SUPPLEMENTARY MATERIAL FOR EXPLOITING SAFE SPOTS IN NEURAL NETWORKS FOR PREEMPTIVE ROBUSTNESS AND OUT-OF-DISTRIBUTION DETECTION

Anonymous authors
Paper under double-blind review

A SAFE SPOT SEARCH ALGORITHM

A.1 PROOF OF LEMMA 1

Since $x_a \in B_\epsilon(x_s)$, we have

$$\ell(x_s, c(x_o)) \leq \sup_{x_a} \ \ell(x_a, c(x_o)) = \tilde{h}(x_s) \leq -\log(0.5). \tag{1}$$

Let $C(x_s)$ be the softmax probability of x_s. Equation (1) implies $C(x_s)_{c(x_o)} \geq 0.5$, *i.e.*, $c(x_s) = c(x_o)$. Finally, we have

$$\begin{aligned} h(x_s) &= \ell(x_s, c(x_o)) + \sup_{x_a} \ \ell(x_a, c(x_s)) \\ &= \ell(x_s, c(x_o)) + \sup_{x_a} \ \ell(x_a, c(x_o)) && (\because c(x_s) = c(x_o)) \\ &\leq 2 \sup_{x_a} \ \ell(x_a, c(x_o)) && (\because \ell(x_s, c(x_o)) \leq \sup_{x_a} \ell(x_a, c(x_o))) \\ &= 2\tilde{h}(x_s). \end{aligned}$$

□

B COMPUTING UPDATE GRADIENT WITH SECOND-ORDER DERIVATIVES

B.1 PROOF OF LEMMA 2

It is enough to compute the Jacobian of $\frac{g}{\|g\|_2}$. By the quotient rule, we have

$$\begin{aligned} \frac{\partial}{\partial x}\left(\frac{g}{\|g\|_2}\right) &= \frac{\|g\|_2 \cdot H - g \cdot (\nabla_x \|g\|_2)^\intercal}{\|g\|_2^2} \\ &= \frac{H}{\|g\|_2} - \frac{g \cdot (\nabla_x \|g\|_2)^\intercal}{\|g\|_2^2}. \end{aligned} \tag{2}$$

Now, we compute $\nabla_x \|g\|_2$. Since $\|g\|_2^2 = \langle g, g \rangle$, we have

$$2\|g\|_2 \cdot \nabla_x \|g\|_2 = \nabla_x \langle g, g \rangle = 2H \cdot g. \tag{3}$$

Plugging Equation (3) into Equation (2), we have

$$\begin{aligned} \frac{\partial}{\partial x}\left(\frac{g}{\|g\|_2}\right) &= \frac{H}{\|g\|_2} - \frac{g \cdot (\nabla_x \|g\|_2)^\intercal}{\|g\|_2^2} \\ &= \frac{H}{\|g\|_2} - \frac{g \cdot g^\intercal \cdot H^\intercal}{\|g\|_2^3} \\ &= \left(I - \left(\frac{g}{\|g\|_2}\right)\left(\frac{g}{\|g\|_2}\right)^\intercal\right)\frac{H}{\|g\|_2}, \end{aligned}$$

since H is symmetric, and it completes the proof. □

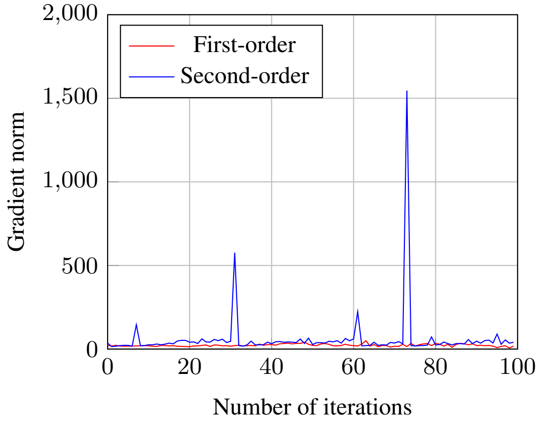


Figure 1: The norm of update gradient at each iteration during safe spot search with the first-order approximation and with the exact computation using second-order derivatives in the CIFAR-10 test set. The approximate gradient update finds a safe spot for the image, while the exact gradient update does not.

B.2 Proof of Proposition 1

Note that $P = I - \left(\frac{g}{\|g\|_2}\right)\left(\frac{g}{\|g\|_2}\right)^\intercal$ is a projection map onto a hyperplane whose normal vector is $\frac{g}{\|g\|_2}$. Since a projection map is a contraction map, we have

$$\begin{aligned}
\left\|\frac{\partial f}{\partial x}^\intercal a\right\|_2 &= \left\|\left(I+\alpha\cdot\left(I-\left(\frac{g}{\|g\|_2}\right)\left(\frac{g}{\|g\|_2}\right)^\intercal\right)\frac{H}{\|g\|_2}\right)^\intercal a\right\|_2 && (\because \text{By Lemma 2}) \\
&= \left\|a+\alpha\cdot\frac{H^\intercal}{\|g\|_2}\left(I-\left(\frac{g}{\|g\|_2}\right)\left(\frac{g}{\|g\|_2}\right)^\intercal\right)^\intercal a\right\|_2 \\
&= \|a\|_2 + \left\|\alpha\cdot\frac{H^\intercal}{\|g\|_2}\left(I-\left(\frac{g}{\|g\|_2}\right)\left(\frac{g}{\|g\|_2}\right)^\intercal\right)^\intercal a\right\|_2 && (\because \text{By the triangular inequality}) \\
&\le \|a\|_2 + \alpha\cdot\frac{\|H^\intercal\|_2}{\|g\|_2}\cdot\left\|\left(I-\left(\frac{g}{\|g\|_2}\right)\left(\frac{g}{\|g\|_2}\right)^\intercal\right)^\intercal\right\|_2\|a\|_2 && (\because \|AB\|_2\le\|A\|_2\|B\|_2) \\
&= \|a\|_2 + \alpha\cdot\frac{\sigma}{\|g\|_2}\cdot\left\|\left(I-\left(\frac{g}{\|g\|_2}\right)\left(\frac{g}{\|g\|_2}\right)^\intercal\right)^\intercal\right\|_2\|a\|_2 && (\because H \text{ is symmetric}) \\
&= \|a\|_2 + \alpha\cdot\frac{\sigma}{\|g\|_2}\cdot\|a\|_2\,, && (\because P \text{ is a contraction map})
\end{aligned}$$

which completes the proof. □

B.3 Experiments on second-order derivatives

Here, we evaluate the performance of the safe spot search with exact gradient computation using second-order derivatives, compared to the approximate gradient update. We experiment on the naturally trained CIFAR-10 model. We consider a ℓ_2 threat model with $\epsilon = 0.5$. The experimental settings are the same as in Section 4.1, except that we randomly sample 1,000 images from the test set. We tune the learning rate β within a range of $\{0.01, 0.005, 0.001, 0.0005\}$ and choose the best $\beta = 0.005$. Table 1 shows the result. We observe that using second-order derivatives rather degrades the performance of the safe spot search.

To examine whether the exact gradient computation exhibits exploding gradient behavior, we also measure the ℓ_2 norm of update gradient at each step for both the methods. Figure 1 shows that the safe spot search with the exact gradient computation results in unstable update gradients, in contrast to the approximate gradient update.

Method	Clean accuracy	Adv. accuracy
First-order	96.3%	62.0%
Second-order	95.8%	55.2%

Table 1: Classification accuracy under ℓ_2 threat with $\epsilon = 0.5$ on a naturally trained CIFAR-10 model.

C Implementation details

We fix MAXITER $= 100$, $T = 20$, $\alpha = \epsilon/4$ on all settings. We set $N = 5$ on naturally trained classifiers to stabilize the safe spot update procedure, while $N = 1$ is sufficient for robust classifiers. On ℓ_∞ threat model, we transform x_o-centered δ-ball space to $\mathbb{R}^n$ via tanh transformation and use RMSProp optimizer (Hinton et al., 2012) to update safe spots. On ℓ_2 threat model, we use projected gradient descent method as default.

C.1 Experiments on CIFAR-10 and ImageNet

CIFAR-10 We train the models for 200 epochs with an initial learning rate of 0.1, decayed with a factor of 0.1 on epoch 100 and 150. We use SGD optimizer with weight decay $2E - 4$, momentum 0.9, and batch size 128. The ADV model (Madry et al., 2017) is trained with 10-step PGD with a step size of $\epsilon/4$.

We take full 10,000 images from CIFAR-10 test set and generate safe spots. On ℓ_∞ threat model with $\epsilon = 8/255$, we set the learning rate $\beta = 0.1$, and on ℓ_2 threat model with $\epsilon = 3.0$, we set $\beta = 0.001$. Each β is chosen by tuning between range of $\{1.0, 0.5, 0.1, 0.05, 0.01, 0.001\}$ and $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$.

ImageNet We resize and crop the test images to 224×224. For the natural and ADV models, we use pre-trained models provided by Pytorch and Engstrom et al. (2019). We additionally train Fast model from Wong et al. (2020), a naturally trained model applying the same training techniques, and the safe spot-aware model S-FGSM+Fast. For these Fast-type models, we follow the hyper-parameter settings from the original paper, except that we omit phase 3 and double the number of epochs in phase 2 since phase 3 uses 288×288 sized images. On S-FGSM+Fast, we tune the step size of S-FGSM to $\epsilon/2$.

We randomly sample 10,000 images from the test set and generate safe spots. We consider two ℓ_∞ threat models with $\epsilon \in \{4/255, 8/255\}$, and one ℓ_2 threat model with $\epsilon = 3.0$. On the ℓ_∞ threat models, β is set to 0.05 and 0.1 respectively, where each β is chosen by tuning between range of $\{1.0, 0.5, 0.1, 0.05, 0.01\}$. On the ℓ_2 threat model, we set β to 0.01 for the natural model and 0.1 for the ADV model. We tuned β between range of $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$.

C.2 Experiments on randomized smoothing

We use the same network structures from Section 4.1 and 4.2. The Gaussian model is trained with Gaussian noise drawn from $\mathcal{N}(0, \sigma^2 I)$. For the Gaussian model on ImageNet, we use the pre-trained model provided by Cohen et al. (2019).

To evaluate the empirical robustness of smoothed classifiers, we set the noise level σ to 0.05 for the natural model, and 0.1 for the Gaussian model on CIFAR-10. For ImageNet, σ is set to 0.12 for the natural model and 0.25 for the Gaussian model. The PGD settings are the same as in the experiments for base classifiers, except that we set random starts $N = 1$ on both the classifiers. We take 5 Gaussian samples for safe image generation and 50 Gaussian samples for evaluation.

To measure the certified robustness of smoothed classifiers, we set noise levels to be larger than those used in evaluating the empirical robustness since a larger σ leads to larger certified radii. We set σ to 0.25 and 1.0 for the Gaussian models in CIFAR-10 and ImageNet, respectively.

For class prediction, we take 50 Gaussian samples and choose the most probable class by a majority vote. In the case of the natural model, however, the standard accuracy can be reduced when evaluating with randomized smoothing since it is not trained to be robust against Gaussian noise. Therefore, given an original image x_o, we first predict the class of the image without randomized smoothing, then create a safe spot x_s from x_o with the predicted class, and evaluate x_s with randomized smoothing. For certification, we take 100,000 Gaussian samples.

For CIFAR-10, the learning rate β is set to 0.001 for the natural model and 0.005 for the Gaussian model. We tune the learning rate within a range of $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$. For ImageNet, the learning rate β is set to 0.05 for the natural model and 0.1 for the Gaussian model. we tune the learning rate within a range of $\{1.0, 0.5, 0.1, 0.05, 0.01\}$.

C.3 Safe spot out-of-distribution detection histogram

We use the S-PGD+ADV model trained with CIFAR-10 dataset and use LSUN as the out-of-distribution dataset. For all images, we regard the prediction labels as the ground truth labels. In the left plot, we measure the original image's cross-entropy loss value. On the right plot, we first generate a safe spot from an image, perturb it with a 20-step untargeted PGD attack, and then measure the cross-entropy loss. The dotted line shows the detection threshold when the true positive rate is set to 95%, where we treat out-of-distribution samples as positive. The histograms show that the in-distribution examples are much more likely to have safe spots in the vicinity than the out-of-distribution samples. In fact, the false positive rate at 95% true positive rate (**FPR95**), where the lower value indicates a better separation between in-distribution and out-of-distribution samples, is lower when using perturbed safe spots. Concretely, the detection algorithm using the original images obtains FPR95 of 50.32% while using the perturbed safe spots obtains FPR95 of 32.16%.

Figure 2 shows the loss values histograms when using other out-of-distribution datasets, SVHN, CIFAR-100, and TinyImageNet. We observe that the perturbed safe spot's loss value is a better metric to detect out-of-distribution samples than the original image's loss value for all the datasets.

D More safe spot evaluations

D.1 ImageNet results under ℓ_2 threat

Table 2 shows the ImageNet results under ℓ_2 threat model with $\epsilon = 3.0$. Similar to the results on the ℓ_∞ threat model, our algorithm can find safe spots for most correctly classified images for the robust classifier.

Model	**Method**			
	None	**S-FGSM**	**S-PGD**	**S-Full**
Natural	75.63%/00.11%	75.14%/01.13%	75.63%/01.31%	75.63%/10.14%
ADV	56.73%/33.19%	56.73%/54.94%	56.73%/55.28%	56.73%/**56.20%**

Table 2: Classification accuracy under ℓ_2 threat with $\epsilon = 3.0$ on ImageNet. (clean acc./adv acc.)

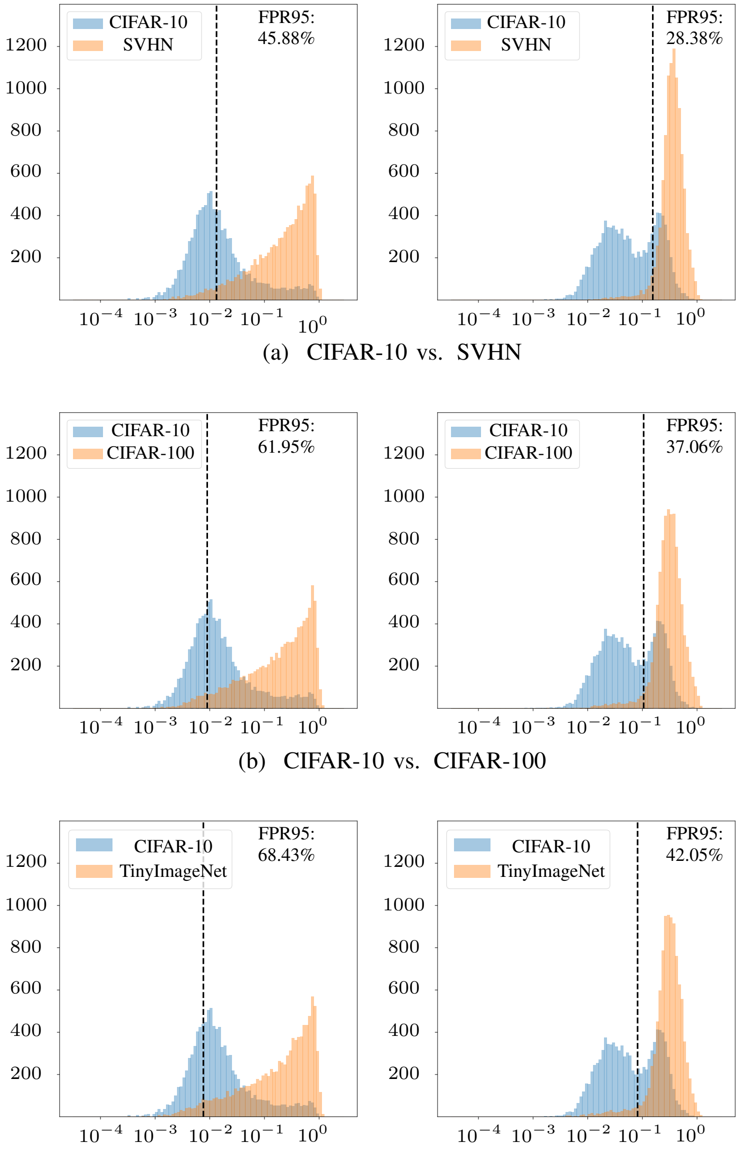
D.2 Certified robustness

Table 3 shows the certified robustness results on the smoothed Gaussian model. We find that our algorithm also improves the certified robustness on both the CIFAR-10 and ImageNet datasets.

Model	**Method**	
	None	**S-Full**
Gaussian + Smoothing	82.84/55.58	84.72/**77.95**

Model	**Method**	
	None	**S-Full**
Gaussian + Smoothing	47.02/12.68	52.66/**27.89**

Table 3: Certified robustness of randomized smoothed networks under ℓ_2 threat with $\epsilon = 0.5$ on CIFAR-10 (left) and with $\epsilon = 3.0$ on ImageNet (right). (clean acc./cert acc.)



(a) CIFAR-10 vs. SVHN

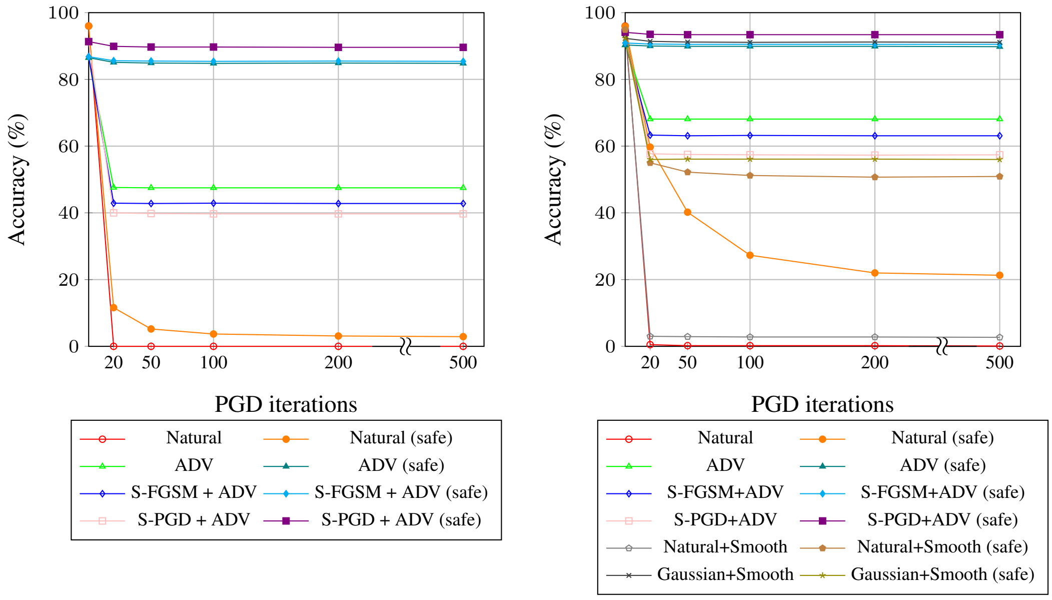
(b) CIFAR-10 vs. CIFAR-100

(c) CIFAR-10 vs. TinyImageNet

Figure 2: Histograms for the loss values of images $\ell(x_o, c(x_o))$ (left) and the loss values of the perturbed safe spot solution $\sup_{x_a^* \in B_\epsilon(x_s^*)} \ell(x_a^*, c(x_o))$ (right). A safe spot-aware adversarially trained model without fine-tuning is used as the classifier. The dotted lines are where the false positive rate is 95%.

D.3 Experiments with higher PGD iterations

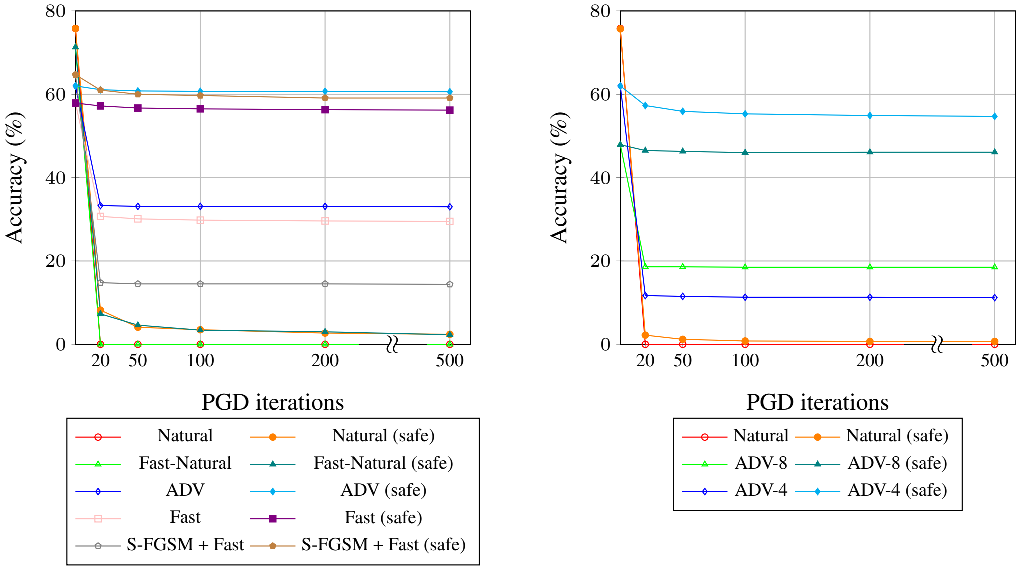
We additionally provide adversarial accuracy results under higher PGD iterations for all threat models considered in Section 4.1 and 4.2. We used the same networks from the main paper for all experiments. PGD step size was set to $(5 \cdot \epsilon/\text{iterations})$ except for when iterations $= 10$, where step size was set to $(2.5 \cdot \epsilon/\text{iterations})$. For these experiments, we used 10,000 randomly selected test set images for CIFAR-10 and ImageNet each. Figure 3 to Figure 5 show the results.



(a)

(b)

Figure 3: Evaluation with higher PGD iterations, under (a) ℓ_∞ threat with $\epsilon = 8/255$ and (b) ℓ_2 threat with $\epsilon = 0.5$ on CIFAR-10. (safe) denotes using safe spot images in place of original images.



(a)

(b)

Figure 4: Evaluation with higher PGD iterations, under ℓ_∞ threat with (a) $\epsilon = 4/255$ and (b) $\epsilon = 8/255$ on ImageNet. (safe) denotes using safe spot images in place of original images. Fast-Natural indicates the natural model trained with efficient training techniques from Fast. ADV-8 and ADV-4 each indicate ADV models trained under $\epsilon = 8/255$ and $\epsilon = 4/255$.

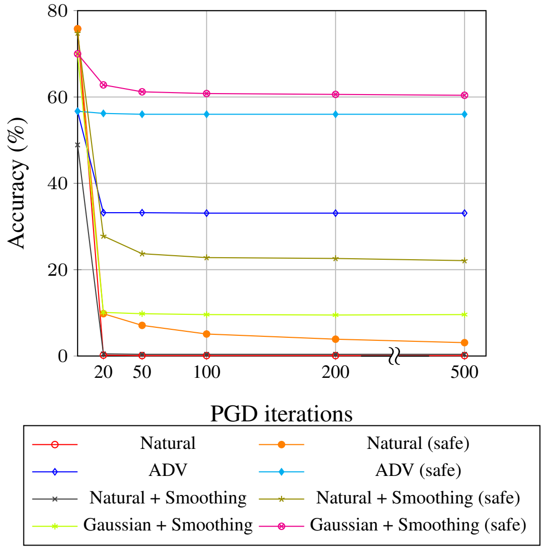


Figure 5: Evaluation with higher PGD iterations, under ℓ_2 threat with $\epsilon = 3.0$ on ImageNet. (safe) denotes using safe spot images in place of original images.

D.4 Experiments on other attacks

We ran additional experiments with multiple PGD restarts as well as CW attack Carlini & Wagner (2017) on CIFAR-10 under the ℓ_2 threat model and present the results below.

Model	**Method**	
	None	**S-Full**
Natural	96.0/00.5/00.0/01.4	96.0/59.0/20.3/22.5
ADV	90.3/68.2/68.1/71.2	90.3/90.0/89.3/90.0
S-FGSM+ADV	90.9/63.3/63.1/65.1	90.9/90.6/90.1/90.7
S-PGD+ADV	94.1/57.7/57.3/58.8	94.1/**93.5/93.0/93.9**

Table 4: Classification accuracy under ℓ_2 threat with $\epsilon = 0.5$ on CIFAR-10. (clean acc./PGD acc./PGD(m) acc./CW acc.)

'PGD(m)' denotes 500-step PGD with ten restarts, and 'CW' denotes CW attack with epsilon-ball projection at the end, following the experimental protocol of Song et al. (2017). The Table 4 results show that our algorithm also makes images safer for different types of attacks.

E Out-of-distribution detection with safe spots

E.1 Datasets

Gaussian The Gaussian noise dataset consists of 10,000 32 × 32 synthetic noise images, where each RGB pixel value is sampled i.i.d from Gaussian distribution with mean 0.5 and unit variance. The pixel values are clipped into the range $[0, 1]$.

CIFAR-100 The CIFAR-100 dataset consists of 60,000 32 × 32 color images in 100 classes. There are 50,000 training images and 10,000 test images. Although CIFAR-10 and CIFAR-100 datasets do have some similarities, their classes are mutually exclusive.

SVHN The SVHN dataset contains color images of house numbers in 32 × 32 size. It includes 604,388 train images and 26,032 test images.

TinyImageNet The TinyImageNet dataset is a 200-class subset of the ImageNet, where each class has 500 training images, 50 validation images, and 50 test images. The images are resized and cropped to 64×64 resolution. We downsample the images again to 32×32 resolution using Torchvision library (Paszke et al., 2019) to fit our CIFAR-10 classifiers.

LSUN The Large-scale Scene UNderstanding dataset (LSUN) is a dataset with 10 scene classes and consists of 10,000 test images. Similar to TinyImageNet, we downsample and center-crop the images to 32×32 using Torchvision library (Paszke et al., 2019).

80 Million Tiny Images (Torralba et al., 2008) The Tiny Images consists of 79,302,017 unlabeled images, each of which is a 32×32 image. Note that CIFAR-10 and CIFAR-100 are labeled subsets of Tiny Images.

E.2 Training and evaluation

We use the same WRN-34-10 architecture from previous CIFAR-10 experiments. Following the experimental protocol of Hendrycks et al. (2019), we train natural and S-PGD+ADV models for 100 epochs using SGD optimizer with weight decay $5E-4$, except for the cosine learning rate schedule. Instead, we set the initial learning rate to 0.1 and decay it with a factor 0.1 on 50 and 75 epochs. For the S-PGD+ADV model, we use 7-step ℓ_2-PGD with $\epsilon = 0.25$ to generate a safe spot and its adversarial example.

After training, we fine-tune the models for 10 epochs using a cosine learning rate schedule with an initial learning rate of 0.001. We use a subset of 80 Million Tiny Images that all the images in the CIFAR datasets are excluded as $\mathcal{D}_{\text{out}}^{\text{train}}$. For the natural model, we fine-tune it with the training objective proposed in Hendrycks et al. (2019), which is expressed by

$$\underset{\theta}{\text{minimize}} \quad \mathop{\mathbb{E}}_{(x_o,y)\sim\mathcal{D}_{\text{in}}} [\ell(x_o, y; \theta)] + \gamma \cdot \mathop{\mathbb{E}}_{\hat{x}_o\sim\mathcal{D}_{\text{out}}^{\text{train}}} [\ell(\hat{x}_o, \bar{y}; \theta)].$$

We tune γ within a range of $\{0.5, 1.0, 2.0\}$ and choose the best $\gamma = 1.0$. For the S-PGD+ADV model, we tune γ and λ within a range of $\{0.5, 1.0, 2.0\}$ and set $\gamma = 1.0$, $\lambda = 1.0$.

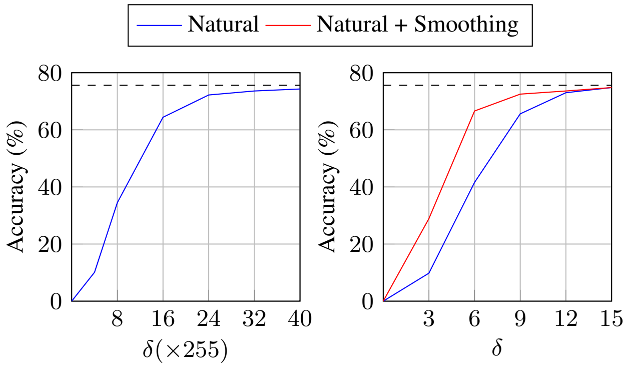
For evaluation, we randomly sample 10,000 test images from the in-distribution and out-of-distribution datasets and measure their **AUROC**, **AUPR**, and **FPR95** scores. The first two metrics summarize a method's detection performance across multiple thresholds, where higher values are better. **FPR95** evaluates the performance on a single threshold where the true positive rate becomes 95%, and the lower value is better. We treat OOD samples as the positive class. All results are averaged over 10 runs. To make safe spot search more efficient, we use ℓ_2 FGSM with $\epsilon = 0.25$ to solve the inner maximization in the safe spot objective. We set $\delta = 0.25$, MAXITER = 20, and $\beta = 0.0002$. For the score function, we tune μ within a range of $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and set $\mu = 0.5$.

F Transferability of safe spots

In this subsection, we examine whether the robustness of safe spots generated from one classifier transfers when they are fed to other classifiers. Table 5 shows safe spot transfer results for various robust classifiers (Madry et al., 2017; Liu et al., 2019; He et al., 2019; Wang & Yu, 2019; Zhang et al., 2019), where safe spots were generated from a certain robust classifier (source) and fed to other classifiers (target) robustly trained with different defense techniques. The result shows that regardless of defense methods or neural network architecture selections, at least 60% of our safe spots are transferable. The results also show that safe spots generated from more robust classifiers (*e.g.*TRADES) are more transferable to other robust classifiers. The results suggest that robust training methods, regardless of their procedural details, force decision boundaries to align with certain features, which is uncommon for standard training.

G Safe spot examples on larger δ

Until now, we searched for safe spots with δ set equal to ϵ. In the next experiments, we relax this condition and search for safe spots with larger δ values. Since safe spots can be easily found for



(a) ℓ_∞ threat, $\epsilon = 4/255$

(b) ℓ_2 threat, $\epsilon = 3.0$

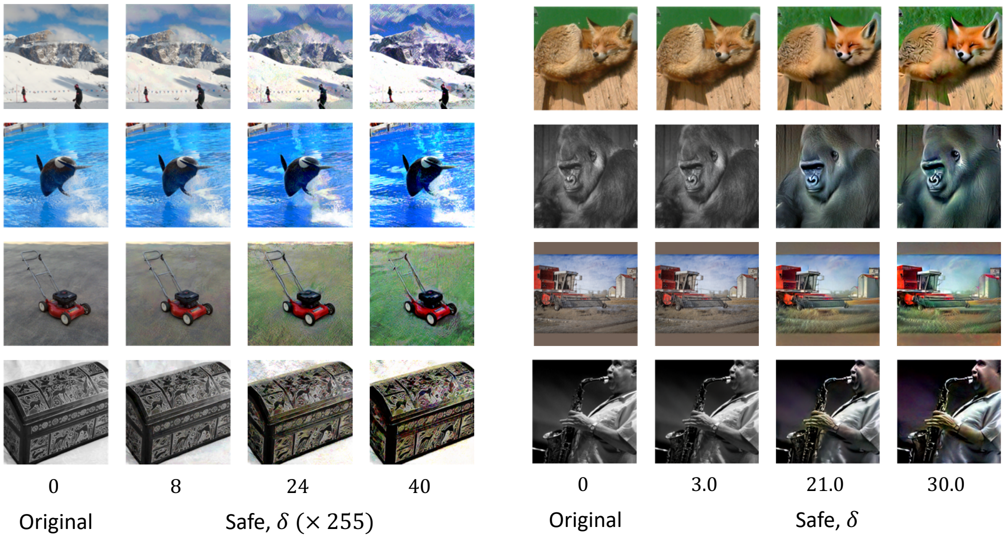
Figure 6: Adversarial accuracy of safe spots for a naturally trained ResNet-50 classifier on ImageNet under (a) ℓ_∞ threat with $\epsilon = 4/255$ and (b) ℓ_2 threat with $\epsilon = 3.0$ as δ-ball increases. The dashed line indicates the clean accuracy.

S/T	**Defense**	**Architecture**	**Method**		**S/T**	**Defense**	**Architecture**	**Method**	
			None	**S-Full (transfer)**				**None**	**S-Full (transfer)**
Source	ADV	WRN-34-10	86.6 / 49.3	86.6 / 84.7	Source	TRADES	WRN-34-10	85.2 / 55.6	85.2 / 84.8
Target	Adv-BNN	VGG-16	78.8 / 46.0	82.1 / 57.0	Target	Adv-BNN	VGG-16	78.8 / 46.0	83.0 / 66.2
	PNI	Noise-RN-20	83.1 / 39.6	86.3 / 50.7		PNI	Noise-RN-20	83.1 / 39.6	84.5 / 61.6
	RobustDL	RN-18	86.0 / 44.1	87.3 / 63.0		RobustDL	RN-18	86.0 / 44.1	86.8 / 62.4
	ADV	RN-50	87.0 / 52.4	87.6 / 66.4		ADV	WRN-34-10	86.6 / 49.3	85.8 / 67.0
	TRADES	WRN-34-10	85.2 / 55.6	86.6 / 67.5		ADV	RN-50	87.0 / 52.4	85.3 / 73.0

Table 5: Transferability results under ℓ_∞ threat with $\epsilon = 8/255$ on CIFAR-10. (clean acc. / adv acc.). Safe images from the source model are transfered to target models.

robust classifiers without increasing δ, we focus on the natural models. Figure 6 shows the results on ImageNet using the naturally trained ResNet-50 classifier. While Figure 6a and Figure 6b show safe spots can be found on most samples within ℓ_p balls of size 10ϵ and 5ϵ respectively, we note that on ℓ_∞ setting with $\epsilon = 8/255$, increasing δ values could not achieve the similar performance. These results suggest that as the strength of adversarial attack passes a certain threshold, safe spots may disappear on natural classifiers.

For robust classifiers, we increase ϵ along with δ to visualize safe spots that are *more safe*. The results in robust classifiers are shown in Figure 7. Interestingly, we observe that safe spots for robust classifiers emphasize certain human perceptible features of original images, such as mountain peak covered with snow, fur color of the fox, and field of grass. Also, the safe spots tend to render colors much more vivid, and the object boundaries crisper. The results suggest that safe spot algorithms could be applied to image synthesis tasks such as style transfer, super-resolution, or colorization, aligned with findings of Santurkar et al. (2019) on robust classifiers.



(a) ℓ_∞

(b) ℓ_2

Figure 7: Examples of safe spots for robust classifiers on ImageNet in ℓ_∞ and ℓ_2 setting. To encourage finding more safe spots, ϵ is increased to match δ. Inferred labels for ℓ_∞ are 'alp', 'killer whale', 'lawn mower', and 'chest', and inferred labels for ℓ_2 are 'red fox', 'gorilla', 'harvester', and 'saxophone'.

References

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2017.

Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *CVPR*, 2019.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 2012.

Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. In *ICLR*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Computer vision with a single (robust) classifier. In *NIPS*, 2019.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *ICLR*, 2017.

Antonio Torralba, Rob Fergus, and T. William Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. In *Pattern Analysis and Machine Intelligence*, 2008.

Huaxia Wang and Chun-Nam Yu. A direct approach to robust deep learning using adversarial networks. In *ICLR*, 2019.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.