A. Experiment Setup

For data pre-processing, we resize the images in CelebA to 128×128 using bicubic interpolation, and use 10% of total images as test data. For both datasets, we normalize the data into the range of $[-1, 1]$. On Fashion MNIST, we use a LeNet-style CNN (Table A). For CelebA dataset, we use the standard ResNet [21] with depth 20. Models are trained using stochastic gradient descent with momentum.

Table A: Architecture of CNN used in Fashion MNIST.

Net
Conv(128,3,3) + Relu
Conv(64,3,3) + Relu
Dropout(0.25)
FC(128) + Relu
Dropout(0.5)
FC(10) + Softmax

B. Attack Setting

For each attack setting, we generate adversarial examples under the whitebox setting using two standard methods: Fast Gradient Sign Method (FGSM) [19], Projected Gradient Descent (PGD) [39]. Additionally, we also evaluate our pipelines on blackbox setting by using SPSA algorithm [55].

For PGD attacks, we evaluate 10 steps and 40 steps PGD, denoted as 'PGD-10' and 'PGD-40' separately. We conduct an additional experiment by changing the number of PGD steps to verify the convergence of PGD algorithms. For ℓ_∞ distance of 8/256, the step size is set to be 0.005. For ℓ_∞ distance of 25/256 , we use step size 0.015. We use Robust Canny for evaluation of adversarial robustness. Here we report the hyper-parameters used in Robust Canny, which are chosen using the validation set to trade off robustness and accuracy. For Fashion MNIST, we set $\sigma = 1, \theta_l = 0.1, \theta_h = 0.2, \alpha = 0.3$. For CelebA, we set $\sigma = 2.5, \theta_l = 0.2, \theta_h = 0.3, \alpha = 0.2$. For CIFAR-10 and Tiny ImageNet, we set $\sigma = 1, \theta_l = 0.2, \theta_h = 0.3, \alpha = 0.3$.

For SPSA attack, we select the number of SPSA iterations as 40 and SPSA sample size as 1024 for Fashion MNSIT and CelebA. For CIFAR-10 and Tiny ImageNet, we select the number of SPSA iterations as 8 and SPSA samples size as 2048.

C. Differentiable Canny

Note that the last three steps in the Robust Canny algorithm are non-differentiable transformations. However, in a stronger white-box attack scenario one needs to backpropagate gradient through the edge detection algorithm for constructing adversarial samples. While obfuscating gradients through non-differentiable transformations is a commonly used defense technique, Athalye et al. [1] show that the attacker can replace such transformation with differentiable approximations, refered to as the Backward Pass Differentiable Approximation (BPDA) technique, to construct adversarial examples. Therefore, to realize a stronger attack on our method, we find a differentiable approximation of the Robust Canny algorithm as follows.

Assuming x to hold the pixel intensities in the original image, and x_e to be the output of the Robust Canny algorithm, we can break the transformation into two stages: $C_1(\cdot)$, comprised of step 1-3, and $C_2(\cdot)$ for steps 4-6 (Thresholding operation in step 3 can be formulated as a shifted ReLU function). Note that $C_2(\cdot)$ is a non-differentiable operation, where the output is a masked version of the input: $C_2(x) = M(x) \otimes x$, where $M(\cdot)$ produces the mask (i.e., an array of zeros and ones) produced by steps 3-6, and $\otimes$ denotes element-wise multiplication. Therefore, we can write:

$$x_e = \textit{R-Canny}(x) = C_2\left(C_1(x)\right) = M\left(C_1(x)\right) \otimes C_1(x) \tag{3}$$

To obtain a differentiable approximation of Robust Canny for BPDA, we assume the mask to be constant during the back propagation phase. In other words, we only backpropagate gradients through $C_1(\cdot)$, and not $M(\cdot)$.

D. Details On the Combined Edge

As shown in Figure 3 (right), the edge extracted by Robust Canny (second row in the figure) contains limited shape information due to the poor image quality of CIFAR-10. Thus, as shown in Table 3, when we apply the EdgeNetRob by using the edge extracted by Robust Canny algorithm, the clean accuracy is only 67.85%. To improve the clean accuracy, we need to get more informative shape. In the third row of Figure 3 (right), we leverage train a CNN-based edge detector to extracted the edge image. We observe that the edge extracted by CNN-based algorithm is more informative then robust canny. For example, the edge in the third row succeeds to reflect some detailed information (e.g. the window of the car). When we apply EdgeNetRob on these edges, we could get 87.11% clean accuracy. However, the problem of CNN edge as mentioned before is the vulnerability against adaptive attack. As shown in Table 3, the robust accuracy agaisnt PGD attack is 0.82% which is much lower than 36.31% achieved by Robust Canny. Because the edge from robust canny is less informative but robust while the edge from CNN-based algorithm is informative but vulnerable, it motivates us to combine them together to achieve an informative and robust edge detector algorithm. Therefore, we proposed a combined edge algorithm for CIFAR-

Table B: Hyper-parameter settings in the experiments.

Dataset	Model	Optimizer	Momentum	Epochs	Learning Rate	LR Step Decay
Fashion MNIST	LeNet	SGD	0.9	60	0.001	30, 45
CelebA	ResNet 20	SGD	0.9	40	0.1	20, 30
CIFAR-10	ResNet 20	SGD	0.9	160	0.1	80,120
Tiny ImageNet	ResNet 20	SGD	0.9	90	0.1	30,60

10.

$$e_{\text{combined}} = \beta \odot (e_{\text{cnn}} \odot e_{\text{robust canny}}) + (1-\beta) \odot e_{\text{robust canny}} \quad (4)$$

where β is a random value in range [0,1] with the uniform distribution.

E. Additional Experimental Results

In this section, we will show the additional quantitative and qualitative results among different robustness settings.

E.1. Robustness against Adversarial Attacks

Figure 3 (left) show the edges of clean (benign) and adversarial examples among different edge detector (vanilla canny, cnn-based, robust canny) on Fashion MNIST and CelebA. We could observe that the edges between benign and adversarial images are different for the Canny and CNN-based edge detection algorithms. However, for the proposed robust canny algorithm, the edges are almost similar between benign and adversarial images. These visualization results also indicated the vulnerability of vanilla and CNN-based edge detectors. As the vanilla edge images are also different between adversarial and benign images on CIFAR-10 and Tiny ImageNet, we do not visualize the edges extracted by vanilla Canny in Figure 3 (right). Figure 3 (right), as described in the previous section, mainly aims to show the edge informative's property among different edge detectors on CIFAR-10 and Tiny ImageNet.

E.2. Robustness against Backdoor Attacks

Figure A shows the qualitative results of EdgeGANRob and EdgeNetRob for backdoor attacks on Fashion MNIST, CelebA, CIFAR-10 and Tiny ImageNet datasets. We can also observe that the poisoning pattern can be slightly removed by EdgeNetRob and the patterns for each of the generated images do not share the similar patterns.

E.3. Generalization to texture datasets

We tested our approach on Describable Textures Dataset. To evaluate the effectiveness of edge information for recognizing textures, we trained both EdgeNetRob and EdgeGANRob and evaluate with different texture recognition methods. We compared with the numbers in Table 2 in [10].

Table C: Comparison of EdgeNetRob and EdgeGANRob on DTD with standard texture recognition benchmarks.

	IFV	BoVW	VLAD	LLC	KCB	DeCAF
Baseline	61.2	55.5	59.7	54.7	53.2	54.8
EdgeNetRob	50.3	45.3	50.9	51.2	45.2	39.4
EdgeGANRob	55.2	52.8	55.7	54.3	49.9	50.2

From Table C, we observe that edge information is still helpful for texture recognition. EdgeGANRob achieves better accuracy than EdgeNetRob as EdgeGANRob reconstructs the original images, which may be better for extracting useful encodings for texture recognition.

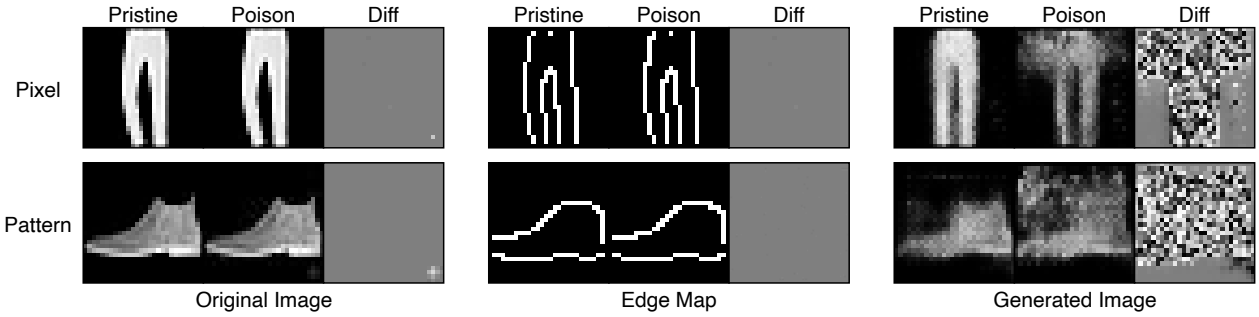
To evaluate adversarial robustness, we use the DeCAF framework as it is a deep convolutional network. We select the standard perturbation budget as $\ell_\infty = 8/255$. The results are shown in Table D, We observe that EdgeGANRob still provides better adversarial robustness than baseline approach on the Describable Texture Datasets.

Table D: Robustness evaluation on DTD.

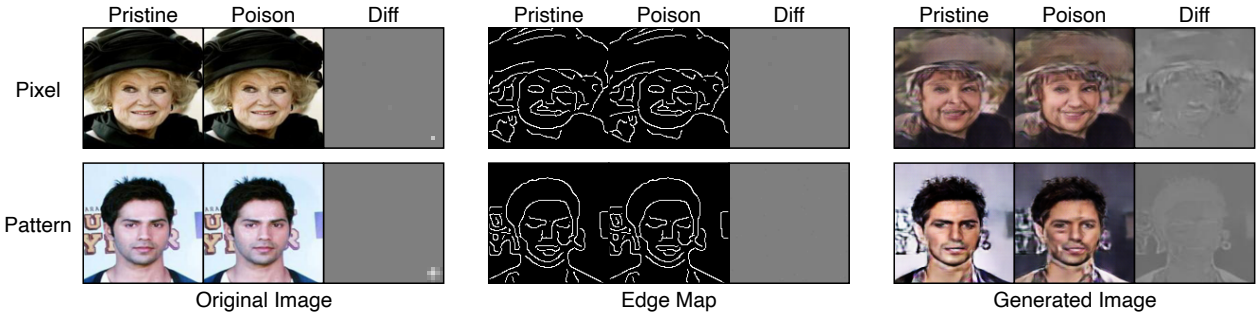
	Clean Acc	FGSM	PGD-10	PGD-40
baseline	54.8	34.2	10.5	2.9
EdgeGANRob	50.2	44.9	39.2	38.4

Table E: Evaluation of adversarial robustness on various datasets. The results of edge feature enabled pipelines are shown in grey.

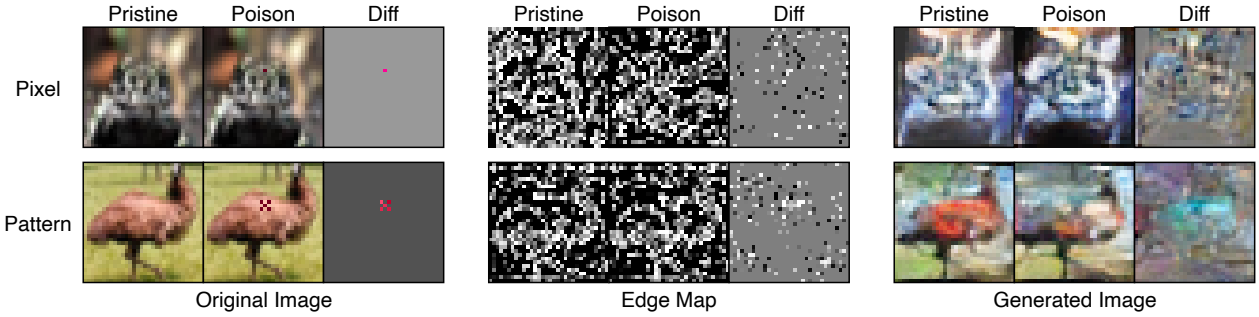
Dataset	Method	Clean	Whitebox	Blackbox
Fashion MNIST	Vanilla Net	92.88	0.48	2.11
	PGD-training	86.99	72.62	74.21
	EdgeNetRob	87.00	**76.75**	73.94
	EdgeGANRob	**87.14**	72.69	**77.09**
CelebA	Vanilla Net	98.30	0.00	0.23
	PGD-training	92.75	81.31	83.69
	EdgeNetRob	94.51	82.81	83.01
	EdgeGANRob	**95.88**	**84.60**	**85.02**
CIFAR-10	Vanilla Net	91.89	0.00	35.21
	PGD-training	76.50	**44.15**	53.69
	EdgeNetRob	**79.21**	33.08	56.38
	EdgeGANRob	76.25	37.15	**59.26**
Tiny-ImageNet	Vanilla Net	58.55	0.00	16.20
	PGD-training	48.10	**22.31**	**36.18**
	EdgeNetRob	**48.20**	19.53	34.75
	EdgeGANRob	44.30	13.55	31.06



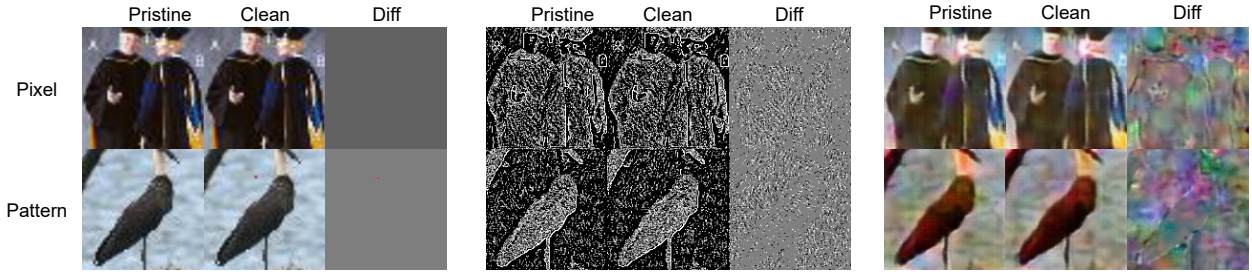
(a) Fashion MNIST



(b) CelebA



(c) CIFAR-10



(d) Tiny ImageNet

Figure A: Qualitative results of EdgeGANRob (EdgeNetRob) for backdoor attacks. We show the figures for two backdoors we used in the experiments (pixel and pattern).