Supplementary Material for: Adversarial Defense via Learning to Generate Diverse Attacks

A. Experiment Details (Section 5)

We provide additional details on the model architecture, hyperparameters, and examples of noises of L2L-DA.

A.1. Architecture of our Generator Network g_ϕ

Recall that our generator g_ϕ in Eq. (5) and Algorithm 1 takes an image $\mathbf{x}$, the corresponding ground-truth label $\mathbf{y}$, a random noise $\mathbf{z} \in \mathbb{R}^d$, the accumulated noise $\delta^{(t)}$ up to the time step t starting from $\delta^{(0)} = \mathbf{0}$ (Eq. (5)), and the gradient backpropagated to the image $\nabla_{\mathbf{x}^{(t)}} \mathcal{L}(\mathbf{x}^{(t)})$. In this section, we describe the sequence of operations used for generating the adversarial noise.

First, in each time step, we obtain the gradient backpropagated to the input space $\nabla_{\mathbf{x}^{(t)}} \mathcal{L}(\mathbf{x}^{(t)})$ and normalize the size of each (height, width) matrix by its Frobenius norm. We denote the result of this operation as $\widetilde{\nabla}_{\mathbf{x}^{(t)}} \mathcal{L}(\mathbf{x}^{(t)}) = \nabla_{\mathbf{x}^{(t)}} \mathcal{L}(\mathbf{x}^{(t)}) / \|\nabla_{\mathbf{x}^{(t)}} \mathcal{L}(\mathbf{x}^{(t)})\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm. This operation allows our generator to control the magnitude of the adversarial attack independent of the choice of the backbone classifier (*i.e.* the magnitude of the gradient over the input $\mathbf{x}$ varies depending on the network), without losing the direction of the gradient.

After the normalization, we then concatenate $\mathbf{x}$, $\delta^{(t)}$ and $\widetilde{\nabla}_{\mathbf{x}^{(t)}} \mathcal{L}(\mathbf{x}^{(t)})$ channel-wise and denote it as $\tilde{\mathbf{x}} = [\mathbf{x}; \delta^{(t)}; \widetilde{\nabla}_{\mathbf{x}^{(t)}} \mathcal{L}(\mathbf{x}^{(t)})] \in \mathbb{R}^{N \times H \times W \times 3C}$, where N, H, W, C denotes the mini-batch size, height, width and the number of channels of $\mathbf{x}$ (*e.g.* C = 3 for RGB images and C = 1 for gray-scale images).

Also, we sample a random noise $\mathbf{z}$ from the standard normal distribution $\mathcal{N}(0, \mathbf{I})$ and feed it to two fully-connected layers with the same number of hidden states (d), with ReLU activation after each fully-connected layer, to obtain a richer expression $\tilde{\mathbf{z}}$, following the idea of [5].

We generate the adversarial attack from $\tilde{\mathbf{x}}$ by passing it to a set of 2D convolutions (`conv`(l, k, s)), 2D deconvolutions (`deconv`(l, k, s)) of padding size 1, and channel-wise concatenations with an additional tensor $\mathbf{m}$ (`ccat`($\mathbf{m}$)): here, l means the number of filters, k means the width and height of the square kernel, and s means the stride. As illustrated in Figure A, most of the design choices are adopted from Zhu *et al.* [10] and Chen *et al.* [1].

The generator g_ϕ of L2L-DA applies the following operations to $\tilde{\mathbf{x}}$ to get the adversarial noise: $\tilde{\mathbf{x}}$ - `ccat`(`tile`($\tilde{\mathbf{z}}$)) - `conv`(64, 3, 1) - `LeakyReLU` - [*skip*$_1$] - `ccat`(`tile`($\tilde{\mathbf{z}}$)) - `conv`(64, 4, 2) - `CondBN`($\mathbf{y}$) - `LeakyReLU` - [*skip*$_2$] - `ccat`(`tile`($\tilde{\mathbf{z}}$)) - `conv`(128, 4, 2) - `LeakyReLU` - `deconv`(64, 4, 2) - `CondBN`($\mathbf{y}$) - `ReLU` - `ccat`(*skip*$_2$) - `deconv`(64, 4, 2) - `CondBN`($\mathbf{y}$) - `ReLU` - `ccat`(*skip*$_1$) - `conv`(C, 3, 1) - `tanh`, where `tile`($\tilde{\mathbf{z}}$) denotes the spatial tiling of the vector $\tilde{\mathbf{z}}$ to have the same width and height of the current tensor, `LeakyReLU` [9] has a negative slope of 0.2, `CondBN`($\mathbf{y}$) is the class-conditional batch normalization [3, 2] with ground-truth label input $\mathbf{y}$ as the condition, and $[(\cdot)]$ denotes a branch for skip connections in the deconvolution stage. Note that we use $\mathbf{y}$ solely for the class-conditional batch normalization to encourage our model focus on its class-conditional distribution rather than the global statistics.

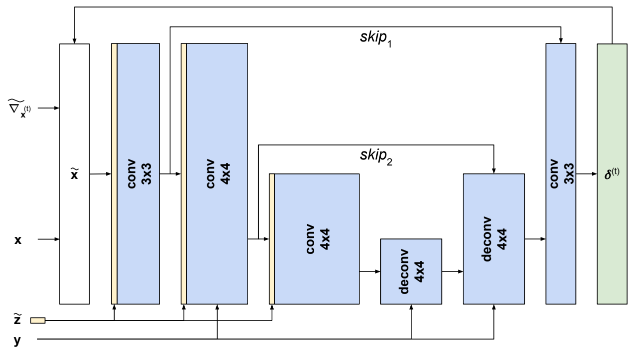
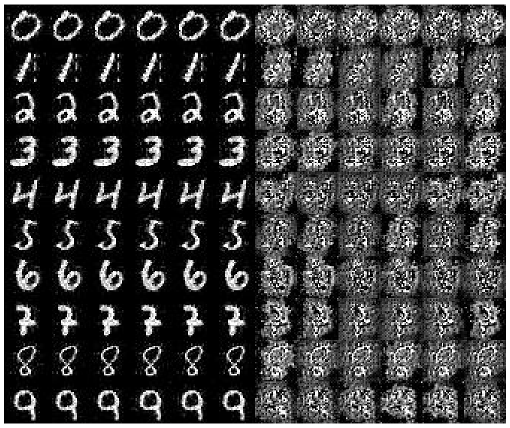


Figure A: Network architecture of the generator g_ϕ (used in Figure 1 of the main paper)

A.2. Hyperparameters

We empirically choose the hyperparameter λ in Eq. (8), to balance between the diversity loss and the classification loss, by validation. We use $\lambda = 0.02$ for MNIST with LeNet-5 classifier [7], $\lambda = 0.5$ for CIFAR-10 [6] with ResNet-20 classifier [4], and $\lambda = 0.3$ for Tiny ImageNet [8] with ResNet-18 classifier. For the dimension of $\mathbf{z}$, we set it to $d = 8$ for all the experiments (MNIST, CIFAR-10, and Tiny ImageNet).


(a) MNIST [7]

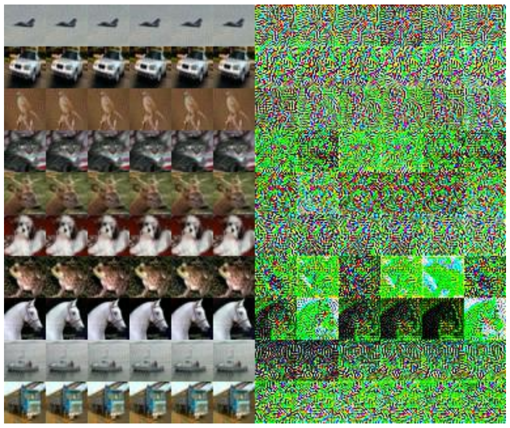

(b) CIFAR-10 [6]

Figure B: Visualization of adversarial examples and the corresponding adversarial noises. We feed different $\mathbf{z}$ (corresponds to each column) to an item per each class in the test dataset (each row) to our generator. The left half of the images are the adversarial outputs, and the right half of the images are the adversarial noises applied to the original test image. To visualize the noises, we first re-scale the size of the ℓ_∞-ball to 1/2 and then shift every value by 1/2.

Defense \ Attack	Natural	FGSM	PGD10	PGD100	CW100	CW1000	AdvGAN	GAP	L2L-DA	Min
Plain	**68.27**	0.93	0.01	0.00	0.00	0.00	**53.07**	32.24	0.00	0.00
PGD10	40.69	12.69	9.58	8.97	9.57	9.43	39.11	37.06	9.78	8.97
PGD40	47.41	16.10	12.85	12.09	11.99	11.96	44.19	44.20	18.79	11.96
L2L*	59.89	14.67	0.27	0.05	0.00	0.00	0.48	0.12	0.56	0.00
L2L-DA (`full`)	52.64	**20.05**	**16.24**	**15.53**	**14.47**	**14.34**	48.73	**48.83**	**24.70**	**14.34**

Table A: Accuracy of each trained classifier on Tiny ImageNet dataset [8] with white-box attacks. Each classifier is pretrained on ResNet-18 [4] (each row), and each of the white-box attack (each column) is applied to each classifier.

B. Experimental Results on Tiny Imagenet dataset (Section 5.2.2)

As a sanity check, we additionally test our approach on the Tiny ImageNet dataset[1] [8], which includes 500/50/50 train/validation/test images of 64x64 from each of the 200 object classes. We train each model with ResNet-18 backbone [4] for 100 epochs, and 20 epochs for PGD40 due to its training speed. Besides that, we use the validation set for the evaluation instead of the test set because it does not have ground-truth labels. This is because we use the classification accuracy of the model, which requires the label information for computation, as a proxy measure of robustness. We present the result of the white-box attacks in Table A.

Similar to the experiment in Section 5.2.2, we train the classifiers with adversarial attacks (each row) on ResNet-18 [4] and measure the accuracy of the classifier after performing a white-box attack (each column). As in the Table A, the classifier trained with our method shows 2.38%p higher classification accuracy than all the other baseline methods in terms of 'Min' accuracy.

[1]`https://learningai.io/projects/2017/06/29/tiny-imagenet.html`

C. Qualitative Evaluation of Noises and Adversarial Data

We visualize some of the noises generated by our generator trained in Section 5.2 with MNIST and CIFAR-10 dataset. Figure B shows that the noises generated by our generator are diverse, and perceptually indistinguishable when added to the given test image for generating adversarial samples. For example, the sixth row of Figure B.(a) shows a very different style of noises generated by sampling different $\mathbf{z}$'s: some of them put more weights on the background, whereas others add more noise on the digit itself. Similarly, in the adversarial noises in the seventh row of Figure B.(b), some of the noises try to change the center area of the image to black, while others tend to change it to green, depending on the $\mathbf{z}$ provided. Even though the original ℓ_∞-ball for the CIFAR-10 dataset is 8/255, our model tries to generate attacks with different noises. This result demonstrates that our method is able to generate a diverse attack.

References

[1] Zhehui Chen, Haoming Jiang, Bo Dai, and Tuo Zhao. Learning to Defense by Learning to Attack. *arXiv:1811.01213*, 2018.

[2] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating Early Visual Processing by Language. In *NeurIPS*, 2017.

[3] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A Learned Representation For Artistic Style. In *ICLR*, 2017.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[5] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *CVPR*, 2019.

[6] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, Citeseer, 2009.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 1998.

[8] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel Recurrent Neural Networks. In *ICML*, 2016.

[9] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv:1505.00853*, 2015.

[10] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward Multimodal Image-to-Image Translation. In *NeurIPS*, 2017.