Supplementary for FDA: Feature Disruptive Attack

Aditya Ganeshan*
Preferred Networks Inc.
Tokyo, Japan
aditya@preferred.jp

Vivek B.S.
Indian Institute of Science
Bengaluru, India
svivek@iisc.ac.in

R. Venkatesh Babu
Indian Institute of Science
Bengaluru, India
venky@iisc.ac.in

1. Introduction

This supplementary document is organized as follows:

- Section 2: Analysis of adversarial features
  - Subsection 2.1: Feature Statistics
  - Subsection 2.2: Feature Inversion
- Section 3: Performance of FDA in black-box setting
- Section 4: Ablation study
- Section 5: Comparison of FDA with other existing attack methods
  - Subsection 5.1: Baseline Comparison
  - Subsection 5.2: Evaluation against normally trained models
  - Subsection 5.3: Evaluation against Defense Proposals
  - Subsection 5.4: Evaluation against Defended CIFAR-10 models
- Section 6 : Attacking Feature-Representation based tasks
  - Subsection 6.1 : Attack on Caption generation models
  - Subsection 6.2: Attack on Style transfer models

2. Analysis of adversarial features

2.1. Feature Statistics

Feature Cosine distance: Here, we show the cosine distance between intermediate feature representations of clean and its corresponding adversarial samples generated by our FDA attack. Figure 1 shows the cosine distance plots obtained for models trained on ImageNet [12] dataset.

Dissimilarity metrics: Table 1 shows metrics measuring the dissimilarity between pre-logits of clean and its corresponding adversarial samples, obtained for models trained on ImageNet dataset. These metrics are obtained for different optimization budgets. It can be observed that, our method FDA exhibits stronger dissimilarity.

Table 1. Metrics for measuring the dissimilarity between adversarial pre-logits and clean pre-logits on different networks. Comparison on normally trained models, with the different optimization budgets (ϵ, nb_{iter}, ϵ_{size}). Our method FDA exhibits stronger dissimilarity.

Metrics	Cosine Distance				NRT Distance			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
Optimization budget: (ϵ: 4, nb_{iter}: 5, ϵ_{size}: 1)								
VGG-16	0.49	0.32	0.60	**0.76**	18.17	15.78	20.32	**22.46**
ResNet-152	0.33	0.23	0.40	**0.62**	12.59	11.22	13.52	**16.92**
Inc-v3	0.41	0.33	0.36	**0.51**	14.75	13.31	15.23	**18.72**
IncRes-v2	0.43	0.33	0.33	**0.48**	13.40	11.80	12.43	**15.75**
PNasNet-Large	0.74	0.66	0.68	**0.83**	23.84	22.44	23.65	**26.22**
Optimization budget: (ϵ: 8, nb_{iter}: 10, ϵ_{size}: 1)								
VGG-16	0.64	0.48	0.81	**0.95**	20.10	18.32	23.34	**24.64**
ResNet-152	0.49	0.37	0.60	**0.81**	15.00	13.56	16.29	**19.17**
Inc-v3	0.51	0.41	0.49	**0.55**	16.11	14.97	17.38	**18.99**
IncRes-v2	0.49	0.41	0.48	**0.50**	14.82	13.31	15.10	**16.24**
PNasNet-Large	0.81	0.75	0.82	**0.85**	25.01	23.62	25.66	**26.79**
Optimization budget: (ϵ: 16, nb_{iter}: 20, ϵ_{size}: 2)								
VGG-16	0.67	0.52	0.83	**0.98**	20.42	19.18	23.90	**24.76**
ResNet-152	0.54	0.40	0.62	**0.84**	15.74	14.05	16.76	**19.66**
Inc-v3	0.56	0.43	0.53	**0.57**	16.46	15.26	17.75	**19.05**
IncRes-v2	0.51	0.42	**0.54**	0.50	15.08	13.59	15.87	**16.33**
PNasNet-Large	0.84	0.77	**0.87**	0.85	25.23	23.92	26.14	**27.04**

2.2. Feature Inversion

While feature inversion has a long history in machine learning, we restrict ourselves to only present the formulation presented by Mahendran *et al.* [10]. Feature inversion can be summarized as the problem of finding the sample whose representation is the closest match to a given representation [16]. More formally, given a representation function $\psi : R^{h \cdot w \cdot c} \rightarrow R^d$, we find an input x_I, such that:

$$x_I = \underset{x \in (h \cdot w \cdot c)}{\arg\min} \big(l(\psi(x), \psi(x_I)) + \lambda R(x_I) \big) \tag{1}$$

*Work done as a member of the Video Analytics Lab, IISc, India

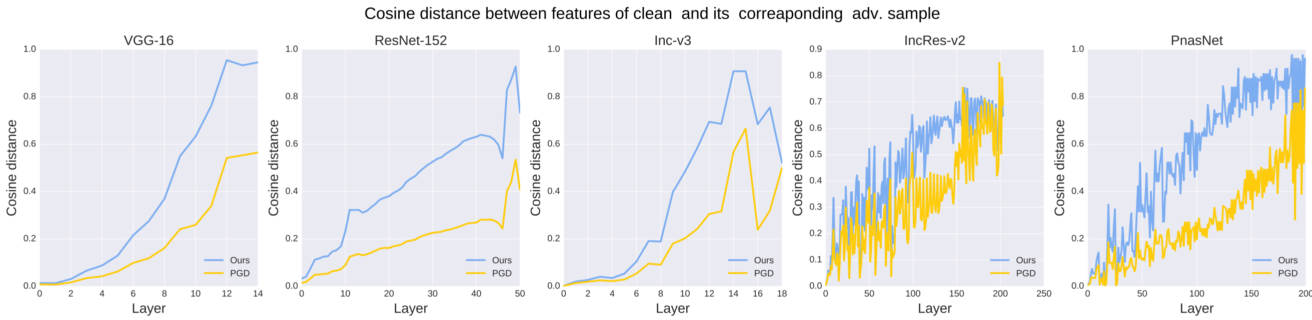

Figure 1. Cosine distance between features of clean image and its corresponding adversarial sample, at different layer of Column-1: VGG-16, Column-2: ResNet-152, Column-3: Inc-v3, Column-4: IncRes-v2, and Column-5: PNASNet.

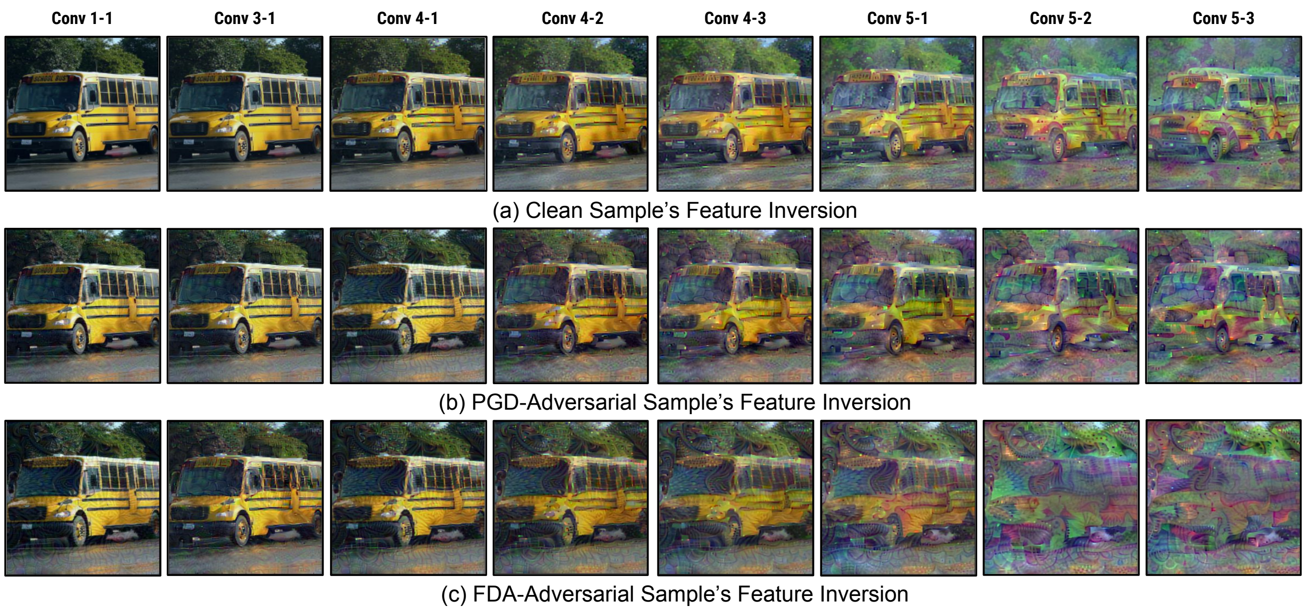

Figure 2. Feature Inversion: Layer-by-layer Feature Inversion [10] of clean, PGD-adversarial and *FDA*-adversarial sample. Note the complete removal of clean sample information in later layers of *FDA*-adversarial sample.

where, l captures the dissimilarity of $\psi(x)$ and $\psi(x_I)$, and R representation regularization used to induce natural image priors in $\hat{x}$.

For deep feature representations, this objective is highly ill-posed due to existence of multiple solutions. [10] propose utilizing TV norm minimization and l_6 normalization as regularizers while using l_2 distance, or euclidean distance for reconstruction. Inclusion of multiple loss terms lead to extensive requirement of hyperparameter tuning, which can be different for the different layers. Furthermore, for deep features, the gradients remain noisy even with the regularization and lead to poor feature inversion. We address these drawbacks by introducing two innovations:

- **Weak/Noisy Gradients from Deep features**: While inverting deep features, it is observed that reconstructed input $\hat{x}$ mostly contains high frequencies. Hence while $\hat{x}$ achieves low dissimilarity error, it remains uninterpretable for the human eye. One way to circumvent this issue is by normalizing the gradients by boosting the low frequency components and decreasing the high frequency components. Following common practices in *Deep Dream* [1], we utilize Laplacian pyramid gradient normalization (LaPGN) for normalizing our gradients.

- **Extensive Hyperparameter Tuning**: We observe that proper weighting of the combined objective 1 becomes even more critical after applying LaPGN, as gradients from one or more of the objectives can be completely lost due to poor weighting scheme. Hence, we instead separately normalize the gradients of each objective, and utilize a weighted combination of these gradients. This allows the optimization to be more stable with

Table 2. Performance of proposed attack in black-box setting, measured in terms of Fooling Rate (FR↑). For all attack methods, optimization budget is set to (ϵ=16, nb_{iter}=10, ϵ_{step}=2). * Method is designed for black-box setting.

Source Model	Attack	Target Model	
		Inception-v3	PNASNet
VGG-16	PGD	68.10	65.60
	PGD-LL	9.50	4.20
	PGD-CW	37.30	30.30
	MI-FGSM*	90	88
	FDA (ours)	90.90	85.30
ResNet-152	PGD	32.50	25.70
	PGD-LL	9.4	3.60
	PGD-CW	20.60	15.80
	MI-FGSM*	61	52
	FDA (ours)	56.60	44.10

respect to the weighting scheme, allowing us to use only a single weighting scheme for each network. Additionally, we remove the L6 normalization objective, and use ADAM optimizer in our algorithm. In Figure 2, we show an example of feature-inversion of adversarial samples at multiple intermediate layers for VGG-16.

3. Performance of FDA in black-box setting

In this section, we compare the performance of FDA with other existing attacks in black-box setting (i.e., limited or no information of the target model is available to the attacker). Table 2 shows the obtained plot. Source model is used for generating adversarial samples and these samples are tested on target model. From table 2, it can observed that the Fooling rate (FR), NLOR and OLNR of FDA attack is better PGD attacks and is on par with MI-FGSM attack. Note that MI-FGSM attack is designed for black-box setting.

4. Ablation study

In this section we show results for the proposed attack with different choices of C (measure of central tendency), in white-box setting. Table 3 shows the obtained results. It can be observed that for C as median and variance, there is drop in the Fooling Rate (FR), OLNR and NLOR. Whereas, FDA with C as spatial-mean achieves consistent performance (i.e. FR, OLNR and NLOR) across different networks.

5. Comparison of FDA with other existing attack methods

5.1. Baseline comparisons

In this subsection, we provide results for baseline attack formulations. We modify GD-UAP [11] (GD-UAP$_{mod}$) to perform image specific attack, and we also modify PGD-CW [9] attack (PGD-CW-LL) in order to boost the confidence of least likely predicted class. Table 4 presents the comparison of baseline attack formulations with the proposed attack(FDA). It can be observed that for all the three metrics FDA achieves superior performance.

Table 3. Performance of proposed attack for different choices of C (measure of central tendency) in white-box setting. The optimization budget is set to (ϵ=4, nb_{iter}=5, ϵ_{step}=1)

C	Inception-v3			PNASNet		
	FR	NLOR	OLNR	FR	NLOR	OLNR
Mean	100	540	663	99	485	516
Spatial Mean (FDA)	100	553	693	99	502	521
Median	85	201.69	122.63	73	78.6	21.99
Spatial Median	95	221	157	100	503	515
Variance	47	41	19	51	46.92	13.7
Spatial Variance	48	40	13	49	36	49

5.2. Evaluation against normally trained models

In this subsection, we compare the performance of various attacks on normally trained models, for different optimization budgets i.e., (ϵ, nb_{iter}, ϵ_{iter}). Table 5 shows the performance of various attacks, it can observed that FDA achieves superior performance in all the three metrics.

5.3. Evaluation against Defense Proposals

We now present the exhaustive set of experiments we conducted to perceive the effectiveness of *FDA* in the presence of various defense proposals. As observed previously, *FDA* is found to be consistently stronger than previous state-of-the-art attack formulations.

5.3.1 Adversarially trained models

We evaluate various attack formulations on adversarially trained models, namely, Simple (adv) [7], Ensemble ($ens3$) [13] and Adversarial-logit-pairing (alp) [5] based adversarially trained models.We present results with different *optimization budgets*, specified by the tuple (l_∞ bound, No. iterations, step-size).

ens and adv models: Table 6 presents the comparison. Note that we evaluate low iteration methods on these models as the authors only claim robustness to single/two iteration white-box attacks.

alp model: Table 7 presents the comparison. It can be observed that our attack has high performance on all metrics at the same time.

5.3.2 Input Defense

We evaluate various attack formulations on models that are defended by input transformation [4], and randomization [17] methods. Table 8, 9 and 10 shows the performance of various attack on defended Inc-v3, IncRes-v2 and PNASNet models respectively. It can be observed that our FDA attack not only achieves higher fooling rate but also higher NLOR and OLNR.

Table 4. Evaluation of various baseline attack formulations. Evaluation on normally trained models, with the optimization budget (ϵ=8, nb_{iter}=10, ϵ_{size}=2).

Metrics	Fooling Rate			NLOR			OLNR		
	GD-UAP$_{mod}$	PGD-CW-LL	Ours	GD-UAP$_{mod}$	PGD-CW-LL	Ours	GD-UAP$_{mod}$	PGD-CW-LL	Ours
	Optimization budget: (ϵ: 8, nb_{iter}: 10, ϵ_{size}: 1)								
VGG-16	99.90	**100.00**	**100.00**	638.46	91.97	**976.82**	585.14	454.63	**878.88**
ResNet-152	76.10	99.90	**100.00**	173.68	42.39	**968.55**	114.76	420.96	**685.81**
Inc-V3	82.63	**100.00**	**100.00**	316.28	128.33	**951.76**	190.36	698.87	**768.00**
IncRes-v2	46.29	99.60	**100.00**	226.43	241.96	**836.34**	71.53	687.69	**709.43**
PNasNet-Large	48.20	99.00	**100.00**	313.51	310.36	**795.42**	190.22	662.94	**720.11**

Table 5. Evaluation of various attacks. Comparison on normally trained models, with the different optimization budgets (ϵ, nb_{iter}, ϵ_{size}). The salient feature of our attack is high performance on all metrics at the same time.

Metrics	Fooling Rate				NLOR				OLNR			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
	Optimization budget: (ϵ: 4, nb_{iter}: 5, ϵ_{size}: 1)											
VGG-16	**99.90**	**99.90**	93.80	97.80	57.26	6.17	**539.92**	433.33	308.34	29.19	217.98	**455.26**
ResNet-152	99.50	**99.60**	88.15	97.69	20.62	5.12	**593.64**	412.52	247.22	21.84	89.58	**380.04**
Inc-v3	99.20	99.10	89.06	**99.80**	61.73	21.95	**599.49**	549.57	524.65	63.86	92.45	**669.31**
IncRes-v2	94.18	94.58	74.30	**99.60**	75.43	44.51	314.20	**492.95**	314.14	44.46	67.02	**487.76**
PNasNet-Large	92.60	92.40	81.40	**99.00**	123.93	59.44	319.18	**473.54**	335.63	70.67	118.73	**512.21**
	Optimization budget: (ϵ: 8, nb_{iter}: 10, ϵ_{size}: 1)											
VGG-16	99.90	**100.00**	99.70	**100.00**	88.25	37.36	**976.82**	714.77	452.81	90.48	558.70	**878.88**
ResNet-152	99.90	99.90	99.70	**100.00**	40.54	33.04	**968.55**	593.66	426.82	85.20	306.66	**685.81**
Inc-v3	99.90	99.80	99.10	**100.00**	126.51	70.98	**951.76**	580.88	670.50	133.67	326.74	**768.00**
IncRes-v2	99.30	99.30	96.79	**100.00**	222.39	109.46	**826.34**	553.84	605.43	104.77	355.49	**709.43**
PNasNet-Large	99.30	99.00	95.90	**100.00**	270.75	127.90	**795.42**	596.23	571.44	150.68	459.09	**720.11**
	Optimization budget: (ϵ: 16, nb_{iter}: 20, ϵ_{size}: 2)											
VGG-16	**100.00**	**100.00**	**100.00**	**100.00**	79.36	23.86	**997.88**	770.59	465.62	73.74	635.57	**926.46**
ResNet-152	99.90	99.90	99.90	**100.00**	39.96	13.80	**990.84**	607.49	452.23	68.07	357.82	**726.12**
Inc-v3	99.90	99.90	99.90	**100.00**	98.08	67.35	**996.39**	615.85	754.56	136.24	439.37	**816.89**
IncRes-v2	99.70	**100.00**	99.90	**100.00**	202.88	100.65	**983.49**	570.76	705.94	113.30	552.66	**757.67**
PNasNet-Large	99.80	99.90	99.80	**100.00**	238.84	102.42	**986.36**	605.15	597.22	143.07	645.25	**771.47**

5.4. Evaluation against Defended CIFAR-10 models

In this subsection, we show the performance of various attacks on defended models that are trained on CIFAR-10 [6] dataset. Table 11 and 12 shows the effectiveness of various attack formulation in white-box and grey-box attack settings respectively.

6. Attacking Feature-Representation based tasks

6.1. Attack on Caption generation models

In this subsection, we show the effectiveness of our attack FDA in grey-box setting. We perform ”grey-box” attack on “Show-and-Tell(SAT) [15], with different optimization budgets. Table 13 present the performance of various attack formulations. The right-most column tabulates the metrics when complete white noise is given as input. It can be observed that FDA adversaries generated from Inception-V3 are highly effective for disrupting SAT.

6.2. Attack on Style transfer models

In this subsection, we provide qualitative results to show the effectiveness of FDA. Figure 3 shows the effectiveness of FDA attack on style transfer networks [14], column-1 represents the style images, the 1^{st} image in column 2 and 3 represents the content image. The 2^{nd} and the 3^{rd} image of column-2 and 3 represents the output of style transfer network with and without FDA attack respectively. It can be observed that due to FDA attack, content of stylized image is severely damaged.

Furthermore, we have added examples of attack on video clips as well, which are the primary use-case for Fast-style-

Table 6. Evaluation of various attacks. Comparison on adversarially trained models (adv & ens), with the different budgets. The salient feature of our attack is high performance on all metrics at the same time.

Metrics	Fooling Rate				NLOR				OLNR			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
	Optimization budget: (ϵ: 8, nb_{iter}: 1, ϵ_{size}: 8)											
Inc-V3$_{adv}$	11.45	**11.75**	8.43	9.14	3.56	**3.97**	3.44	3.43	3.49	3.64	**4.57**	3.54
Inc-V3$_{ens3}$	**57.23**	56.83	40.56	35.94	26.85	21.91	42.76	**76.44**	32.63	24.19	31.28	**34.11**
IncRes-V2$_{adv}$	**6.02**	5.92	4.62	5.02	2.22	**2.95**	1.41	2.22	**1.87**	1.61	1.76	1.52
IncRes-V2$_{ens3}$	43.47	**45.88**	24.20	26.41	8.54	9.01	16.50	**59.90**	**19.50**	14.25	15.69	12.98
	Optimization budget: (ϵ: 8, nb_{iter}: 2, ϵ_{size}: 4)											
Inc-V3$_{adv}$	88.45	**88.45**	53.01	86.45	33.55	19.96	81.38	**271.80**	151.51	21.87	27.12	**159.84**
Inc-V3$_{ens3}$	**94.08**	91.97	66.37	93.47	74.65	44.99	152.62	**353.29**	229.49	55.78	78.35	**264.86**
IncRes-V2$_{adv}$	66.77	69.48	37.75	**72.79**	28.42	23.10	56.44	**245.70**	92.66	22.29	19.45	**98.87**
IncRes-V2$_{ens3}$	73.49	73.80	44.38	**82.83**	39.72	29.87	75.86	**303.81**	107.04	22.25	19.98	**146.06**
	Optimization budget: (ϵ: 8, nb_{iter}: 5, ϵ_{size}: 2)											
Inc-V3$_{adv}$	97.89	97.69	80.62	**99.70**	68.03	34.56	346.59	**545.89**	281.75	39.08	77.80	**629.93**
Inc-V3$_{ens3}$	98.69	97.49	88.76	**100.00**	114.96	68.76	450.66	**533.49**	386.16	106.58	142.65	**634.55**
IncRes-V2$_{adv}$	91.27	89.66	61.65	**99.70**	81.80	39.68	284.36	**504.51**	234.66	33.20	67.27	**571.46**
IncRes-V2$_{ens3}$	98.69	97.49	88.76	**100.00**	114.96	68.76	450.66	**533.49**	386.16	106.58	142.65	**634.55**

Table 7. Evaluation of various attacks. Comparison on adversarially trained models (alp), with the different budgets. The salient feature of our attack is high performance on all metrics at the same time.

Metrics	Fooling Rate				NLOR				OLNR			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
	Optimization budget: (ϵ: 8, nb_{iter}: 2, ϵ_{size}: 4)											
Res-50$_{alp}$	77.91	79.72	51.00	**80.32**	19.78	12.45	21.05	**120.11**	39.04	13.01	14.56	**79.95**
	Optimization budget: (ϵ: 8, nb_{iter}: 5, ϵ_{size}: 2)											
Res-50$_{alp}$	85.04	**87.15**	51.10	80.02	22.28	10.83	20.60	**119.41**	77.55	11.14	14.90	**81.73**
	Optimization budget: (ϵ: 16, nb_{iter}: 10, ϵ_{size}: 2)											
Res-50$_{alp}$	96.99	**98.29**	64.56	94.28	41.51	12.26	77.40	**259.78**	**302.03**	14.97	25.66	241.43
	Optimization budget: (ϵ: 16, nb_{iter}: 20, ϵ_{size}: 2)											
Res-50$_{alp}$	96.69	**98.39**	64.86	94.78	46.07	12.21	89.22	**257.09**	**325.30**	14.33	24.54	238.07

transfer. While a viewer may still like the style on the attacked videos, we emphasize that the fundamental drawback to be noted is the lack of fine-object details and object edges.

Table 8. Evaluation of various attacks in the presence of input transformation based defense measures with different optimization budgets on **Inception-V3**. While achieving higher fooling rate, we also achieve higher *NLOR* and *OLNR*.

Metrics	**Fooling Rate**				**NLOR**				**OLNR**			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
	Optimization budget: (ϵ: 8, nb_{iter}: 5, ϵ_{step}: 2)											
Gaussian Filter	**81.73**	39.76	74.20	77.81	35.80	47.77	26.39	**263.90**	40.34	19.86	10.77	**144.37**
Median Filter	**54.52**	28.01	47.69	52.51	16.85	21.28	16.61	**87.50**	15.29	12.85	8.36	**54.51**
Bilateral Filter	**50.90**	22.09	42.37	39.06	9.29	9.49	6.35	**38.32**	4.84	4.10	2.63	**23.12**
Bit Quant.	56.22	35.14	46.69	**60.34**	15.16	18.97	16.22	**78.57**	13.61	14.37	11.70	**50.17**
JPEG Comp.	**68.78**	27.21	61.85	52.41	16.76	12.95	10.06	**114.78**	12.82	5.86	3.52	**47.42**
TV Min.	**34.64**	21.59	30.82	34.34	5.16	3.59	3.52	**19.49**	3.85	3.40	2.59	**13.94**
Quilting	30.82	21.49	29.02	**33.43**	4.64	3.79	3.78	**7.14**	4.78	3.93	4.15	**9.21**
Randomize [17]	79.82	42.97	71.99	**84.74**	53.46	69.15	38.03	**312.11**	58.53	24.41	13.32	**208.26**
	Optimization budget: (ϵ: 16, nb_{iter}: 10, ϵ_{step}: 2)											
Gaussian Filter	88.76	46.49	81.93	**92.77**	79.10	139.25	43.65	**413.13**	76.62	43.58	16.74	**309.66**
Median Filter	62.25	35.44	54.72	**70.38**	35.81	34.82	28.45	**182.71**	33.19	32.38	16.39	**113.93**
Bilateral Filter	**67.27**	28.51	55.12	64.86	23.61	11.71	17.46	**132.41**	9.11	9.57	4.78	**78.99**
Bit Quant.	77.61	48.09	69.28	**87.95**	51.51	63.19	41.79	**278.16**	48.34	28.16	23.45	**224.75**
JPEG Comp.	81.93	35.84	73.49	**84.44**	46.89	49.61	28.08	**281.45**	35.32	15.31	8.78	**168.33**
TV Min.	50.00	27.41	40.36	**55.82**	12.99	11.42	12.24	**61.72**	7.96	6.73	4.95	**42.84**
Quilting	41.27	29.72	34.34	**46.59**	7.80	10.31	7.07	**37.21**	7.03	8.15	5.88	**22.47**
Randomize [17]	83.03	50.80	77.61	**93.37**	80.15	141.81	51.16	**411.57**	82.55	38.32	22.81	**321.25**

Table 9. Evaluation of various attacks in the presence of input transformation based defense measures with different optimization budgets on **Inception-Resnet-V2**. While achieving higher fooling rate, we also achieve higher *NLOR* and *OLNR*.

Metrics	**Fooling Rate**				**NLOR**				**OLNR**			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
	Optimization budget: (ϵ: 8, nb_{iter}: 5, ϵ_{step}: 2)											
Gaussian Filter	73.80	28.11	61.75	**75.40**	41.26	46.77	26.40	**325.66**	31.96	18.37	9.36	**156.10**
Median Filter	43.37	15.36	32.53	**49.80**	18.62	14.99	17.19	**147.32**	11.70	18.33	9.60	**64.03**
Bilateral Filter	**41.47**	12.85	28.41	36.45	8.44	5.47	7.23	**73.85**	6.04	5.68	3.43	**28.70**
Bit Quant.	52.81	26.91	40.06	**64.56**	20.66	20.38	27.03	**137.10**	9.49	11.15	8.06	**79.52**
JPEG Comp.	**68.78**	21.18	55.32	67.27	25.32	20.59	19.48	**235.64**	17.62	8.67	6.22	**85.50**
TV Min.	27.61	10.24	18.78	**29.32**	5.21	3.38	2.86	**40.55**	6.28	5.19	4.52	**18.43**
Quilting	27.71	16.57	22.69	**37.45**	5.57	3.96	8.52	**40.13**	4.30	3.67	3.15	**16.82**
Randomize [17]	76.51	32.63	60.84	**86.04**	71.98	97.71	46.74	**369.13**	49.01	20.23	16.67	**245.00**
	Optimization budget: (ϵ: 16, nb_{iter}: 10, ϵ_{step}: 2)											
Gaussian Filter	81.93	36.95	68.57	**92.87**	74.59	133.03	34.52	**443.16**	63.44	27.98	12.40	**364.81**
Median Filter	50.40	23.19	38.45	**70.88**	34.75	24.49	20.36	**238.69**	27.03	19.07	14.40	**139.86**
Bilateral Filter	54.52	19.18	41.47	**70.18**	23.48	15.21	13.47	**217.54**	14.20	10.69	7.18	**94.56**
Bit Quant.	73.90	40.86	62.05	**91.77**	71.64	68.30	51.65	**363.12**	40.80	27.58	18.65	**328.54**
JPEG Comp.	79.82	31.83	66.67	**96.18**	55.99	70.58	37.38	**418.41**	41.44	16.75	9.15	**342.41**
TV Min.	38.96	17.67	27.81	**55.72**	12.53	10.10	8.76	**130.38**	10.27	8.36	5.55	**63.08**
Quilting	38.35	24.10	30.82	**56.63**	17.95	9.95	9.54	**121.63**	7.64	7.18	5.39	**62.95**
Randomize [17]	81.93	42.87	68.17	**98.19**	114.94	140.35	70.11	**469.98**	84.24	26.48	26.76	**430.47**

Table 10. Evaluation of various attacks in the presence of input transformation based defense measures with different optimization budgets on **PNASNet** [8]. While achieving higher fooling rate, we also achieve higher *NLOR* and *OLNR*.

Metrics	**Fooling Rate**				**NLOR**				**OLNR**			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
	Optimization budget: (ϵ: 8, nb_{iter}: 5, ϵ_{size}: 2)											
Gaussian Filter	**83.40**	45.00	73.30	74.80	88.16	136.01	51.86	**346.85**	73.78	21.09	18.78	**170.85**
Median Filter	**65.00**	29.70	52.60	63.60	48.60	60.70	42.27	**229.15**	30.60	17.92	9.37	**83.53**
Bilateral Filter	**52.70**	21.60	41.10	43.10	22.85	22.16	18.26	**113.25**	12.17	5.65	6.26	**39.93**
Bit Quant.	46.20	29.10	38.40	**56.70**	19.51	20.89	20.11	**119.46**	11.34	13.36	7.22	**54.29**
JPEG Comp.	**69.50**	28.20	58.90	57.40	42.09	39.40	29.06	**192.71**	22.77	7.88	7.00	**62.46**
TV Min.	34.50	17.60	24.80	**36.50**	9.72	16.88	9.15	**51.40**	6.78	6.11	4.08	**30.33**
Quilting	29.90	19.30	24.40	**37.20**	16.69	10.54	8.16	**45.48**	12.06	6.37	7.11	**21.10**
Randomize [17]	82.60	52.40	71.90	**91.50**	117.89	183.05	69.07	**450.78**	113.73	64.16	29.96	**349.79**
	Optimization budget: (ϵ: 16, nb_{iter}: 10, ϵ_{size}: 2)											
Gaussian Filter	86.90	49.30	80.60	**91.10**	126.30	275.02	82.44	**453.84**	128.44	61.76	31.40	**332.29**
Median Filter	71.00	33.40	61.30	**79.60**	70.47	98.60	45.35	**335.62**	51.50	32.98	16.27	**182.40**
Bilateral Filter	69.20	31.50	57.20	**78.80**	52.06	61.86	32.17	**284.45**	32.54	14.02	12.27	**157.42**
Bit Quant.	70.00	42.80	63.40	**88.50**	69.75	83.59	37.30	**342.21**	49.21	32.42	20.12	**242.70**
JPEG Comp.	84.10	40.10	74.00	**92.20**	91.62	116.03	57.73	**387.58**	66.08	24.37	16.65	**240.50**
TV Min.	48.70	24.30	39.70	**62.50**	21.32	25.24	25.65	**150.21**	14.89	13.83	10.26	**89.05**
Quilting	40.10	26.00	32.90	**56.10**	20.52	22.51	24.37	**122.35**	10.78	12.70	7.61	**52.33**
Randomize [17]	85.40	55.80	78.70	**98.70**	163.96	281.98	97.14	**507.89**	163.88	99.48	48.26	**489.07**

Table 11. Evaluation of various attacks in the presence of defense measures on CIFAR-10 dataset. We show evaluation at multiple optimization budgets. While achieving lower fooling rate at times, we achieve higher *NLOR* and *OLNR*.

Metrics	**Fooling Rate**				**NLOR**				**OLNR**			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
	Optimization budget: (ϵ: 8, nb_{iter}: 10, ϵ_{size}: 1)											
Normal Model	62%	**100%**	74%	**100%**	2.94	2.08	5.51	**6.51**	3.12	7.00	4.87	**8.32**
Madry *et al.* [9]	20%	**29%**	13%	7%	2.25	2.00	2.19	**3.72**	2.21	2.47	2.24	**2.88**
Dhillon *et al.* [3]	56%	**75%**	43%	57%	2.56	2.35	3.20	**5.12**	2.67	3.58	3.26	**4.64**
Buckman *et al.* [2]	-	**28%**	6%	14%	-	2.14	2.23	**2.72**	-	2.38	2.25	**2.56**
	Optimization budget: (ϵ: 16, nb_{iter}: 20, ϵ_{size}: 1)											
Normal Model	71%	**100%**	85%	**100%**	3.96	2.16	5.86	**6.96**	4.12	8.68	5.71	**8.97**
Madry *et al.* [9]	39%	**70%**	25%	32%	2.73	2.06	3.38	**5.83**	2.54	**4.64**	3.14	4.23
Dhillon *et al.* [3]	71%	**100%**	75%	98%	3.65	2.51	5.51	**6.66**	3.9	6.78	5.12	**6.98**
Buckman *et al.* [2]	0%	**78%**	14%	69%	-	2.35	2.50	**5.00**	-	**4.18**	2.57	3.89

Table 12. Evaluation of various attacks in the presence of defense measures on CIFAR-10 dataset in a "Grey-box" setting, where the attacker is not aware of the defense mechanism. We show evaluation at multiple optimization budgets. While achieving lower fooling rate at times, we achieve higher *NLOR* and *OLNR*.

Metrics	**Fooling Rate**				**NLOR**				**OLNR**			
	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours	PGD-ML	PGD-CW	PGD-LL	Ours
	Optimization budget: (ϵ: 8, nb_{iter}: 10, ϵ_{size}: 1)											
Dhillon *et al.* [3]	32%	**99%**	75%	**99%**	2.73	2.57	5.51	**5.80**	2.91	5.11	5.12	**7.43**
	Optimization budget: (ϵ: 16, nb_{iter}: 20, ϵ_{size}: 1)											
Dhillon *et al.* [3]	60%	**100%**	55%	**100%**	3.62	2.53	4.40	**6.53**	3.93	8.44	3.91	**8.90**

Table 13. Attacking "Show-and-Tell"(SAT) [15] in a "Grey-box" setup with different optimization budgets. The right-most column tabulates the metrics when complete white noise is given as input. *FDA* Adversaries generated from Inception-V3 are highly effective for disrupting SAT.

Metrics	No Attack	PGD-ML	PGD-LL	MI-FGSM	Ours	PGD-ML	PGD-LL	MI-FGSM	Ours	PGD-ML	PGD-LL	MI-FGSM	Ours	Noise
		(4, 5, 1)				(8, 10, 1)				(16, 20, 1)				
CIDEr	103.21	71.72	80.41	63.20	**16.33**	47.95	47.13	49.23	**4.90**	35.25	23.58	38.33	**3.35**	2.84
Blue-1	71.61	63.87	65.80	60.95	**46.33**	57.04	55.68	57.18	**39.80**	52.41	48.27	53.56	**38.29**	37.60
Rough$_L$	53.61	47.01	48.72	45.15	**34.56**	42.15	41.24	42.65	**30.70**	39.03	36.37	39.75	**29.71**	29.30
METEOR	25.58	20.88	22.02	19.56	**11.93**	17.50	16.78	17.34	**10.02**	15.15	12.92	15.70	**8.86**	7.84
SPICE	18.07	13.56	15.04	12.13	**4.28**	9.60	9.45	10.02	**2.04**	7.68	5.68	8.44	**1.71**	1.00

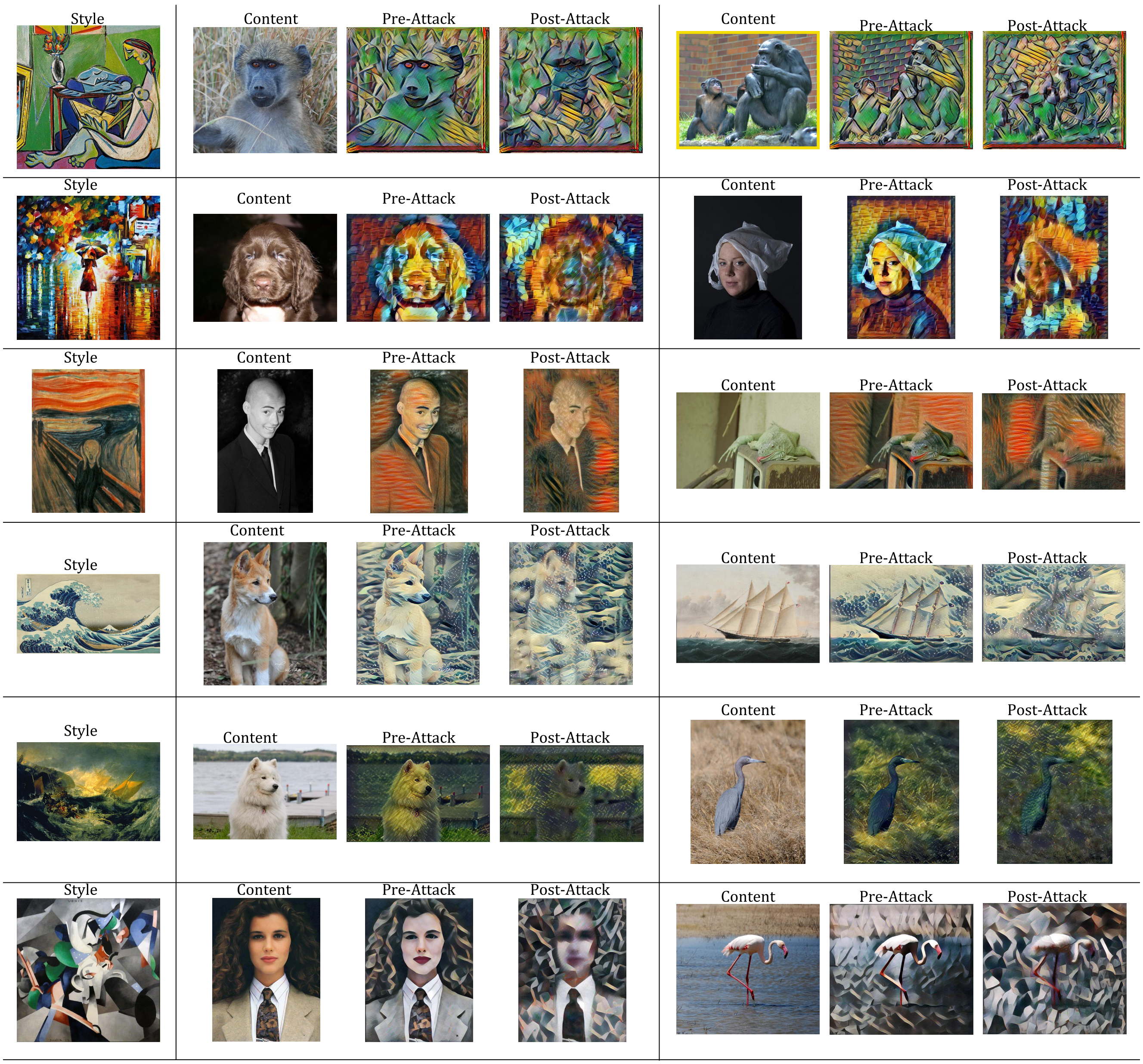

Figure 3. Multiple Examples of style transfer using Ulyanov *et al*. [14]'s approach. The attacked samples are severely degraded.

References

[1] Inceptionism: Going deeper into neural networks. https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html. 2

[2] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. 7

[3] G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018. 7

[4] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. 3

[5] H. Kannan, A. Kurakin, and I. J. Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018. 3

[6] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 4

[7] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 3

[8] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *The European Conference on Computer Vision (ECCV)*, September 2018. 7

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 3, 7

[10] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2

[11] K. R. Mopuri, A. Ganeshan, and R. V. Babu. Generalizable data-free objective for crafting universal adversarial perturbations. 2018. 3

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[13] F. Tramr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. 3

[14] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4105–4113. IEEE Computer Society, 2017. 4, 8

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):652–663, April 2017. 4, 8

[16] R. J. WILLIAMS. Inverting a connectionist network mapping by backpropagation of error. *Proc. of 8th Annual Conference of the Cognitive Science Society*, pages 859–865, 1986. 1

[17] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 3, 6, 7