Supplementary for "Meta-Attack: Class-agnostic and Model-agnostic Physical Adversarial Attack"

Weiwei Feng[1], Baoyuan Wu[2,3,†], Tianzhu Zhang[1,†], Yong Zhang[4], Yongdong Zhang[1]
[1] University of Science and Technology of China
[2]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[3]Shenzhen Research Institute of Big Data, Shenzhen, China [4] Tencent AI Lab

1. Exp.2 on GTSRD

1.1. Configurations

Setting: Attack an seen model on images of an unseen class. Here we present the Exp.2 on GTSRD. Firstly, following the detailed instructions in **Section 4.2** of the paper, we configure the hyperparameter C to 5, which represents the number of the image classes sampled from GTRSD dataset (label list [2 (Speed limit 50km/h), 5 (Speed limit 80km/h), 9 (No passing), 36 (Go straight or right), 37 (Go straight or left)]). The value of the parameter M is also set to 30, which means the fine-tune steps.

1.2. Results

The results of **Exp.2** on GTSRD are reported in Table 1 and Figure 1, which present the generalization and robustness of adversarial examples on GTSRD dataset. From Table 1, it shows our proposed model can effectively generate robust adversarial examples on an unseen class and get an ASR with a high value. In general, based on Table 1, the average ASR can surprisingly achieve 78.5% in the digital domain and 63.0% in the physical domain.

The robustness of the adversarial examples from various unseen classes is also evaluated by changing different spatial transformations. Figure 1 shows that when our adversarial examples undergo spatial transformations, the fluctuation of ASR is very small, which shows the superior robustness of our adversarial examples. We infer that the superior robustness of our adversarial examples is mainly because of the contributions of EOT [1].

Table 1. Experimental results of Exp.2 on GTSRD.

Domain →	Digital		Physical	
Source Label ↓	ASR	Conf	ASR	Conf
2 (Speed limit 50km/h)	0.877	0.846	0.821	0.798
5 (Speed limit 80km/h)	0.800	0.589	0.705	0.550
9 (No passing)	0.667	0.478	0.595	0.467
36 (Go straight or right)	0.722	0.412	0.417	0.258
37 (Go straight or left)	0.861	0.670	0.611	0.515
ave	0.785	0.599	0.630	0.518

† indicates corresponding authors. This work corresponds to wubaoyuan@cuhk.edu.cn and tzzhang@ustc.edu.cn

Table 2. Results of adapting to different attacked models under different Spatial tansformations on GTSRD.

Attacked Model →	VGG-16		VGG-19		ResNet-50	
Spatial Transformation ↓	ASR	Conf	ASR	Conf	ASR	Conf
Digital domain	0.878	0.866	0.833	0.821	0.742	0.727
Resize 1 + Rotation 0°	0.833	0.832	0.750	0.754	0.333	0.294
Resize 1 + Rotation 20°	0.817	0.819	0.716	0.719	0.317	0.282
Resize 1 + Rotation −20°	0.817	0.817	0.733	0.739	0.317	0.283
Resize 1.2 + Rotation 0°	0.833	0.780	0.716	0.718	0.333	0.297
Resize 1.2 + Rotation 20°	0.833	0.778	0.706	0.707	0.333	0.297
Resize 1.2 + Rotation −20°	0.817	0.770	0.716	0.718	0.325	0.289
Resize 0.8 + Rotation 0°	0.817	0.791	0.683	0.680	0.308	0.277
Resize 0.8 + Rotation 20°	0.783	0.560	0.683	0.679	0.300	0.265
Resize 0.8 + Rotation −20°	0.783	0.556	0.683	0.680	0.308	0.279

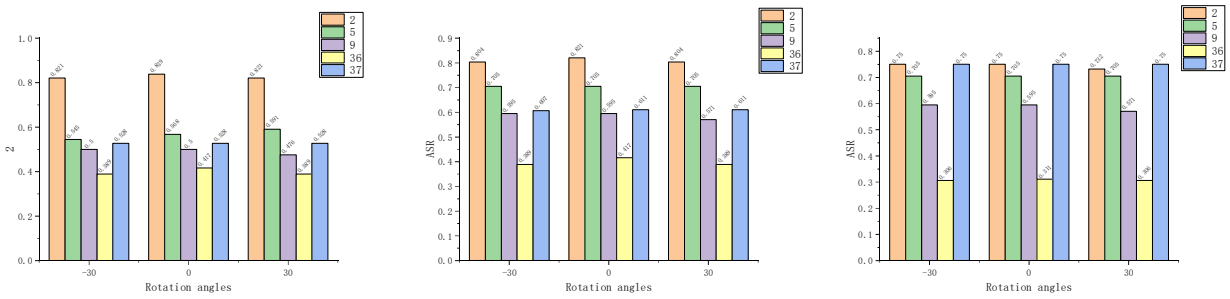
2. Exp.3 on GTSRD

2.1. Configurations

Setting: Attack an unseen model on images of a seen class. We explain the configurations of **Exp.3** on GTSRD, which are exactly the same as those on ImageNet. According to the design of the experiment in **Section 4.2** of the paper, the hyperparameters are specified as follows: $R = 3, K = 50, M = 30$. These 3 target models include VGG-16, VGG-19 and ResNet-50, whose final fully connected layer is modified to adapt to the classfication results of 43 classes. Besides the models are all fine-tuned on GTSRD, and achieve a correct rate of 99.9%.

2.2. Results

Exp.3 is to examine the generalization ability of our proposed method when attacking an unseen DNN model. Table 2 presents the results on GTSRD dataset, which shows that our model performs well on an unseen DNN model. In the digital domain, whether the unseen DNN model is VGG-16, VGG-19 or ResNet-50, our model can achieve an ASR with a high value, which can be achieved after only a few steps of fine-tuning. In particular, when treating VGG-16 as the unseen DNN model, the ASR in the digital domain is still 87.8%. If we replace the attacked DNN model with VGG-19, ASR is also with a high value of 83.3%. Although the physical attack performance degrades to some extent, but the performance is still acceptable. For example, when regarding VGG-19 as the unseen DNN model, and rotating



(a) Resize 0.8 (b) Resize 1.0 (c) Resize 1.2

Figure 1. Adversarial images generated by **Exp.**2 against different spatial transformations on GTSRD dataset.

Table 3. Result of adapting to different attacked classes and different DNN models on GTSRD dataset.

Source Label → / Attacked Model ↓			17	21	35
VGG-16	Digital	ASR	0.833	0.842	0.825
		Conf	0.796	0.813	0.799
	Physical	ASR	0.642	0.625	0.592
		Conf	0.595	0.577	0.512
VGG-19	Digital	ASR	0.85	0.858	0.842
		Conf	0.823	0.814	0.820
	Physical	ASR	0.658	0.65	0.608
		Conf	0.607	0.611	0.576
ResNet-50	Digital	ASR	0.783	0.75	0.758
		Conf	0.741	0.733	0.699
	Physical	ASR	0.283	0.258	0.308
		Conf	0.239	0.219	0.253

the adversarial images by 20 degrees, scaling them by 1.2, the ASR is still 70.6%. The results once again indicate that our generative attack model has good generalization ability on target DNN models and robustness under spatial transformations.

3. Exp.4 on GTSRD

3.1. Configurations

Setting: Attack an unseen model on images of an unseen class. We explain the configurations of **Exp.4** on GTSRD. According to the design of **Exp.4** in **Section 4.2** of the paper, we configure the parameters as follows: $R = 3, M = 30, C = 3$ (label list [17 (No entry), 21 (Double curve), 35 (Ahead only)]). These 3 target models include VGG-16, VGG-19 and ResNet-50, whose final fully connected layer is modified to adapt to the classfication results of 43 classes. Besides the models are all fine-tuned on GTSRD, and achieve a correct rate of 99.9%.

3.2. Results

Figure 3 reports the results of our method when attacking an unseen DNN model on images of an unseen class. It is obviously observed that our method performs well when attacking an unseen DNN model on images of an unseen class. Our model can achieve an ASR with a high value, which can be achieved after only a few steps of fine-tuning. In particular, when treating VGG-16 as the unseen DNN model, 17 as the unseen class, the ASR in the digital domain is still 83.3%, while the physical ASR is 64.2%. If we focus on ResNet-50 and label 35, ASR in the digital domain is also with a high value of 75.8%, and the physical ASR reaches 30.8%. Consistently across all cases, our method has a good performance on different classes and different DNN models, which proves the generalization ability of our method.

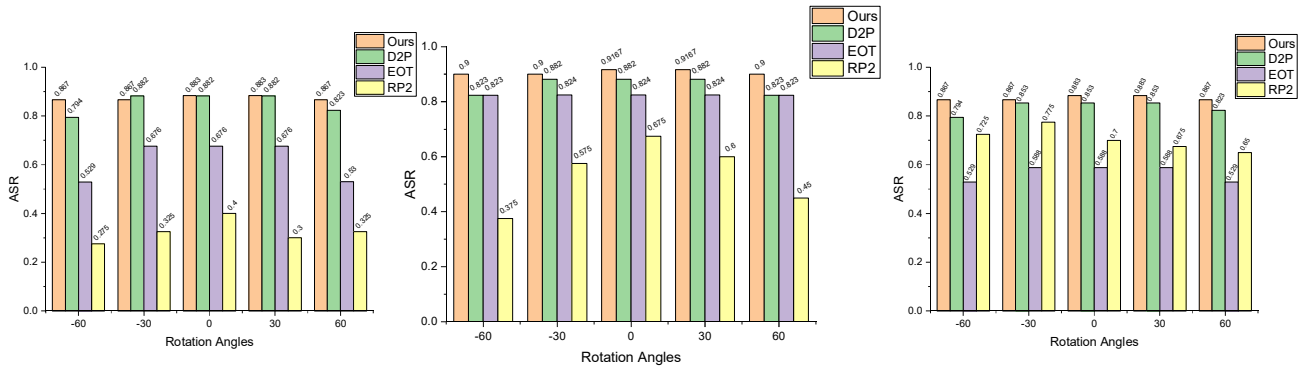
4. Extra Experiments

4.1. Attack on a Single Image

Setting. Follow the conventional methods [5, 2], we use the standard training strategy to learn a generative attack model that is proposed in the paper and perform attack on a single digital image from ImageNet. Given a target image, we firstly scale it to 288×288, then crop it with a 256×256 patch for K times and finally we perform the 1:1-D2P transformation to each cropped image, forming K pairs of digital and physical images. These pairs are used as the training set to learn the generative model by solving the min-max problem Eq. (5). After training, we perform attack on the target image with the learned generative attack model.

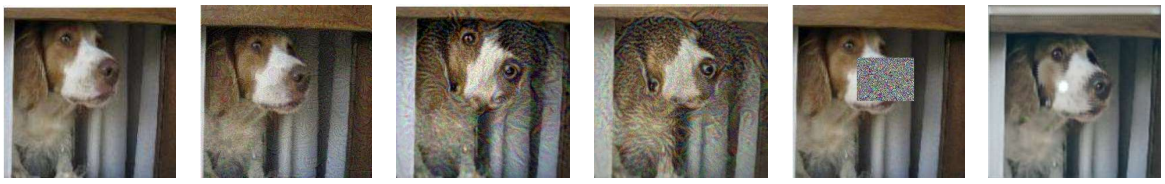
The training procedure can be regarded as an optimization process, which is similar to previous attack methods. Although this strategy can achieve a high attack success rate and generate realistic adversarial examples, it has two drawbacks that limit its broad applications. First, it has unsatisfying efficiency. When attacking each unseen image, we have to collect many physical images of the cropped images, which is tedious and time-consuming. Second, it is not generalizable to an unseen image. The generative attack model trained on one image overfits this specific image, and fails to create an effective adversarial example for an unseen image.

Results. The results are given in Table 4, which shows almost all methods perform well in the digital domain. And even the ASR of all methods can reach 100%. However, in the physical domain, the ASR of our method can reach

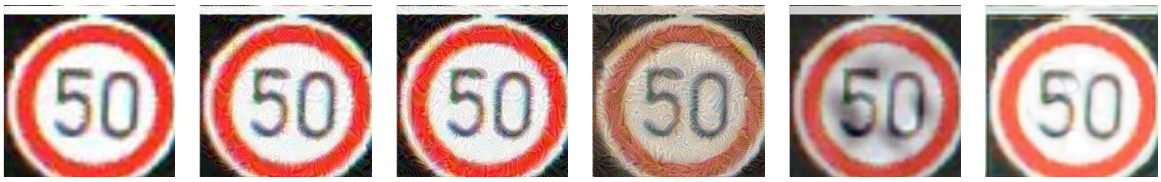


(a) Resize 0.8 (b) Resize 1.0 (c) Resize 1.2

Figure 2. Physical adversarial images generated by different methods against different spatial transformation on ImageNet.



(a) Benign image (b) BIM (c) EOT (d) D2P (e) RP2 (f) Ours



(g) Benign image (h) BIM (i) EOT (j) D2P (k) RP2 (l) Ours

Figure 3. Adversarial images generated by different methods. The classifier mis-classify the input image from the original label "Brittany spaniel" to "Badger" (a)∼(f), (from " SpeedLimit 50" to "SpeedLimit 30" (g)∼(l)) .

Table 4. Results of different methods on ImageNet.

Domain →	Digital		Physical	
Attack↓	ASR	Conf	ASR	Conf
BIM [4]	**1.0**	0.987	0.111	0.049
EOT [1]	**1.0**	0.989	0.824	0.816
D2P [3]	**1.0**	0.992	0.882	**0.880**
RP2 [2]	**1.0**	0.996	0.675	0.317
Ours	**1.0**	**0.998**	**0.917**	0.870

91.7%, which shows the best performance. Besides, the confidence is also not bad, reaching 87.0%.

Then we examine the robustness of the adversarial examples by performing different spatial transformations on them. Figure 2 presents the ASR of the physical adversarial examples generated by the four schemes (EOT [1], D2P [3], RP2[2], Ours) under different spatial transformations. To a certain extent, we can observe that all these methods perform not bad under different spatial transformations. This is mainly because of the great contribution of synthetic geometric transformations (EOT) [1]. And our method can perform better when comparing with others. For example, when the image is scaled by 0.8 times and rotated 60 degrees, our ASR still reaches 86.7%.

To further understand adversarial examples, we replace the capture device (scanner) with a mobile phone. Table 5 shows the results of our evaluation under different spatial rotation transformations. We can perceive that the distances and view angles of the mobile phone are able to affect the effect of capturing images, which is more uncontrollable than 1:1-D2P transformation by a scanner. But our method still has an ASR of 76.9% without any rotation, which achieves state-of-the-art performance. Similarly, after rotating 45 degrees, there is still a 69.2% success rate of our method that can really outperform other methods. In despite of changing different capture devices, our proposed

Table 5. Photographing adversarial images with different rotations on ImageNet.

Attack →	EOT		D2P		RP2		Ours	
Rotation ↓	ASR	Conf	ASR	Conf	ASR	Conf	ASR	Conf
 −45°	0.538	0.212	0.690	0.226	0.154	0.025	**0.700**	**0.332**
 −20°	0.615	0.225	0.640	0.231	0.308	0.044	**0.693**	**0.364**
 0°	0.636	0.243	0.730	0.248	0.615	0.135	**0.769**	**0.396**
 20°	0.620	0.221	0.640	0.230	0.308	0.040	**0.727**	**0.336**
 45°	0.541	0.213	0.690	0.224	0.231	0.026	**0.692**	**0.333**

generative attack model can still perform well.

4.2. Comparative Baselines under Exp2,3,4

1) Firstly, we declare that baselines (BIM, EOT, RP2, D2P) formulate constructing adversarial examples as optimization problems, which are via iterative gradient updating for each image. Besides, baselines don't take into account the settings of unseen classes or unseen models, and they cannot cope with the settings of Exp.2, 3, 4 in the paper. **2)** Different from baselines, our method can perform well on an unseen model and an unseen class with a feed-forward network. Our method only needs to update for few steps on few-shot images of the unseen class instead of iterative gradient updating for each image, which demonstrates a good generalized ability of our method. **3)** We also present extra Table 6, 7, and 8, which show the results of baselines in digital domain on ImageNet under the settings of Exp.2, 3, 4, respectively. To present extra results of Exp.2, 3, 4 for baselines, we use these baselines to generate an perturbation on an image, then apply the perturbation directly to unseen tested images and models. It is observed that baseline methods can hardly attack successfully, and Table 6, 7, and 8 consistently demonstrate the superiority of our method again.

Table 6. Results of baselines on Exp.2 in digital domain based on ImageNet database (unseen class).

Methods	BIM		EOT		RP2		D2P	
Label	ASR	Conf	ASR	Conf	ASR	Conf	ASR	Conf
288	0.075	0.033	0.092	0.054	0.133	0.064	0.083	0.055
340	0.067	0.021	0.1	0.053	0.117	0.051	0.108	0.059
215	0.083	0.037	0.083	0.047	0.125	0.063	0.092	0.049
388	0.05	0.019	0.092	0.057	0.108	0.068	0.108	0.066

Table 7. Results of baselines on Exp.3 in digital domain based on ImageNet database (unseen model).

Methods	BIM		EOT		RP2		D2P	
Model	ASR	Conf	ASR	Conf	ASR	Conf	ASR	Conf
VGG16	0.067	0.033	0.108	0.074	0.108	0.044	0.075	0.048
VGG19	0.042	0.021	0.100	0.053	0.142	0.761	0.083	0.049
Res50	0.025	0.012	0.075	0.045	0.125	0.081	0.100	0.065

Table 8. Results of baselines on Exp.4 in digital domain based on ImageNet database (unseen class and unseen model).

Methods	BIM		EOT		RP2		D2P	
(Label, Model)	ASR	Conf	ASR	Conf	ASR	Conf	ASR	Conf
(288, VGG16)	0.067	0.023	0.108	0.045	0.092	0.049	0.083	0.041
(340, VGG19)	0.075	0.035	0.092	0.058	0.092	0.067	0.067	0.036
(215, Res50)	0.042	0.022	0.067	0.045	0.117	0.071	0.108	0.053

4.3. Efficiency Analysis

We present the efficiency of various methods in Table 9. Although our method needs pre-training and fine-tuning, during attacking evaluation, we only need to feed tested images into the fune-tuned model without gradient calculating, which costs further less time than baselines.

Table 9. Efficiency comparison results of different attack methods.

	EOT	D2P	RP2	Ours
Pre-training	✗	✓($\sim$ 1 day)	✗	✓($\sim$ 1 day)
Fine-tuning	✗	✗	✗	✓($\sim 45s$)
Iteratively gradient updating required for each image	✓	✓	✓	✗
Attacking Runtime	$27.53s$	$27.45s$	$293.51s$	$1.32s$

5. Adversarial Examples Generated by Different Methods

In Figure 3, we show several adversarial examples generated by different methods including BIM [4], EOT [1], D2P [3], RP2 [2] and Ours on ImageNet and GTSRD datasets. On ImageNet, the original label is "Brittany spaniel" while the DNN model classifies the adversarial examples as "badger". On GTSRD, the original label is "Speed limit 50 km/h" while the DNN model classifies the adversarial examples as "Speed limit 30 km/h". But as shown in Figure 3, when we look at the adversarial examples, we can still visually perceive them as "Brittany spaniel" and "Speed limit 50 km/h".

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018. 1, 3, 4

[2] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 2, 3, 4

[3] Steve TK Jan, Joseph Messou, Yen-Chen Lin, Jia-Bin Huang, and Gang Wang. Connecting the digital and physical world: Improving the robustness of adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 962–969, 2019. 3, 4

[4] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 3, 4

[5] Jinqi Luo, Tao Bai, Jun Zhao, and Bo Li. Generating adversarial yet inconspicuous patches with a single image. 2020. 2