Supplementary Materials for $\mathcal{N}$ATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks

Yandong Li [*1] **Lijun Li** [*1] **Liqiang Wang** [1] **Tong Zhang** [2] **Boqing Gong** [3]

In this supplementary document, we provide the following details to support the main text:

Section A: descriptions of the 13 defense methods studied in the experiments,

Section B: architecture of the regression neural network for initializing our $\mathcal{N}$ATTACK algorithm, and

Section C: run-time analysis about $\mathcal{N}$ATTACK and BPDA (Athalye et al., 2018).

A. More Details of the 13 Defense Methods

- **Thermometer encoding (THERM).** To break the hypothesized linearity behavior of DNNs (Goodfellow et al., 2014a), Buckman et al. (2018) proposed to transform the input by non-differentiable and non-linear thermometer encoding, followed by a slight change to the input layer of conventional DNNs.

- **ADV-TRAIN & THERM-ADV.** Madry et al. (2018) proposed a defense using adversarial training (ADV-TRAIN). Specially, the training procedure alternates between seeking an "optimal" adversarial example for each input by projected gradient descent (PGD) and minimizing the classification loss under the PGD attack. Furthermore, Athalye et al. (2018) find that the adversarial robust training (Madry et al., 2018) can significantly improve the defense strength of THERM (THERM-ADV). Compared with ADV-TRAIN, the adversarial examples are produced by the logit-space projected gradient ascent in the training.

- **Cascade adversarial training (CAS-ADV).** Na et al. (2018) reduced the computation cost of the adversarial training (Goodfellow et al., 2014b; Kurakin et al., 2016) in a cascade manner. A model is trained from the clean data and one-step adversarial examples first. The second model is trained from the original data, one-step adversarial examples, as well as iterative adversarial examples generated against the first model. Additionally, a regularization is introduced to the unified embeddings of the clean and adversarial examples.

- **Adversarially trained Bayesian neural network (ADV-BNN).** Liu et al. (2019) proposed to model the randomness added to DNNs in a Bayesian framework in order to defend against adversarial attack. Besides, they incorporated the adversarial training, which has been shown effective in the previous works, into the framework.

- **Adversarial training with adversarial examples generated from GAN (ADV-GAN).** Wang & Yu (2019) proposed to model the adversarial perturbation with a generative network, and they learned it jointly with the defensive DNN as a discriminator.

- **Stochastic activation pruning (SAP).** Dhillon et al. (2018) randomly dropped some neurons of each layer with the probabilities in proportion to their absolute values.

- **RANDOMIZATION.** (Xie et al., 2018) added a randomization layer between inputs and a DNN classifier. This layer consists of resizing an image to a random resolution, zero-padding, and randomly selecting one from many resulting images as the actual input to the classifier.

- **Input transformation (INPUT-TRANS).** By a similar idea as above, Guo et al. (2018) explored several combinations of input transformations coupled with adversarial training, such as image cropping and rescaling, bit-depth reduction, JPEG compression.

- **PIXEL DEFLECTION.** Prakash et al. (2018) randomly sample a pixel from an image and then replace it with another pixel randomly sampled from the former's neighborhood. Discrete wavelet transform is also employed to filter out adversarial perturbations to the input.

[*]Equal contribution [1]University of Central Florida [2]Hong Kong University of Science and Technology [3]Google. Correspondence to: Yandong Li <lyndon.leeseu@outlook.com>, Boqing Gong <BoqingGo@outlook.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

Table 1. Average run time to find an adversarial example ($\mathcal{N}$**ATTACK-R** stands for $\mathcal{N}$ATTACK initialized with the regression net).

Defense	Dataset	BPDA (Athalye et al., 2018)	$\mathcal{N}$ATTACK	$\mathcal{N}$**ATTACK-R**
SAP (Dhillon et al., 2018)	CIFAR-10 (L_∞)	33.3s	29.4s	–
RANDOMIZATION (Xie et al., 2018)	ImageNet (L_∞)	3.51s	70.77s	48.22s

- **GUIDED DENOISER.** Liao et al. (2018) use a denoising network architecture to estimate the additive adversarial perturbation to an input.

- **Random self-ensemble (RSE).** Liu et al. (2018) combine the ideas of randomness and ensemble using the same underlying neural network. Given an input, it generates an ensemble of predictions by adding distinct noises to the network multiple times.

B. Architecture of the Regression Network

We construct our regression neural network by using the fully convolutional network (FCN) architecture (Shelhamer et al., 2016). In particular, we adapt the FCN model pretrained on PASCAL VOC segmentation challenge (Everingham et al., 2010) to our work by changing its last two layers, such that the network outputs an adversarial perturbation of the size $32 \times 32 \times 3$. We train this network by a mean square loss.

C. Run Time Comparison

Compared with the white-box attack approach BPDA (Athalye et al., 2018), $\mathcal{N}$ATTACK may take longer time since BPDA can find the local optimal solution quickly being guided by the approximate gradients. However, $\mathcal{N}$ATTACK can be executed in parallel in each episode. We leave implement the parallel version of our algorithm to the future work and compare its sing-thread version with BPDA below.

We attack 100 samples on one machine with fou TITAN-XP graphic cards and calculate the average run time for reaching an adversarial example. As shown in Table 1, $\mathcal{N}$ATTACK can succeed even faster than the white-box BPDA on CIFAR-10, yet runs slower on ImageNet. The main reason is that when the image size is as small as CIFAR10 (3*32*32), the search space is moderate. However, the run time could be lengthy for high resolution images like ImageNet (3*299*299) especially for some hard cases (we can find the adversarial examples for nearly 90% test images but it could take about 60 minutes for a hard case).

We use a regression net to approximate a good initialization of μ_0 and we name $\mathcal{N}$ATTACK initialized with the regression net as $\mathcal{N}$ATTACK-R. We run $\mathcal{N}$ATTACK and $\mathcal{N}$ATTACK-R on ImageNet with the mini-batch size $b = 40$. The success rate for $\mathcal{N}$ATTACK with random initialization is 82% and for $\mathcal{N}$ATTACK-R is 91.9%, verifying the efficacy of the regression net. The run time shown in Table 1 is calculated on the images with successful attacks. The results demonstrate that $\mathcal{N}$ATTACK-R can reduce by 22.5s attack time per image compared with the random initialization.

References

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Buckman, J., Roy, A., Raffel, C., and Goodfellow, I. Thermometer encoding: One hot way to resist adversarial examples. 2018.

Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, June 2010.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014a.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Guo, C., Rana, M., Cisse, M., and van der Maaten, L. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Liao, F., Liang, M., Dong, Y., Pang, T., Zhu, J., and Hu, X. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787, 2018.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In *European Conference on Computer Vision*, pp. 1–8. Springer, 2018.

Liu, X., Li, Y., Wu, C., and Hsieh, C.-J. Adv-BNN: Improved adversarial defense through robust bayesian neural network. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rk4Qso0cKm.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Na, T., Ko, J. H., and Mukhopadhyay, S. Cascade adversarial machine learning regularized with a unified embedding. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyRVBzap-.

Prakash, A., Moran, N., Garber, S., DiLillo, A., and Storer, J. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8571–8580, 2018.

Shelhamer, E., Long, J., and Darrell, T. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1605.06211*, 2016.

Wang, H. and Yu, C.-N. A direct approach to robust deep learning using adversarial networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1lIMn05F7.

Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sk9yuql0Z.