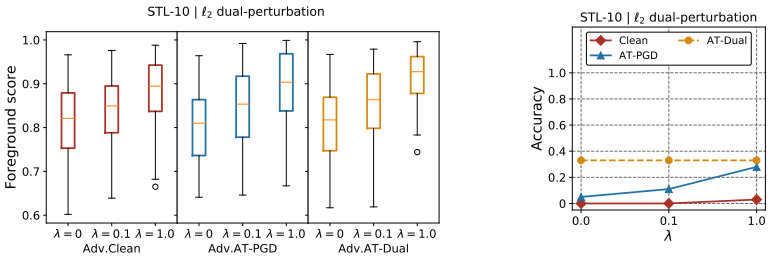

Figure 2: Saliency analysis. The ℓ_2 dual-perturbation attacks are performed by using $\{\epsilon_F, \epsilon_B\} = \{1.0, 5.0\}$, and a variety of λ displayed in the figure. Left: foreground scores of dual-perturbation examples in response to different classifiers. Right: accuracy of classifiers on dual-perturbation examples with salience control.

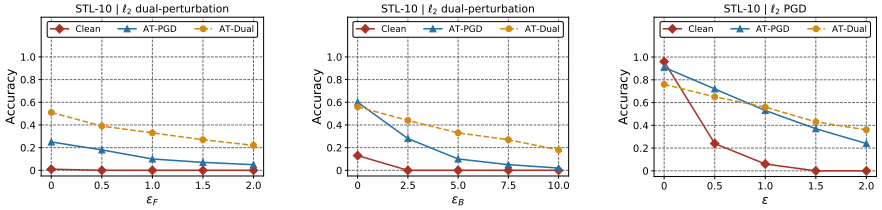

Figure 3: Robustness to white-box ℓ_2 attacks on STL-10. Left: ℓ_2 dual-perturbation attacks with different foreground distortions. ϵ_B is fixed to be 5.0 and $\lambda = 0.1$. Middle: ℓ_2 dual-perturbation attacks with different background distortions. ϵ_F is fixed to be 1.0 and $\lambda = 0.1$. Right: ℓ_2 PGD attacks.

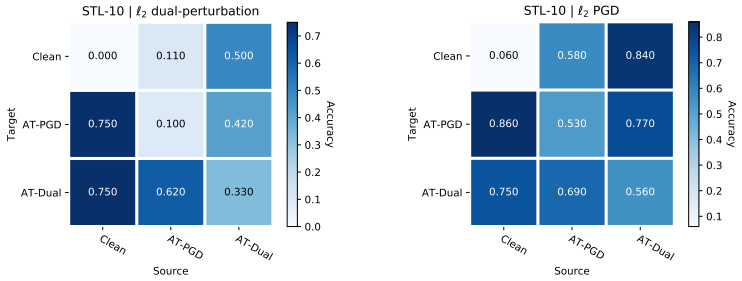

Figure 4: Robustness against adversarial examples transferred from other models on STL-10. Left: ℓ_2 dual-perturbation attacks performed by using $\{\epsilon_F, \epsilon_B, \lambda\} = \{1.0, 5.0, 0.1\}$ on different source models. Right: ℓ_2 PGD attacks with $\epsilon = 1.0$ on different source models.

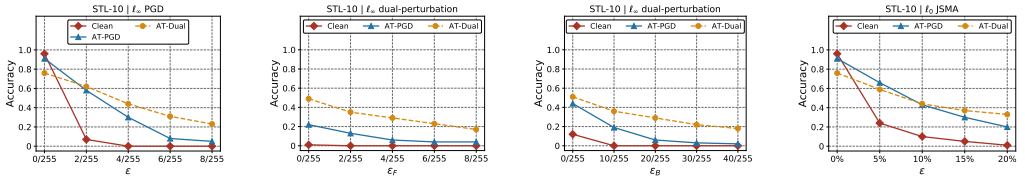

Figure 5: Robustness to additional white-box attacks on STL-10. Left: 20 steps of ℓ_∞ PGD attacks. Middle left: 20 steps of ℓ_∞ dual-perturbation attacks with different foreground distortions. ϵ_B is fixed to be 20/255 and $\lambda = 0.1$. Middle right: 20 steps of ℓ_∞ dual-perturbation attacks with different background distortions. ϵ_F is fixed to be 4/255 and $\lambda = 0.1$. Right: ℓ_0 JSMA attacks.

F Adversarial Training Using ℓ_2 Norm Attacks on Segment-6

Now, we present experimental results of the robustness of classifiers that use adversarial training with ℓ_2 norm attacks on Segment-6. Since DeepGaze II only work on images with more than 35×35 pixels, we are unable to use DeepGaze II to compute the *foreground score (FS)* for Segment-6. Hence, in the following experiment on this dataset, we omit the salience term in the optimization problem of Equation 3 and 4 in the main body of the paper. Specifically, we trained AT-PGD using ℓ_2 PGD attack with $\epsilon = 0.5$, and AT-Dual by using ℓ_2 dual-perturbation attack with $\{\epsilon_F, \epsilon_B\} = \{0.5, 2.5\}$. The results are shown in Figure 6, 7, and 8.

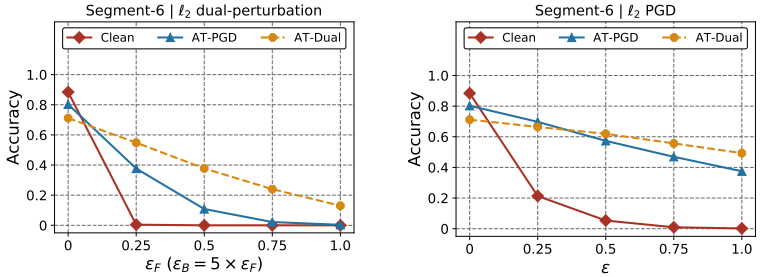

Figure 6: Robustness to white-box ℓ_2 attacks on Segment-6. Left: ℓ_2 dual-perturbation attacks with different foreground and background distortions. Right: ℓ_2 PGD attacks.

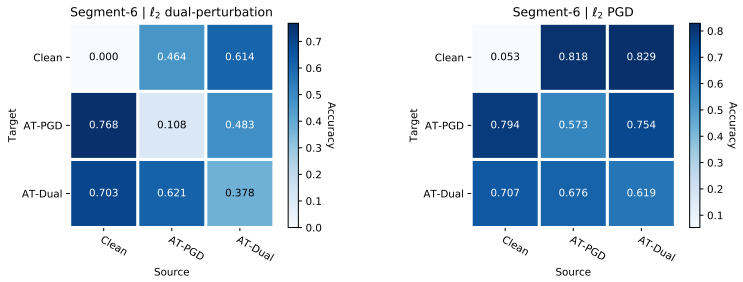

Figure 7: Robustness against adversarial examples transferred from other models on Segment-6. Left: ℓ_2 dual-perturbation attacks performed by using $\{\epsilon_F, \epsilon_B\} = \{0.5, 2.5\}$ on different source models. Right: ℓ_2 PGD attacks with $\epsilon = 0.5$ on different source models.

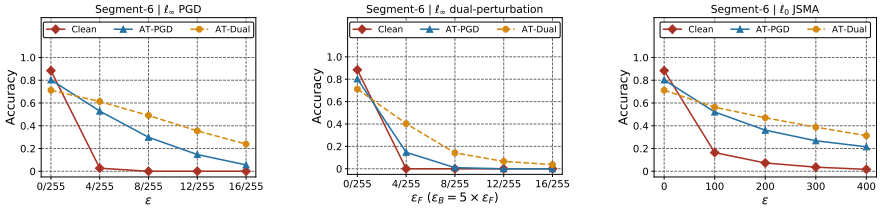

Figure 8: Robustness to additional white-box attacks on Segment-6. Left: 20 steps of ℓ_∞ PGD attacks. Middle: 20 steps of ℓ_∞ dual-perturbation attacks with different foreground and background distortions. Right: ℓ_0 JSMA attacks.

G Adversarial Training Using ℓ_∞ Norm Attacks on ImageNet-10

Next, we present experimental results of the robustness of classifiers that use adversarial training with ℓ_∞ norm attacks on ImageNet-10. Specifically, we trained AT-PGD using ℓ_∞ PGD attack with $\epsilon = 4/255$, and AT-Dual by using ℓ_∞ dual-perturbation attack with $\{\epsilon_F, \epsilon_B, \lambda\} = \{4/255, 20/255, 0.0\}$. The results are shown in Figure 9, 10, 11, and 12.

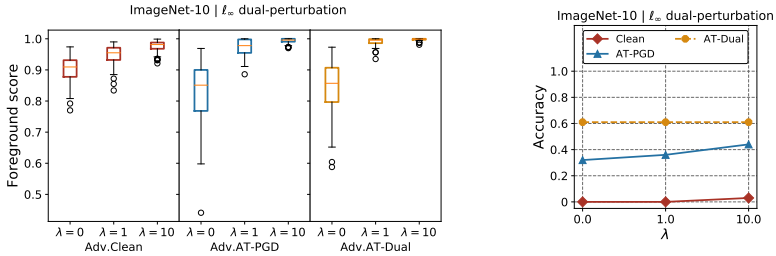

Figure 9: Saliency analysis. The ℓ_∞ dual-perturbation attacks are performed by using $\{\epsilon_F, \epsilon_B\} = \{4/255, 20/255\}$, and a variety of λ displayed in the figure. Left: foreground scores of dual-perturbation examples in response to different classifiers. Right: accuracy of classifiers on dual-perturbation examples with salience control.

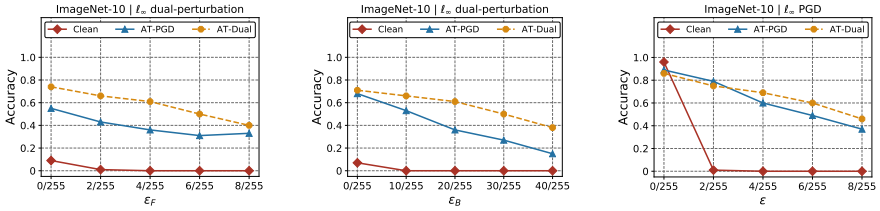

Figure 10: Robustness to white-box ℓ_∞ attacks on ImageNet-10. Left: ℓ_∞ dual-perturbation attacks with different foreground distortions. ϵ_B is fixed to be 20/255 and $\lambda = 1.0$. Middle: ℓ_∞ dual-perturbation attacks with different background distortions. ϵ_F is fixed to be 4/255 and $\lambda = 1.0$. Right: ℓ_∞ PGD attacks.

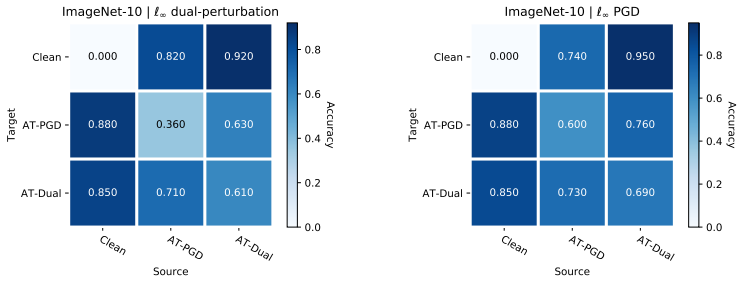

Figure 11: Robustness against adversarial examples transferred from other models on ImageNet-10. Left: ℓ_∞ dual-perturbation attacks performed by using $\{\epsilon_F, \epsilon_B, \lambda\} = \{4/255, 20/255, 1.0\}$ on different source models. Right: ℓ_∞ PGD attacks with $\epsilon = 4/255$ on different source models.

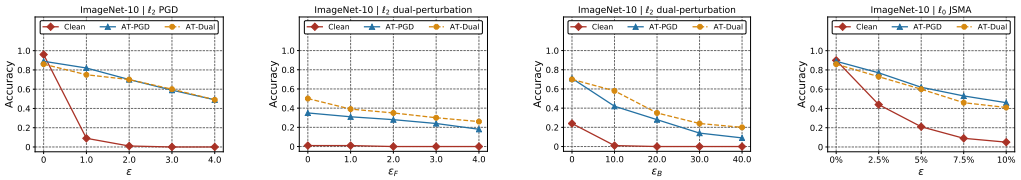

Figure 12: Robustness to additional white-box attacks on ImageNet-10. Left: 100 steps of ℓ_2 PGD attacks. Middle left: 100 steps of ℓ_2 dual-perturbation attacks with different foreground distortions. ϵ_B is fixed to be 2.0 and $\lambda = 1.0$. Middle right: 100 steps of ℓ_2 dual-perturbation attacks with different background distortions. ϵ_F is fixed to be 20.0 and $\lambda = 1.0$. Right: ℓ_0 JSMA attacks.

H Adversarial Training Using ℓ_∞ Norm Attacks on STL-10

Now, we present experimental results of the robustness of classifiers that use adversarial training with ℓ_∞ norm attacks on STL-10. Specifically, we trained AT-PGD using ℓ_∞ PGD attack with $\epsilon = 4/255$, and AT-Dual by using ℓ_∞ dual-perturbation attack with $\{\epsilon_F, \epsilon_B, \lambda\} = \{4/255, 20/255, 0.0\}$. The results are shown in Figure 13, 14, 15, and 16.

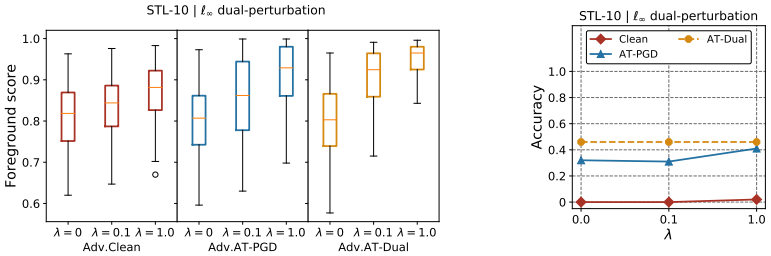

Figure 13: Saliency analysis. The ℓ_∞ dual-perturbation attacks are performed by using $\{\epsilon_F, \epsilon_B\} = \{4/255, 20/255\}$, and a variety of λ displayed in the figure. Left: foreground scores of dual-perturbation examples in response to different classifiers. Right: accuracy of classifiers on dual-perturbation examples with salience control.

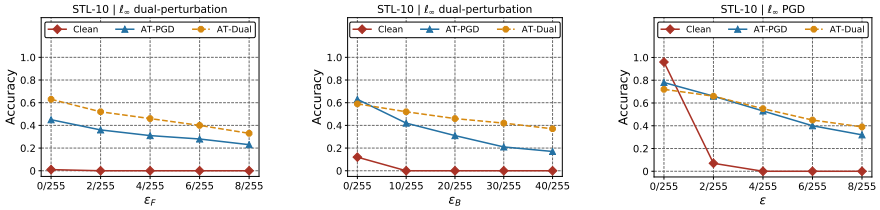

Figure 14: Robustness to white-box ℓ_∞ attacks on STL-10. Left: ℓ_∞ dual-perturbation attacks with different foreground distortions. ϵ_B is fixed to be $20/255$ and $\lambda = 0.1$. Middle: ℓ_∞ dual-perturbation attacks with different background distortions. ϵ_F is fixed to be $4/255$ and $\lambda = 0.1$. Right: ℓ_∞ PGD attacks.

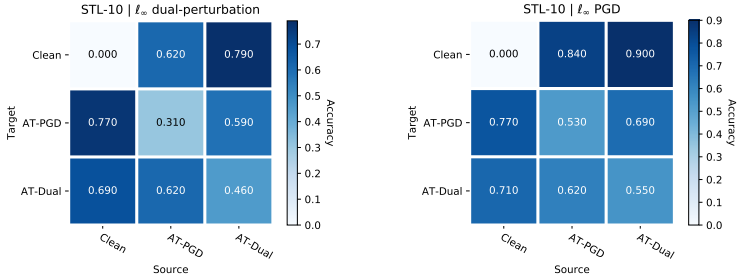

Figure 15: Robustness against adversarial examples transferred from other models on STL-10. Left: ℓ_∞ dual-perturbation attacks performed by using $\{\epsilon_F, \epsilon_B, \lambda\} = \{4/255, 20/255, 1.0\}$ on different source models. Right: ℓ_∞ PGD attacks with $\epsilon = 4/255$ on different source models.

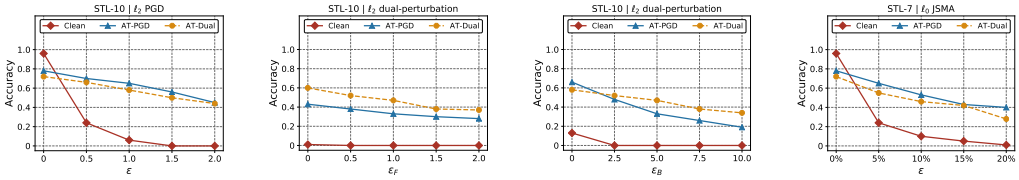

Figure 16: Robustness to additional white-box attacks on STL-10. Left: 100 steps of ℓ_2 PGD attacks. Middle left: 100 steps of ℓ_2 dual-perturbation attacks with different foreground distortions. ϵ_B is fixed to be 5.0 and $\lambda = 0.1$. Middle right: 100 steps of ℓ_2 dual-perturbation attacks with different background distortions. ϵ_F is fixed to be 1.0 and $\lambda = 0.1$. Right: ℓ_0 JSMA attacks.

I Adversarial Training Using ℓ_∞ Norm Attacks on Segment-6

Finally, we present experimental results of the robustness of classifiers that use adversarial training with ℓ_∞ norm attacks on Segment-6. We trained AT-PGD using ℓ_∞ PGD attack with $\epsilon = 8/255$, and AT-Dual by using ℓ_∞ dual-perturbation attack with $\{\epsilon_F, \epsilon_B\} = \{8/255, 40/255\}$. The results are shown in Figure 17, 18, and 19.

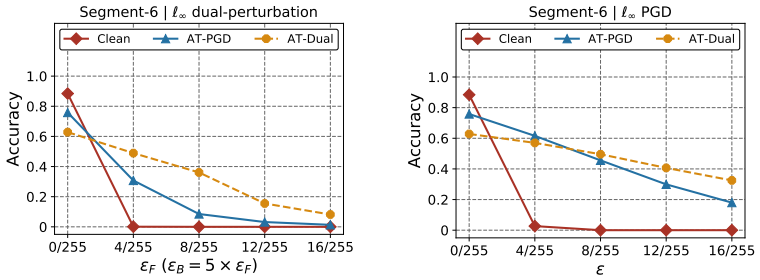

Figure 17: Robustness to white-box ℓ_∞ attacks on Segment-6. Left: ℓ_∞ dual-perturbation attacks with different foreground and background distortions. Right: ℓ_∞ PGD attacks.

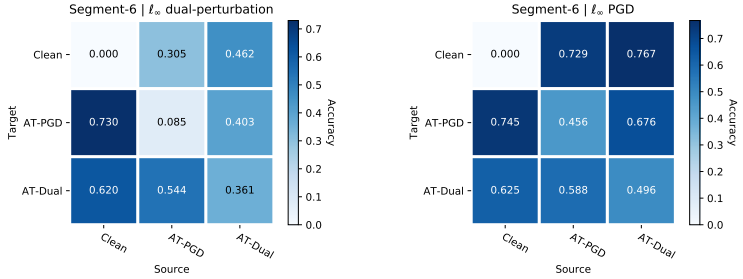

Figure 18: Robustness against adversarial examples transferred from other models on Segment-6. Left: ℓ_∞ dual-perturbation attacks performed by using $\{\epsilon_F, \epsilon_B\} = \{8/255, 40/255\}$ on different source models. Right: ℓ_∞ PGD attacks with $\epsilon = 8/255$ on different source models.

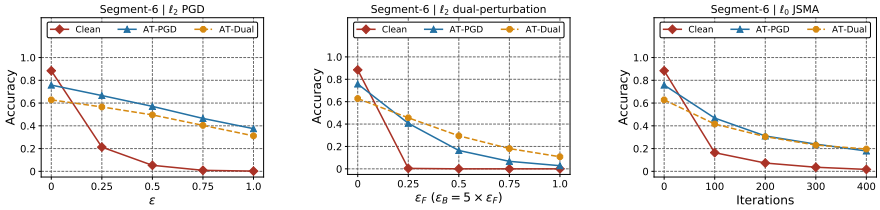

Figure 19: Robustness to additional white-box attacks on Segment-6. Left: 100 steps of ℓ_2 PGD attacks. Middle: 100 steps of ℓ_2 dual-perturbation attacks with different foreground and background distortions. Right: ℓ_0 JSMA attacks.

J Attacking Randomzied Classifiers

In addition to *deterministic classifiers* that make a deterministic prediction for a test sample, our proposed attack can be adapted to *stochastic classifiers* that apply randomization at training and prediction time. For example, for classifiers using *randomized smoothing*, we can refine Equation 3 in the main body of the paper as follows:

$$\max_{\substack{||\boldsymbol{\delta} \circ \mathcal{F}(\boldsymbol{x})||_p \leq \epsilon_F, \\ ||\boldsymbol{\delta} \circ \mathcal{B}(\boldsymbol{x})||_p \leq \epsilon_B}} \mathbb{E}_{\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})} [\mathcal{L}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta} + \boldsymbol{\eta}), y\right) + \lambda \cdot \mathcal{S}\left(\boldsymbol{x} + \boldsymbol{\delta} + \boldsymbol{\eta}\right)], \tag{5}$$

where σ^2 is the variance of the Gaussian data augmentation in randomized smoothing. [3] The optimization problem in Equation 5 can be solved by the same approach used for deterministic

[3] Note that the Gaussian perturbations are only used to compute the expection of loss and are not in the resulting adversarial examples.

classifiers, with the following modification on Equation 3 at the second step in Section A:

$$\begin{cases} g_F = \mathcal{G}(\mathcal{F}(\boldsymbol{x}) \circ \nabla_{\boldsymbol{\delta}^{(k)}} \mathbb{E}_{\boldsymbol{\eta}}[\mathcal{L}(h_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}^{(k)} + \boldsymbol{\eta}), y) + \lambda \cdot \mathcal{S}\left(\boldsymbol{x} + \boldsymbol{\delta}^{(k)} + \boldsymbol{\eta}\right)]) \\ g_B = \mathcal{G}(\mathcal{B}(\boldsymbol{x}) \circ \nabla_{\boldsymbol{\delta}^{(k)}} \mathbb{E}_{\boldsymbol{\eta}}[\mathcal{L}(h_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}^{(k)} + \boldsymbol{\eta}), y) + \lambda \cdot \mathcal{S}\left(\boldsymbol{x} + \boldsymbol{\delta}^{(k)} + \boldsymbol{\eta}\right)]) \end{cases} . \tag{6}$$

J.1 Variance in Gaussian Data Augmentation

Table 4 and 5 show the effectiveness of *Randomized Smoothing (RS)* against the proposed dual-perturbation attack. Here, we use different variances in Gaussian data augmentation of *RS*, and fix the number of noise-corrupted copies at prediction time, n to be 100. It can be seen that *RS* is generally fragile to the dual-perturbation attacks that are adapted to randomized classifiers. Moreover, increasing σ, the variance used in Gaussian data augmentation can only marginally improve adversarial robustness to dual-perturbation attacks while significantly decrease accuracy on non-adversarial data.

Dataset	**Defense approach**	**Attack Strength** ($\epsilon_B = 5 \times \epsilon_F$)				
		$\epsilon_F = 0/255$	$\epsilon_F = 4/255$	$\epsilon_F = 8/255$	$\epsilon_F = 12/255$	$\epsilon_F = 1$
Segment-6	RS, $\sigma = 0.25$	71.4%	9.6%	0.4%	0.1%	0.0%
	RS, $\sigma = 0.5$	61.7%	13.7%	1.9%	0.6%	0.2%
	RS, $\sigma = 1$	47.7%	15.6%	2.8%	0.4%	0.2%

Table 4: Robustness of *RS* against ℓ_∞ dual-perturbation attacks.

Defense approach	**Attack Strength** ($\epsilon_B = 5 \times \epsilon_F$)				
	$\epsilon_F = 0$	$\epsilon_F = 0.25$	$\epsilon_F = 0.5$	$\epsilon_F = 0.75$	$\epsilon_F = 1$
RS, $\sigma = 0.25$	71.4%	29.7%	6.7%	0.9%	0.1%
RS, $\sigma = 0.5$	61.7%	31.6%	11.8%	3.1%	1.3%
RS, $\sigma = 1$	47.7%	28.2%	14.4%	6.0%	1.5%

Table 5: Robustness of *RS* against ℓ_2 dual-perturbation attacks on Segment-6.

J.2 Number of Samples with Gaussian Noise at Prediction Time

It has been observed that *Randomized Smoothing (RS)* can be computationally inefficient at prediction time as it uses a large number of noise-corrupted copies for each test sample at prediction time. It is natural to ask whether the prediction time of *RS* can be reduced without significantly sacrificing adversarial robustness in practice. We answer this question by studying the effectiveness of *RS* with different n, the numbers of noise-corrupted copies at prediction time. Specifically, we fix $\sigma = 0.5$ and set n to be 1, 25, and 100. Note that when $n = 1$, there is no two-sided hypothesis test for prediction; thus, no abstentions are obtained.

Here we use ℓ_∞ dual-perturbation attacks on *RS* for demonstration purposes. The results are shown in Table 6. It can be seen that when $n = 25$, the accuracy on both adversarial and non-adversarial data can drop by up to 10% compared to *RS* using $n = 100$. The reason is that under a small n, the prediction appears more likely to abstain. Interestingly, when $n = 1$, the accuracy can be marginally improved compared to $n = 100$, with the prediction time being reduced by 99%. This indicates that in practice, we would not lose accuracy without using the two-sided hypothesis test at prediction time.

Dataset	**Defense approach**	**Attack Strength** ($\epsilon_B = 5 \times \epsilon_F$)				
		$\epsilon_F = 0/255$	$\epsilon_F = 4/255$	$\epsilon_F = 8/255$	$\epsilon_F = 12/255$	$\epsilon_F = 1$
Segment-6	RS, $n = 1$	66.0%	19.8%	3.2%	0.8%	0.3%
	RS, $n = 25$	49.4%	9.1%	1.3%	0.5%	0.0%
	RS, $n = 100$	61.7%	13.7%	1.9%	0.6%	0.2%

Table 6: Robustness of *RS* against ℓ_∞ dual-perturbation attacks under different numbers of noise-corrupted copies at prediction time.

K VISUALIZATION OF LOSS GRADIENT

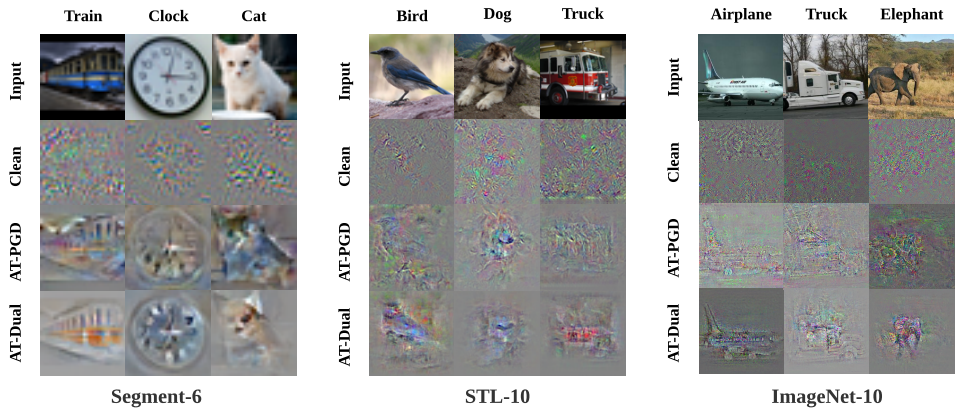

Figure 20: Visualization of loss gradient of different classifiers with respect to pixels of *non-adversarial* inputs. AT-PGD and AT-Dual were obtained using adversarial training with corresponding ℓ_2 norm attacks.

L EXAMPLES OF DUAL-PERTURBATION ATTACKS

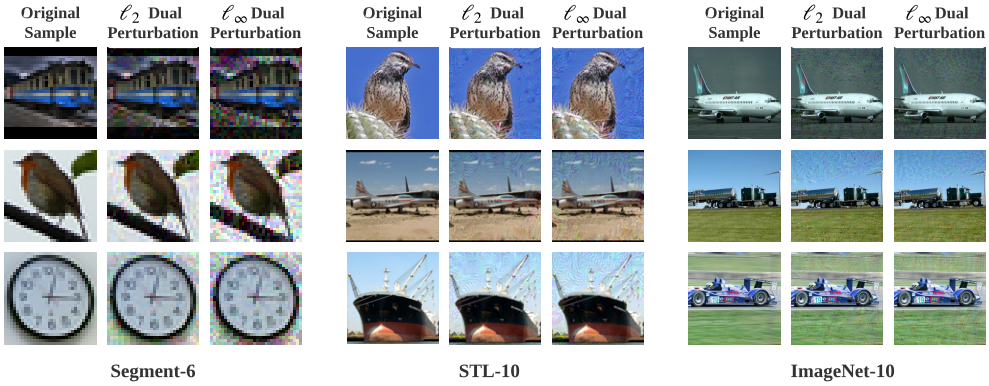

Figure 21: Dual-perturbation attacks. Adversarial examples are produced in response to the *Clean* model for each dataset.

REFERENCES

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*, pp. 770–778, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.