Supplement to "Universal Adversarial Perturbation via Prior Driven Uncertainty Approximation"

Hong Liu[1], Rongrong Ji[12*], Jie Li[1], Baochang Zhang[3], Yue Gao[4], Yongjian Wu[5], Feiyue Huang[5]
[1]Department of Artificial Intelligence, School of Informatics, Xiamen University, [2]Peng Cheng Lab
[3]Beihang University, [4]Tsinghua University, [5]Tencent Youtu Lab

Abstract

In this supplementary material, Section 1 presents more additional quantitive results on ImageNet Validation set. We report results when all approaches have access both to the training data and the model parameters. Section 2 shows the experimental results on the psychical world, which can further extend our method to real-life application.

1. Additional Quantitative Results

In this section, we present qualitative results to show that the proposed approach PD-UA is competitive with respect to all those approaches which has access both to the training data and the model parameters. The proposed PD-UA is general, our model can directly access both the training data and model parameters.

For the pseudo data prior, we use the Gaussian distribution whose mean $\boldsymbol{\mu}$ is equal to the mean of the training data and variance $\boldsymbol{\delta}$ is such that 99.9% of the samples lie in $[0, 255]$, the dynamic range of input. Therefore, the objective function of the virtual Epistemic uncertainty in Eq.5 can be rewritten as follows:

$$p_e(\boldsymbol{\Delta}_{ij}) = \frac{1}{T}\sum_{t,x\sim\mathcal{G}}^{T}\left[-\log\left(\|f_j^{\boldsymbol{W}_i}(\boldsymbol{x}+\boldsymbol{\delta})\|_2 \cdot \boldsymbol{z}_j^t\right)\right], \quad \text{(S.1)}$$

where $\|\boldsymbol{\delta}\|_p < \epsilon$, $\mathcal{G}$ is the Gaussian distribution that the model can access. Beside, similar to [4], the Gaussian distribution can be easily replaced with the actual data distribution $\mathcal{X}$, which can further improve the attack performance. And the objective function of the virtual Epistemic uncertainty can be rewritten as follows:

$$p_e(\boldsymbol{\Delta}_{ij}) = \frac{1}{T}\sum_{t,x\sim\mathcal{X}}^{T}\left[-\log\left(\|f_j^{\boldsymbol{W}_i}(\boldsymbol{x}+\boldsymbol{\delta})\|_2 \cdot \boldsymbol{z}_j^t\right)\right], \quad \text{(S.2)}$$

where $\|\boldsymbol{\delta}\|_p < \epsilon$.

Therefore, these two new objective functions (Eq.S.1 and Eq.S.2) can be integrated into the final objective function in Eq.8, which can also be optimized via Algorithm 1. Table A shows the quantitative results on ImageNet validation set.

Table A. Fooling rate results for UAPs learned with training data.

Methods	VGG16	VGG19	ResNet50
classical UAP	77.80	80.80	81.39
GD-UAP with training data	72.80	67.60	56.40
singular-fool [1]	52.00	60.00	44.00
PD-UA with pseudo data	70.69	64.98	63.50
PD-UA with training data	76.60	73.30	65.80

We add a new compared method [1], termed singular-fool, that also generate the textural-like universal adversarial perturbation. Note that, singular-fool crafts UAP with small-scale training data ($N < 100$), and we test our method in the same setting for a fair comparison. We use 49 training images that are released in [1][1], to find the UAP for both GD-UAP and PD-UA. The results show that PD-UA with a pseudo data prior is better than the singular-fool. PD-UA can achieve a similar fooling rate when compared with the classical UAP, which however requires thousands of images to obtain such results. In addition, for VGG-based CNNs, PD-UA achieves the similar attacking performance comparing to the classical UAP [3], yet classical UAP need access to $10,000$ training samples.

Moreover, we compare our PD-UA with the JPEG compression-based attack method [5] that is a simple image-agnostic attacking method. We use a pre-trained ResNet50 and test the classification accuracy on the first 1,000 images of the ImageNet validation set. The JPEG compression (with 75% quality) achieves 66.3% Top-1 accuracy, while our PD-UA achieves 32.2%. Again, PD-UA is much better (here, a smaller accuracy means the model is more vulnerable to the adversarial attack).

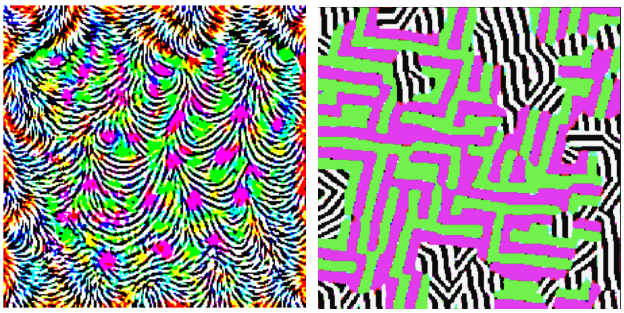
Finally, we also visualize the universal adversarial perturbations form different methods, as shown in Figure D. We observe that singular-fool, GD-UAP, and PD-UA present interesting visual patterns that are similar to the textural patterns. Besides, GD-UAP and PD-UA have a very

*R. Ji (rrji@xmu.edu.cn) is the corresponding author.

[1] https://github.com/KhrulkovV/singular-fool

Table B. The attack performance on VGG-16. "wpp" means that the perturbation are generated via objective function Eq.S.1 and "wap" means that the perturbation are generated via objective function Eq.S.2. "prob" means that the probability of the category output.

	Clean		wpp		wap	
	label	prob	label	prob	label	prob
example 1	long-borned beetle	0.943	cricket	0.481	shopping_basket	0.674
example 3	giant panda	0.999	lesser panda	0.267	monitor	0.700
example 4	pier	0.965	dome	0.246	mosquito net	0.3587
example 6	football helmet	0.771	mountain bike	0.755	backpack	0.306



(a) classical UAP (b) singular-fool

(c) GD-UAP (d) PD-UA

Figure A. The visualization of UAPs based on the backbone VGG-16, when the methods access to both training data and model parameters. (Best view in color.)

similar structure of the generated UAP, both of which contains a pattern of the building. However, the rules of such a pattern's arrangement in PD-UA is different from GD-UAP's, because PD-UA is constrained via the textural circle prior that significantly brings the performance improvement. As a conclusion, PD-UA does help to craft robust universal perturbation, which improves the white-box attacking performance.

2. Real-life Application of the PD-UA

To strength our attack method, this section shows the attack results on many real-life computer vision system, *i.e.,* mobile image recognition and Google Image. We randomly select six images from the ImageNet Validation set, whose examples are shown in Figure B.

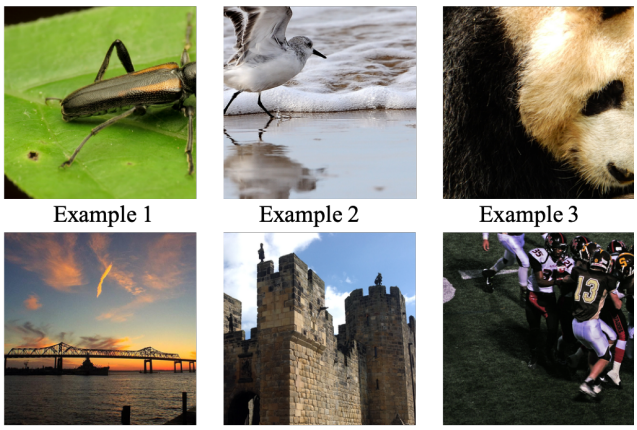
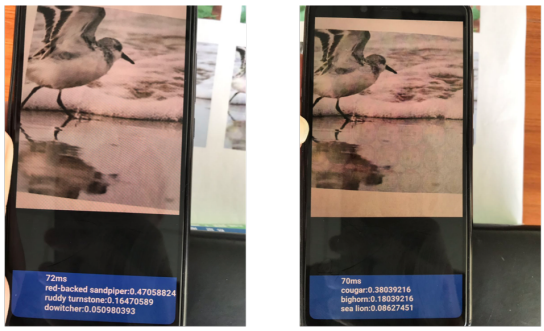


Figure B. The six images are selected for showing the attack results on a real-life application.

First, we computed the adversarial examples for the standard VGG-16 classifier using two different objects, *i.e.,* with pseudo data prior Eq.S.1 and with actual data distribution Eq.S.2. We directly input these images (both original clean images and adversarial images) into the VGG-16 classifier and out the Top-1 category, whose results are reported in Table B. We observe that PD-UA attack all the images successfully. Note that the adversarial example 4 that is crafted for "wpp", the Top-1 category is "dome" that has a similar textural style with the UAP that is shown in Figure 4 (e).

Second, we have tested our PD-UA to fool a publicly available TensorFlow camera demo [2]. Using the same setting as [2], we first print a clean image selected from ImageNet, and generate its adversarial image for the testing. Then, we use a Demo App to classify them, which is shown in Figure C. The clean image (a) is recognized correctly as a "sandpiper" when being perceived through the camera, while adversarial image (b) with PD-UA is misclassified. Moreover, we show more results based on our PD-UA that are generated via the two objective functions above. Different from Figure C, we show the clean image and adversarial

[2] https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android



(a) Original Image **(b) Adversarial Image**

Figure C. Demonstration of the attack on a phone app for image classification using physical adversarial examples. The perturbation is crafted via VGG-16. Enlarge to see details.

image on the computer screen and then use the Demo App to classify them, which is shown in Figure D.

Finally, we show the attack results on a real-world image retrieval system, i.e., Google Image. The first row shows the retrieval results along with original clean image, and the last two rows show the retrieval results along with two different UAPs. As the first column shown in Figure E, PD-UA can also realize the cross-task attack, *i.e.,* UAP that is crafted from classification model can also attack image retrieval system. Note that, the retrieval results of example 4 are the same. We compare the clean image with two adversarial images, we observe that the structure of the bridge is very clear, which causes the texture changes to not reduce the information of this image. Similar results emerged in Figure D (d), where the structure of the castle is very clear. The attack results have demonstrated that the proposed method can generate universal perturbations to fool the real-world search engine.

References

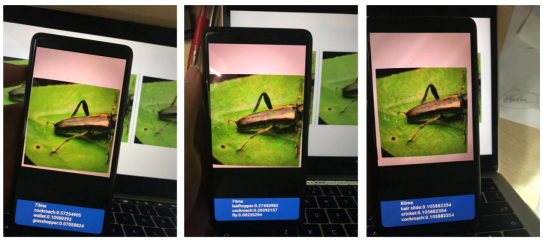
[1] Valentin Khrulkov and Ivan V Oseledets. Art of Singular Vectors and Universal Adversarial Perturbations. In *Proceedings of the CVPR*, 2018. 1

[2] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Examples in The Physical World. In *Proceedings of the ICLR*, 2017. 2

[3] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal Adversarial Perturbations. In *Proceedings of the CVPR*, 2017. 1

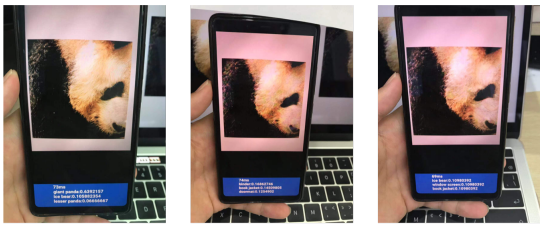
[4] Konda Reddy Mopuri, Aditya Ganeshan, and Venkatesh Babu Radhakrishnan. Generalizable Data-free Objective for Crafting Universal Adversarial Perturbations. *Journal of the IEEE TPAMI*, 2018. 1

[5] Richard Shin and Dawn Song. Jpeg-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, 2017. 1



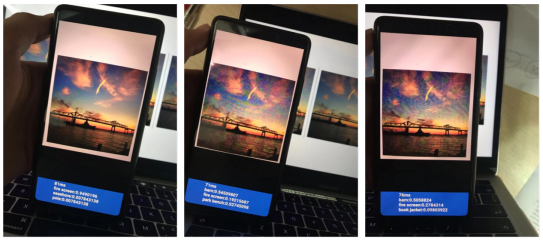
a) Original Image b) Adversarial Image 1 c) Adversarial Image 2

(a) Example 1.



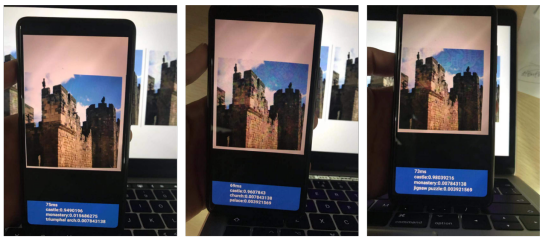
a) Original Image b) Adversarial Image 1 c) Adversarial Image 2

(b) Example 3.



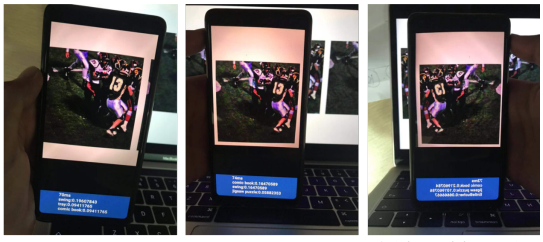
a) Original Image b) Adversarial Image 1 c) Adversarial Image 2

(c) Example 4.



a) Original Image b) Adversarial Image 1 c) Adversarial Image 2

(d) Example 3.



a) Original Image b) Adversarial Image 1 c) Adversarial Image 2

(e) Example 6.

Figure D. Demonstration of the attack on a phone app for image classification using physical adversarial examples. The perturbation is crafted via VGG-16. Enlarge to see details.

a) Original Image b) Adversarial Image 1 c) Adversarial Image 2

Figure E. Example retrieval results on Google Images. Enlarge to see details.