A APPENDIX

Table 1: Results for different kinds of models under 42 different adversarial attacks, arranged according to distance metrics. Almost all attacks in Foolbox v3.2.1. are included. Each entry shows the accuracy of the model for the thresholds of $\epsilon_{L_0} = 12$, $\epsilon_{L_1} = 8$, $\epsilon_{L_2} = 1.5$, and $\epsilon_{L_\infty} = 0.3$. For each $L_p, p = 0, 1, 2, \infty$ norm, we also summarize all attacks in the type, calculating the overall accuracy.

	CNN	biCNN	Madry	biABS	ABS	biDIM	DIM
L_2-metric($\epsilon = 1.5$)							
L_2 ContrastReductionAttack	99%	98%	99%	99%	99%	95%	96%
L_2 DDNAttack	15%	71%	94%	85%	84%	92%	93%
L_2 PGD	30%	76%	96%	86%	88%	93%	94%
L_2 BasicIterativeAttack	17%	67%	95%	83%	83%	93%	94%
L_2 FastGradientAttack (FGM)	55%	92%	97%	94%	86%	94%	95%
L_2 AdditiveGaussianNoiseAttack (GN)	99%	98%	98%	99%	99%	95%	96%
L_2 AdditiveUniformNoiseAttack (UN)	99%	98%	99%	99%	99%	96%	96%
L_2 ClippingAwareGN	99%	98%	98%	99%	99%	96%	96%
L_2 ClippingAwareUN	99%	98%	99%	99%	99%	96%	96%
L_2 RepeatedGN	99%	97%	98%	97%	98%	92%	95%
L_2 RepeatedUN	99%	98%	98%	97%	98%	93%	95%
L_2 ClippingAwareRepeatedGN	99%	97%	98%	97%	98%	92%	95%
L_2 ClippingAwareRepeatedUN	98%	97%	98%	97%	98%	92%	95%
L_2 DeepFoolAttack	21%	21%	95%	49%	83%	75%	89%
L_2 InversionAttack	99%	98%	99%	99%	99%	95%	96%
L_2 BinarySearchContrastReductionAttack	99%	98%	99%	99%	99%	95%	96%
L_2 LinearSearchContrastReductionAttack	99%	98%	99%	99%	99%	95%	96%
L_2 GaussianBlurAttack	99%	98%	98%	98%	99%	95%	96%
L_2 CarliniWagnerAttack	13%	10%	83%	45%	84%	51%	74%
L_2 BrendelBethgeAttack	12%	8%	50%	48%	93%	57%	71%
L_2 BoundaryAttack	19%	62%	54%	93%	90%	80%	80%
All L_2 attacks	9%	7%	41%	41%	**83**%	45%	66%
L_∞-metric($\epsilon = 0.3$)							
L_∞ PGD	0%	73%	95%	88%	11%	89%	85%
L_∞ BasicIterativeAttack	0%	70%	96%	83%	8%	89%	82%
L_∞ FastGradientAttack (FGSM)	7%	78%	96%	86%	38%	90%	89%
L_∞ AdditiveUniformNoiseAttack	96%	98%	99%	98%	99%	96%	96%
L_∞ RepeatedAdditiveUniformNoiseAttack	83%	95%	97%	96%	97%	89%	93%
L_∞ DeepFoolAttack	0%	83%	95%	86%	7%	91%	78%
L_∞ InversionAttack	28%	98%	98%	99%	76%	95%	95%
L_∞ BinarySearchContrastReductionAttack	28%	98%	98%	98%	82%	94%	94%
L_∞ LinearSearchContrastReductionAttack	28%	98%	98%	98%	82%	94%	94%
L_∞ GaussianBlurAttack	97%	97%	98%	97%	98%	93%	95%
L_∞ LinearSearchBlendedUniformNoiseAttack	67%	98%	98%	98%	98%	93%	95%
L_∞ BrendelBethgeAttack	2%	81%	94%	89%	11%	88%	9%
All L_∞ attacks	0%	69%	**93**%	82%	3%	78%	8%
L_0-metric($\epsilon = 12$)							
SaltAndPepperAttack	93%	93%	73%	97%	98%	90%	93%
Pointwise $\times 10$	25%	43%	2%	82%	76%	53%	59%
All L_0 attacks	25%	43%	2%	**82**%	76%	53%	59%
L_1-metric($\epsilon = 8$)							
L_1 InversionAttack	99%	98%	99%	99%	99%	95%	96%
L_1 BinarySearchContrastReductionAttack	99%	98%	99%	99%	99%	95%	96%
L_1 LinearSearchContrastReductionAttack	99%	98%	99%	99%	99%	95%	96%
L_1 GaussianBlurAttack	99%	98%	98%	99%	99%	95%	96%
L_1 LinearSearchBlendedUniformNoiseAttack	99%	98%	99%	99%	99%	95%	96%
L_1 BrendelBethgeAttack	11%	4%	16%	48%	89%	65%	65%
All L_1 attacks	11%	4%	16%	47%	**89**%	65%	65%

Table 2: Results for ablation study under 42 defferent adversarial attacks, arranged according to distance metrics. There are six models: the vanilla CNN, the single-head Internal Model (single-IM), the Internal Model without denoiser (IM), the single-head Internal Model with denoiser (Dn-singleIM), the DIM, and the biDIM.

	CNN	singleIM	IM	Dn-singleIM	DIM	biDIM
L_2-metric($\epsilon = 1.5$)						
L_2 ContrastReductionAttack	99%	95%	96%	95%	96%	95%
L_2 DDNAttack	15%	83%	91%	87%	93%	92%
L_2 PGD	30%	89%	95%	89%	94%	93%
L_2 BasicIterativeAttack	17%	88%	94%	90%	94%	93%
L_2 FastGradientAttack (FGM)	55%	89%	95%	90%	95%	94%
L_2 AdditiveGaussianNoiseAttack (GN)	99%	95%	96%	94%	96%	95%
L_2 AdditiveUniformNoiseAttack (UN)	99%	95%	96%	94%	96%	96%
L_2 ClippingAwareGN.	99%	95%	96%	95%	96%	96%
L_2 ClippingAwareAdditiveUN	99%	95%	96%	94%	96%	96%
L_2 RepeatedGN	99%	93%	96%	93%	95%	92%
L_2 RepeatedUN	99%	93%	95%	93%	95%	93%
L_2 ClippingAwareRepeatedGN	99%	93%	95%	92%	95%	92%
L_2 ClippingAwareRepeatedUN	98%	93%	95%	93%	95%	92%
L_2 DeepFoolAttack	21%	71%	83%	82%	89%	75%
L_2 InversionAttack	99%	95%	96%	95%	96%	95%
L_2 BinarySearchContrastReductionAttack	99%	94%	96%	94%	96%	95%
L_2 LinearSearchContrastReductionAttack	99%	94%	96%	94%	96%	95%
L_2 GaussianBlurAttack	99%	94%	96%	93%	96%	95%
L_2 CarliniWagnerAttack	13%	54%	66%	68%	74%	51%
L_2 BrendelBethgeAttack	12%	61%	58%	70%	71%	57%
L_2 BoundaryAttack	19%	65%	67%	75%	80%	80%
All L_2 attacks	9%	52%	51%	65%	**66**%	45%
L_∞-metric($\epsilon = 0.3$)						
L_∞ PGD	0%	49%	70%	72%	85%	89%
L_∞ BasicIterativeAttack	0%	54%	61%	72%	82%	89%
L_∞ FastGradientAttack (FGSM)	7%	64%	78%	79%	89%	90%
L_∞ AdditiveUniformNoiseAttack	96%	95%	96%	95%	96%	96%
L_∞ RepeatedAdditiveUniformNoiseAttack	83%	90%	93%	91%	93%	89%
L_∞ DeepFoolAttack	0%	44%	61%	66%	78%	91%
L_∞ InversionAttack	28%	96%	95%	92%	95%	95%
L_∞ BinarySearchContrastReductionAttack	28%	93%	94%	91%	94%	94%
L_∞ LinearSearchContrastReductionAttack	28%	93%	94%	91%	94%	94%
L_∞ GaussianBlurAttack	97%	92%	94%	93%	95%	93%
L_∞ LinearSearchBlendedUniformNoiseAttack	67%	94%	95%	93%	95%	93%
L_∞ BrendelBethgeAttack	2%	2%	1%	6%	9%	88%
All L_∞ attacks	0%	2%	0%	6%	8%	**78**%
L_0-metric($\epsilon = 12$)						
SaltAndPepperAttack	93%	90%	92%	91%	93%	90%
Pointwise $\times 10$	25%	54%	50%	58%	59%	53%
All L_0 attacks	25%	54%	50%	58%	**59**%	53%
L_1-metric($\epsilon = 5$)						
L_1 InversionAttack	99%	95%	96%	95%	96%	95%
L_1 BinarySearchContrastReductionAttack	99%	94%	96%	94%	96%	95%
L_1 LinearSearchContrastReductionAttack	99%	94%	96%	94%	96%	95%
L_1 GaussianBlurAttack	99%	94%	96%	94%	96%	95%
L_1 LinearSearchBlendedUniformNoiseAttack	99%	94%	96%	94%	96%	95%
L_1 BrendelBethgeAttack	11%	61%	57%	65%	65%	65%
All L_1 attacks	11%	61%	57%	65%	**65**%	65%

A.1 MODEL & TRAINING DETAILS

A.1.1 HYPERPARAMETERS AND TRAINING DETAILS FOR DIM

In DIM, we train 1 denoiser and 10 internal models, separately. The denoiser contains a fully-connected encoder with 5 layers of the width [784,560,280,140,70], where the last layer uses linear and the others ReLU, and a fully-connected decoder with 5 layers of the width [70,140,280,560,784], where the last layer uses Tanh and the others ReLU. Each internal model contains a fully-connected encoder with 5 layers of the width [784,256,64,12,10] where the last layer uses linear and the others ReLU, and a fully-connected decoder with 5 layers of the width [10,12,64,256,784] where the last layer uses Tanh and the other ReLU. There are two types of noises added onto the input on each stage, an L_∞ noise randomly from the space $[-0.5, 0.5]^n$, and an L_0 noise with a probability 1/12 to increase by 1 and also 1/12 to decrease by 1 for every dimension of all pixels. Both of the denoiser and the internal models are trained using the Adam optimizer with the learning rate of 10^{-3}. In addition, when training the internal models, we also randomly tune the image brightness by multiplying a factor in the range $[0, 1]$ after adding the noise.

A.1.2 HYPERPARAMETERS AND TRAINING DETAILS FOR MADRY

We adopt the same network architecture as in Madry et al. (2018), includes two convolutional layers, two pooling layers and two fully connected layers. We implement the model in PyTorch and perform adversarial training using the same settings as in the original paper.

A.1.3 HYPERPARAMETERS AND TRAINING DETAILS FOR CNN & ABS MODELS

For the CNN and the ABS/biABS cases, we load the pre-trained models provided by Schott et al. (2019). There are 4 convolutional layers with kernel sizes=[5,4,3,5] in the CNN model. The ABS/biABS model contains 10 variational autoencoders, one for each category in the dataset.

A.2 ATTACK DETAILS

To apply gradient-based attacks on the models with input binarization, we exploit a transfer-attack-like procedure. Specifically, a sigmoid function is used in substitute of the direct binarization. Then we place this differentiable proxy model directly under attacks from foolbox v3.2.1. There is a scale parameter α in the sigmod function, i.e. $\frac{1}{1+e^{-\alpha x}}$, which controls how steep the function is when increasing from 0 to 1. For each attack on a binary model, we attack the model 5 times for $\alpha = 10, 15, 20, 50$, and 100, respectively. At last, we adopt a finetune procedure on all the generated adversarial samples. To be concrete, if a pixel value in the adversarial image is different from its original value, we project the value to 0 or 0.5 as long as it retains the same result under binarization.

Note that this fine-tune procedure also applies to the non-gradient-based attacks on binary models. However, in this case the transfer-attack-like procedure is no longer needed.

For normal (i.e., non-binary) models, we use the default settings of attacks in the foolbox v3.2.1 package.