Technical Appendix

Algorithm Supplementary

The second step of the proposed defense is a novel transformation procedure by improving the image distortion method as the preprocessing function. It can provide a great random variance between the original and transformed samples without affecting the model performance.

We improved the dropping-pixels strategy (Xie et al. 2018; Guo et al. 2018). The general idea is to drop certain randomly selected pixels from the original image, and displace each pixel away from the original coordinates. The whole procedure of the second step consists of four stages, as illustrated in Fig. 2 in the paper and Algorithm 1 in this appendix.

ALGORITHM 1: *Step 2 (Image Distortion)*

Input: original image $I \in \mathbb{R}^{h \times w}$
Output: distorted image $I' \in \mathbb{R}^{h \times w}$
Parameters: distortion limit $\delta \in [0, 1]$; size of grid d.

`/* 1.Select a starting point, e.g., upper-left corner */`
1 $x_0 = 0, y_0 = 0$;
`/* 2.Random distortion over grids */`
2 $n_w = w//d, n_h = h//d$;
3 $\mathcal{G}_I = \{(x_m, y_n) | (m, n) \in \{(0, ..., n_w) \times (0, ..., n_h)\}\}$;
4 **for** (x_m, y_n) *in* $\mathcal{G}_I \backslash \{(x_0, y_0)\}$ **do**
5 $\quad \delta_x \sim \mathcal{U}(-\delta, \delta)$;
6 $\quad \delta_y \sim \mathcal{U}(-\delta, \delta)$;
7 $\quad x_m = x_{m-1} + d \times (1 + \delta_x)$;
8 $\quad y_n = y_{n-1} + d \times (1 + \delta_y)$;
9 **end**
`/* 3.Remapping grids in` I `to` I' `*/`
10 $\mathcal{G}_{I'} = \{(x'_m, y'_n) | x'_m = d \times m, y'_n = d \times n, (m, n) \in \{(0, ..., n_w) \times (0, ..., n_h)\}\}$;
11 **for** (x'_m, y'_n) *in* $\mathcal{G}_{I'} \backslash \{(x'_0, y'_0)\}$ **do**
12 $\quad I'(x'_{m-1} : x'_m, y'_{n-1} : y'_n) = \texttt{Remapping}(I(x_{m-1} : x_m, y_{n-1} : y_n))$;
13 **end**
`/* 4.Reshape` I' `to the size of` I `*/`
14 $I' = \texttt{reshape}(I')$ s.t. $I' \in \mathbb{R}^{h \times w}$;
15 **return** I';

(1) One of the four corners is randomly selected as a starting point, e.g. the upper-left corner (line 1).

(2) The original image is a randomly distorted grid by grid. For one grid, it will be either stretched or compressed based on a distortion level sampled from a uniform distribution $\mathcal{U}(-\delta, \delta)$ (line 5-8).

(3) Distorted grids are then remapped to construct a new image (line 10-13). This step will drop pixels: the compressed grids will drop rows or columns of data; the stretched grids will cause the new image to exceed the original boundary such that the pixels mapped outside of the original boundary will be dropped (e.g., in Fig. 1, the grid at the lower-right corner in stage 2 is dropped in stage 3).

(4) Reshape the distorted image to the size of the original image by cropping or padding (line 14).

For the proposed defense, the distortion limit δ has an influence on the distortion level of each grid. It also affects the ratio of pixels that will be dropped. We apply a linear search of δ from 0.01 to 0.30, as shown in Table 2. The ASR becomes 0% under our defenses, which shows that the adversarial perturbation is delicate to this kind of distortion. A larger δ decreases the ACC on clean examples. Thus, a moderate $\delta = 0.15$ is chosen as the optimal value.

Mitigating Standard Attacks

We evaluate our defenses against standard attacks (FGSM, I-FGSM, LBFGS, and C&W). Here, we present the evaluations with experimental details and evaluations. All attacks are conducted as targeted attacks. We randomly select labels that are different from the original ones.

Table 1: Standard attacks on baseline model.

Attack	l_∞	l_2	**Baseline** ACC	ASR
Clean	0.000	0.0000	1.00	Nan
FGSM ($\epsilon = 0.01$)	0.010	0.0099	0.36	0.00
FGSM ($\epsilon = 0.03$)	0.030	0.0294	0.39	0.00
I-FGSM ($\epsilon = 0.01$)	0.010	0.0040	0.13	0.79
I-FGSM ($\epsilon = 0.03$)	0.030	0.0098	0.02	0.95
LBFGS	0.021	0.0013	0.00	1.00
C&W	0.156	0.0162	0.00	1.00

An attack succeeds only if the prediction of the model is the targeted class. We use Cleverhans (Papernot et al. 2018) to generate AE of all standard attacks. For FGSM and I-FGSM, AEs are generated under two different l_∞ constraints ($\epsilon = 0.01, 0.03$). I-FGSM is iterated ten times. For LBFGS

Table 2: Impact of distortion limits on defense performance of the proposed defense

Attack	$\delta = 0.01$		$\delta = 0.05$		$\delta = 0.10$		$\delta = 0.15$		$\delta = 0.20$		$\delta = 0.25$		$\delta = 0.30$	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Clean	0.95	Nan	0.96	Nan	0.95	Nan	0.95	Nan	**0.96**	Nan	0.93	Nan	0.91	Nan
FGSM ($\epsilon = 0.01$)	0.70	0.00	0.66	0.00	0.69	0.00	0.73	0.00	0.69	0.00	**0.75**	0.00	0.72	0.00
FGSM ($\epsilon = 0.03$)	0.51	0.00	0.51	0.00	0.51	0.00	0.53	0.00	0.55	0.00	0.55	0.00	**0.62**	0.00
I-FGSM ($\epsilon = 0.01$)	0.96	0.00	0.05	0.00	0.93	0.00	0.89	0.00	0.90	0.00	0.91	0.00	**0.93**	0.00
I-FGSM ($\epsilon = 0.03$)	0.88	0.01	0.90	0.00	0.86	0.00	**0.93**	0.00	0.92	0.00	0.89	0.00	0.89	0.00
LBFGS	0.95	0.00	**0.97**	0.00	0.93	0.00	0.91	0.00	0.94	0.00	0.94	0.00	0.88	0.00
C&W	0.86	0.00	0.87	0.00	0.85	0.00	**0.87**	0.00	0.83	0.00	0.83	0.00	0.84	0.00

Table 3: Performance of different defenses against standard attacks

Attack	**Baseline**		FD		Rand		Our Method	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Clean	1.00	Nan	**0.97**	Nan	0.96	Nan	0.95	Nan
FGSM ($\epsilon = 0.01$)	0.36	0.00	0.64	0.00	0.60	0.00	**0.73**	0.00
FGSM ($\epsilon = 0.03$)	0.39	0.00	0.47	0.00	0.60	0.00	0.53	0.00
I-FGSM ($\epsilon = 0.01$)	0.13	0.79	**0.96**	0.00	0.92	0.00	0.89	0.00
I-FGSM ($\epsilon = 0.03$)	0.02	0.95	0.87	0.00	0.86	0.01	**0.93**	0.00
LBFGS	0.00	1.00	**0.97**	0.00	0.95	0.00	0.91	0.00
C&W	0.00	1.00	0.84	0.00	0.86	0.00	**0.87**	0.00

and C&W, the optimization process is iterated until all targeted AEs are found under l_2 constraint. For LBFGS, the binary search steps are set to 5, and the maximum number of iterations is set to 1000. For C&W, the binary search steps are set to 5, the maximum number of iterations is set to 1000, and the learning rate is 0.1. We evaluate the model accuracy (ACC) and attack success rate (ASR), as well as the l_∞ norm and l_2 norm, Table 1 (note that FGSM is a one-step attack and it is not really effective as a targeted attack). Its iterative version I-FGSM with $\epsilon = 0.03$ can reach ASR 95%. Two optimization-based attacks, LBFGS and C&W, can even entirely break the baseline model with 100% ASR.

Finally, with chosen hyper-parameters, we compare the performance of different defenses against standard attacks, FD (Liu et al. 2019), Rand (Xie et al. 2018), Our Method, in Table 3. We chose these two methods since they are the two baseline defenses that can mitigate BPDA attack within 50 rounds as shown in Fig. 3 in the paper. Our method has a similar performance of ACC and ASR compared with FD and Rand considering only the standard adversarial attacks.

References

Guo, C.; Rana, M.; Cisse, M.; and van der Maaten, L. 2018. Countering Adversarial Images using Input Transformations. In *International Conference on Learning Representations*.

Liu, Z.; Liu, Q.; Liu, T.; Xu, N.; Lin, X.; Wang, Y.; and Wen, W. 2019. Feature distillation: DNN-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 860–868. IEEE.

Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; Matyasko, A.; Behzadan, V.; Hambardzumyan, K.; Zhang, Z.; Juang, Y.-L.; Li, Z.; Sheatsley, R.; Garg, A.; Uesato, J.; Gierke, W.; Dong, Y.; Berthelot, D.; Hendricks, P.; Rauber, J.; and Long, R. 2018. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv preprint arXiv:1610.00768* .

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2018. Mitigating Adversarial Effects Through Randomization. In *International Conference on Learning Representations*.