Supplementary Material to Understanding Adversarial Examples from the Mutual Influence of Images and Perturbations

Chaoning Zhang* chaoningzhang1990@gmail.com
Philipp Benz* pbenz@kaist.ac.kr
Tooba Imtiaz timtiaz@kaist.ac.kr
In-So Kweon iskweon@kaist.ac.kr
* indicates equal contribution
Robotics and Computer Vision (RCV) Laboratory
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea

Supplementary to the main manuscript we provide further experiments and results.

1. White Image and Gaussian Noise Background for Training

Table 1. Our data free approach with white background and Gaussian noise as proxy datasets for target class 'sea lion'. For each entry the fooling ratio and targeted fooling ratio are reported. The perturbations were crafted for VGG19. 'White' denotes white image as the background and 'Noise' denotes Gaussian noise as the background.

Method	AlexNet		GoogleNet		VGG16		VGG19		ResNet152	
White	57.1	0.7	27.9	0.2	45.6	0.5	37.0	0.1	22.0	0.1
Noise	76.1	0.0	24.8	0.0	31.9	0.0	33.5	0.0	23.2	0.0

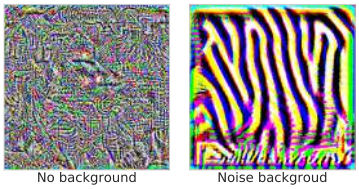

Figure 1. Targeted perturbations crafted for VGG19 with white background images (left) and Gaussian noise background images (right) for target class 'sea lion' with VGG19

As stated in the main paper, we explored generating targeted UAPs with white images, *i.e*. empty background, and Gaussian noise as the proxy dataset, and inferior performance is observed in both scenarios. The results are presented in Table 1. We observe that in the case of white background images the generated perturbation has a feature pattern visually similar to the perturbations crafted with other proxy datasets. To confirm the presence of sea lion features, we investigate the logit vector with the pure perturbation as the network input and find that indeed the index for the highest logit value is 'sea lion'. However, the low targeted fooling ratio indicates that the perturbation is not robust under the influence of images. It is reasonable that in this scenario the perturbation feature is not robust to the influence of images, since the perturbation is optimized with empty background images. On the other hand, the perturbation generated with Gaussian noise as background images, is not able to learn meaningful features of the target class ('sea lion'), which is verified by investigating the logit vector as above. We find that the index for the highest logit value is 'zebra'. The reason could be attributed to the 'stripe-like pattern' in the perturbation.

The above analysis validates the necessity of a proxy dataset for training a targeted UAP without original training data.

2. Results on CIFAR100

We propose to exploit proxy datasets to generate targeted UAPs without access to the original training data. In the main paper, we evaluate the proposed approach on ImageNet with the average performance reported. Here, we report the performance for each chosen random target class in Table 5.

We conduct similar experiments on CIFAR100 and report the results in Table 6. Similar to the observation on ImageNet, we find that on CIFAR100, generating targeted UAPs with proxy datasets leads to comparable performance to perturbations generated with the original training dataset.

3. Comparison with Existing Methods

Here we include further comparison with previous universal perturbation methods.

3.1. Comparison with Targeted Universal GAP

Table 2. Comparison with the performance of targeted UAP crafted with GAP [5] for InceptionV3. The numbers are reported in the targeted fooling ratio (%).

Method	T_1 (805)	T_2 (616)	T_3 (12)
GAP [5]	74.1	63.6	61.8
Ours (ImageNet)	77.9	67.8	73.2
Ours (COCO)	77.5	64.2	68.9
Ours (VOC)	76.4	64.3	61.4
Ours (Places365)	73.4	59.2	61.2

Since Poursaeed *et al*. report the targeted fooling ratio for three cases on ImageNet in their work [5], we craft targeted perturbations on ImageNet and different proxy datasets and compare their performance to ours in Table 2. Using the original ImageNet training dataset to craft the perturbation, our method is able to outperform GAP which uses the original training data as well. Even without the original training data, perturbations crafted on the COCO proxy dataset still outperform GAP with generative models. Note that our approach does not require training another generative model.

3.2. Comparison to GD-UAP with Proxy Dataset

GD-UAP [3] is an improved version of FFF [4]. GD-UAP seeks a perturbation causing additional activation to fire at each layer and thereby misleading the following layer [3]. The performance of GD-UAP and UAP with a proxy dataset were reported in [2]. In contrast to the decision boundary based UAP, whose performance decreases significantly with the proxy dataset, GD-UAP does not craft perturbations via optimizing a fooling objective and thus does not suffer a performance decrease [3]. This aligns well with our finding that decision boundary curvature property exploitation is not needed to generate effective universal perturbations. As shown in Table 3, with the proxy dataset, our approach outperforms UAP trained on a proxy dataset by a large margin, which clearly supports our main argument that images are like noise to the universal perturbation feature and thus there is no need to exploit the decision boundary curvature property. Note that our approach also outperforms GD-UAP trained on a proxy dataset for the metric of fooling ratio and GD-UAP in general can not be used for generating targeted UAPs, since they do not have an explicit fooling objective [3].

4. Dependence on Dataset Size

We investigate the influence of the dataset size on the performance of the crafted universal adversarial perturbations. The results in Table 4 show that the algorithm also performs well under limited access to data samples. Even with only 100 training samples, a targeted fooling ratio of 53.1% can be achieved utilizing the Places365 proxy dataset.

Table 3. Comparison with the data-free methods reported in [3]. All numbers are reported in the fooling ratio metric (%).

Method	AlexNet	GoogleNet
GD-UAP (proxy dataset) [3]	87.02	71.44
UAP (proxy dataset) [3]	73.1	28.2
Ours (ImageNet)	89.9	77.7
Ours (COCO)	89.9	76.8
Ours (VOC)	88.9	76.7
Ours (Places365)	90.0	76.4

Table 4. Dependence of dataset size. The universal adversarial perturbations were crafted with a limited number of training samples for a VGG16 network. As a proxy dataset we use ImageNet and Places365. The numbers in each column refer to the fooling ratio (%) and targeted fooling ratio (%).

Training Samples	Imagenet	Places365
10	87.08 / 66.01	41.86 / 0.72
100	92.28 / 78.72	78.17 / 53.1
1000	93.61 / 81.59	92.77 / 81.72
5000	93.22 / 80.82	92.37 / 82.01
10000	93.58 / 82.32	92.84 / 81.60
All	93.64 / 82.61	93.19 / 83.52

5. Parameter Ablation Studies

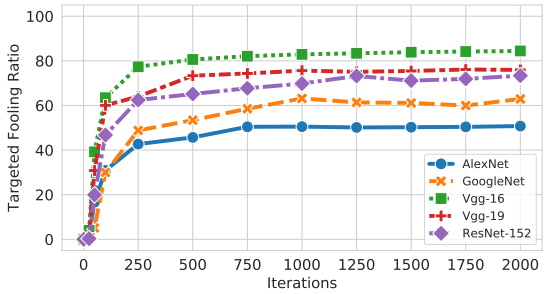

Figure 2. Targeted fooling ratio over the number of iterations I.

We perform ablation studies to find appropriate values for the hyper-parameters I and κ, and the results are available in Figure 2 and Figure 3, respectively. We observe that extra training after 1000 iterations provides only a marginal performance boost, thus we set I to 1000. We set κ to 10 because it provides overall better performance than other values.

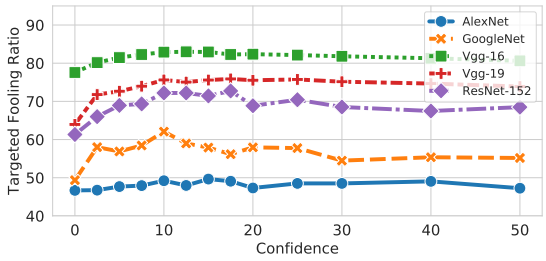

Figure 3. Targeted fooling ratio over the confidence value κ.

6. Further Logit Vector Analysis Results

We provide further results for the proposed logit vector analysis using different images and target classes on different target networks in Figures 5, 6, 7. The results in Figures 5, 6, 7 further support the claims in the main paper, that the image having a higher PCC value dominate the final classification outcome of the combined image, that targeted UAPs dominate the classification outcome and the image behaves like noise to the perturbation, and that image-dependant perturbations have a noise-like behavior but are able to fool the network in combination with the image.

7. Failure Case Analysis

The above analysis explains why UAPs are able to attack most images. However, a few images remain unfooled by the UAP. Here, we provide an analysis for the failure case, with the results shown in Figure 4. The ground-truth label for the image is 'pineapple', after adding a universal perturbation the prediction remains the same as the initial prediction. The PCC analysis reveals that the pineapple feature is more dominant than the universal perturbation feature. Not surprisingly, this sample also dominates another natural image with high confidence. The failure case analysis partially reveals the mystery why universal perturbations can not attack all images. The reason lies in the difference of the images. Specifically, the feature dominance of the corresponding image plays a crucial role in determining whether it can be easily attacked by a universal perturbation. To our best knowledge, this is the first attempt at analyzing the failure case for universal perturbations. Together with the analysis in the above subsection, our exploration suggests that the logit vector based PCC metric can be one effective tool for analyzing the DNN response to the features in the input. We believe that this simple tool has potential to be used in general scenarios for understanding DNN feature.

7.1. Perturbation Visualization

More qualitative results of the generated perturbations for different target classes on different networks are available in Figure 8. We observe relevant target class patterns in most of the perturbations. Thus, our method can also be helpful for visualizing what the network has learned.

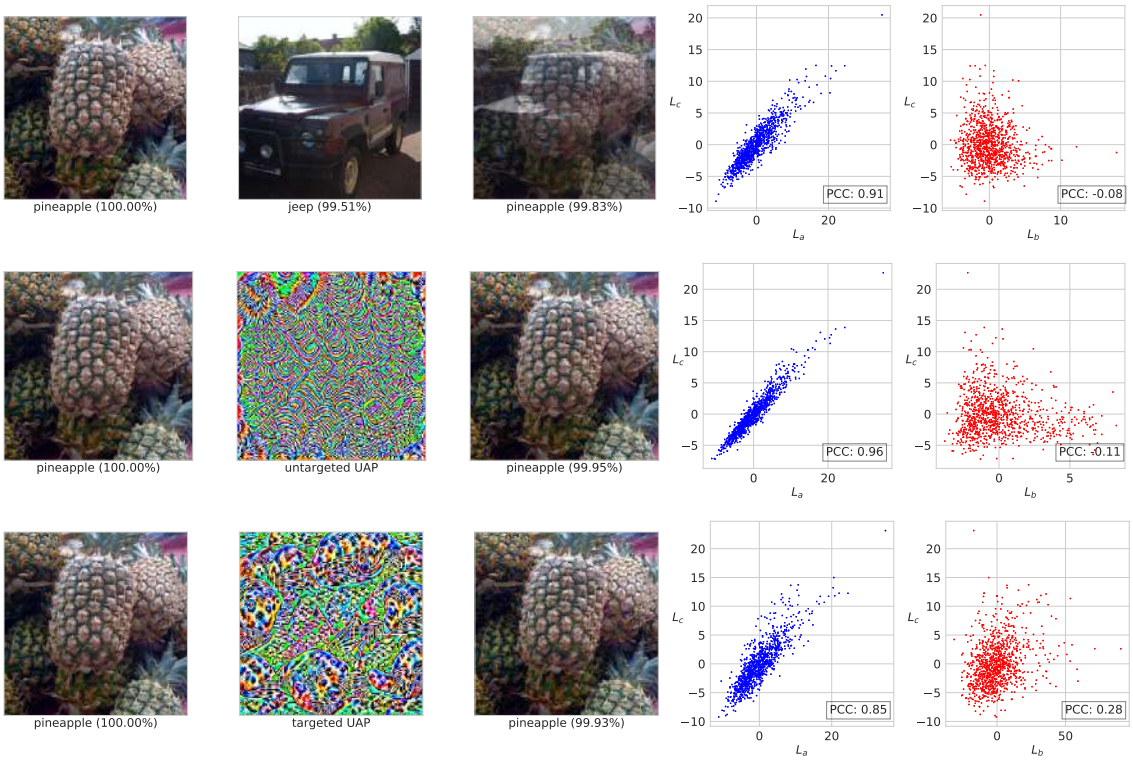

Figure 4. **Failure case analysis result for one sample image 'pineapple', target network VGG19, and target class 'sea lion'** (150)**.** The columns show input a, input b, input $c = a + b$, logit vector analysis of L_c over L_a and that of L_c over L_b. The image is shown in combination with another image (top), a non-targeted UAP (middle) and a targeted UAP (bottom). Below each image, the classification label and the softmax output are stated. The perturbations shown are amplified for visualization.

Table 5. Detailed results for 8 different target classes for networks trained on ImageNet. The values in each column represent the non-targeted fooling ratio (%) and targeted fooling ratio (%)

	Data	T_1 (150)		T_2 (426)		T_3 (843)		T_4 (715)		T_5 (952)		T_6 (507)		T_7 (590)		T_8 (62)		Avg.	
AlexNet	ImageNet	90.0	50.2	88.8	45.2	88.7	60.4	90.7	54.8	94.7	72.3	88.3	31.1	86.9	30.1	91.3	45.1	89.9 ± 2.2	48.6 ± 13.3
	COCO	90.5	49.4	89.4	43.8	88.1	54.8	91.5	50.0	94.5	71.6	87.2	33.4	86.6	29.0	91.6	45.6	89.9 ± 2.6	47.2 ± 13.1
	VOC	88.7	48.9	88.7	43.9	88.7	56.9	90.0	52.4	93.9	69.3	84.5	36.6	86.3	24.3	90.5	42.5	88.9 ± 2.6	46.9 ± 12.7
	Places365	90.0	46.2	89.0	39.3	89.0	55.8	91.0	54.5	94.4	70.2	89.3	16.3	86.5	29.6	90.8	29.0	90.0 ± 2.1	42.6 ± 16.4
GoogleNet	ImageNet	76.0	61.6	80.1	46.1	75.3	64.8	75.4	58.5	79.2	67.2	77.6	66.9	73.5	54.3	84.3	59.6	77.7 ± 3.2	59.9 ± 6.6
	COCO	73.0	58.4	81.1	45.8	75.7	66.3	74.0	57.9	77.3	65.0	77.8	68.5	72.7	53.5	83.0	63.2	76.8 ± 3.7	59.8 ± 7.5
	VOC	72.6	56.2	79.9	48.3	75.4	65.5	76.6	61.9	78.4	66.6	74.8	62.5	73.3	52.1	82.5	58.0	76.7 ± 3.2	58.9 ± 6.0
	Places365	73.9	61.5	78.2	51.0	75.4	65.6	74.2	58.9	77.9	65.9	75.8	64.5	71.5	52.1	84.4	60.8	76.4 ± 3.7	60.0 ± 5.4
VGG16	ImageNet	93.6	82.3	92.5	65.2	91.4	83.2	92.5	80.0	94.9	83.1	91.2	63.6	90.6	68.0	93.3	75.0	92.5 ± 1.3	75.0 ± 7.8
	COCO	93.5	82.8	92.4	64.9	90.7	83.3	92.4	79.8	95.0	84.0	91.1	54.3	89.6	69.8	93.1	72.4	92.2 ± 1.7	75.1 ± 12.3
	VOC	93.6	81.3	92.2	65.4	91.1	83.4	92.9	80.9	94.6	83.3	91.2	63.1	89.2	69.8	92.5	70.4	92.2 ± 1.6	74.7 ± 7.9
	Places365	93.3	81.9	92.0	65.5	90.8	83.0	92.6	80.2	94.7	83.6	91.1	56.0	89.5	66.5	93.0	70.4	92.1 ± 1.5	73.4 ± 9.6
VGG19	ImageNet	92.8	74.8	90.7	63.5	90.9	82.4	90.8	77.4	93.4	74.4	89.9	63.0	90.8	63.3	93.2	74.2	91.6 ± 1.3	71.6 ± 6.9
	COCO	92.7	72.3	91.4	68.6	91.2	82.6	90.5	76.4	94.1	51.6	89.4	66.6	90.3	61.4	92.9	70.9	91.6 ± 1.5	68.8 ± 9.4
	VOC	92.5	74.1	90.1	64.9	90.3	81.4	85.5	59.9	92.9	76.5	89.9	66.2	89.7	55.3	93.2	72.1	90.5 ± 2.3	68.8 ± 8.2
	Places365	92.8	74.9	89.9	62.6	91.0	82.1	90.8	77.3	94.3	24.3	89.1	56.3	90.8	68.3	93.0	70.3	91.5 ± 1.6	64.5 ± 17.0
ResNet152	ImageNet	82.9	73.6	81.8	66.6	79.4	71.2	77.8	63.2	83.0	72.8	80.6	54.1	76.4	56.8	84.2	71.7	80.8 ± 2.6	66.3 ± 7.0
	COCO	81.3	70.6	82.9	67.7	80.8	73.3	77.2	63.5	80.6	70.4	79.2	56.4	74.4	51.8	83.1	71.9	79.9 ± 2.9	65.7 ± 7.8
	VOC	80.4	71.0	81.5	65.6	80.4	72.6	73.6	57.2	80.6	69.8	78.4	54.2	74.3	58.5	83.8	72.8	79.1 ± 3.3	65.2 ± 7.1
	Places365	80.1	70.0	80.4	66.0	80.1	72.1	72.2	58.1	79.6	68.6	77.2	40.2	73.6	56.5	81.2	68.5	78.0 ± 3.2	62.5 ± 9.9

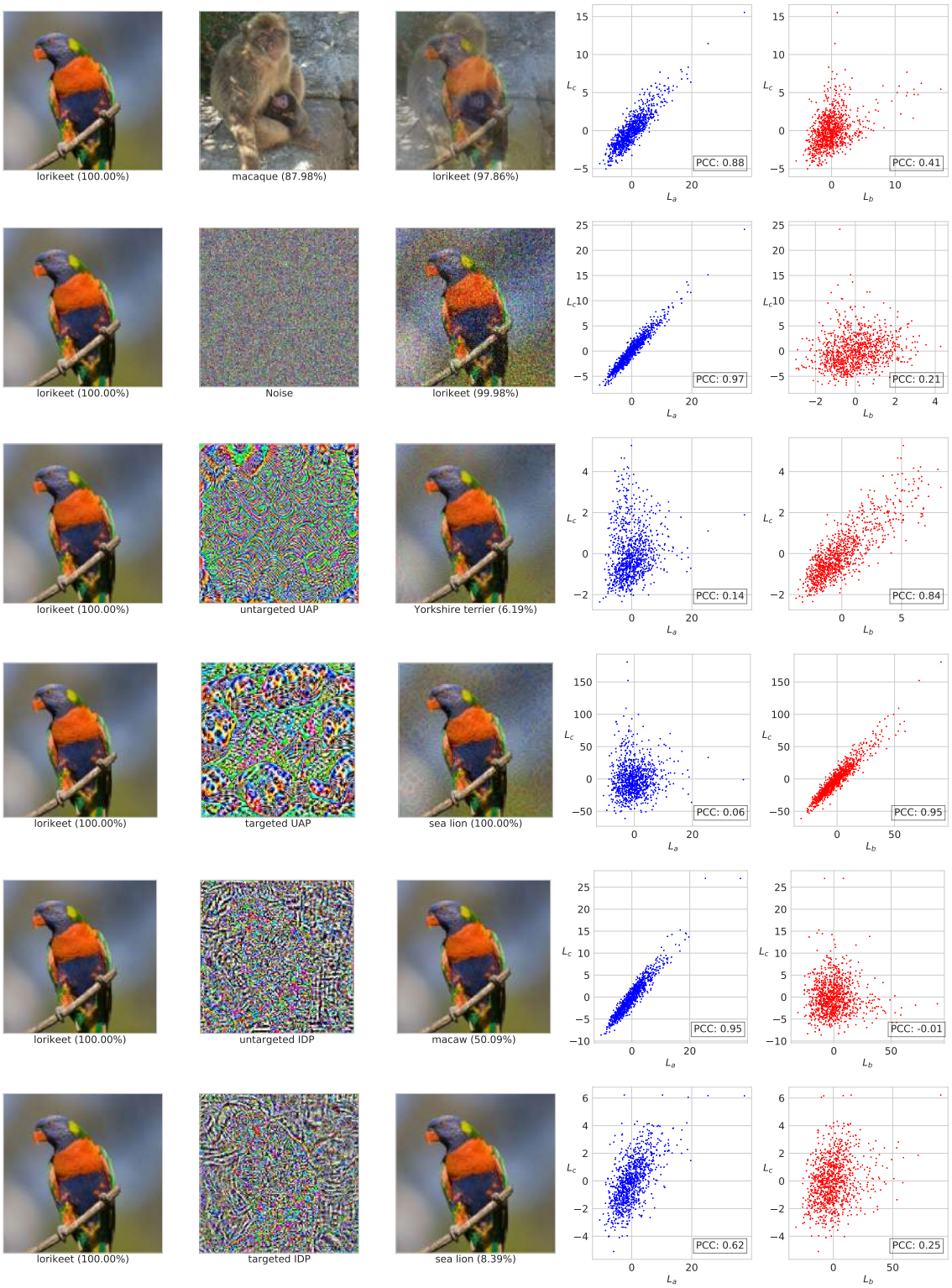

Figure 5. **Analysis result for one sample image 'lorikeet', target network VGG19, and target class 'sea lion'** (150)**.** Multiple scenarios of input combinations and their respective logit vector analysis are considered. The columns show input a, input b, input $c = a + b$, logit vector analysis of L_c over L_a and that of L_c over L_b. Below each image, the classification label and the softmax output are stated. PGD [1] is used to generate the image-dependant perturbations (IDP). All perturbations shown are amplified for visualization.

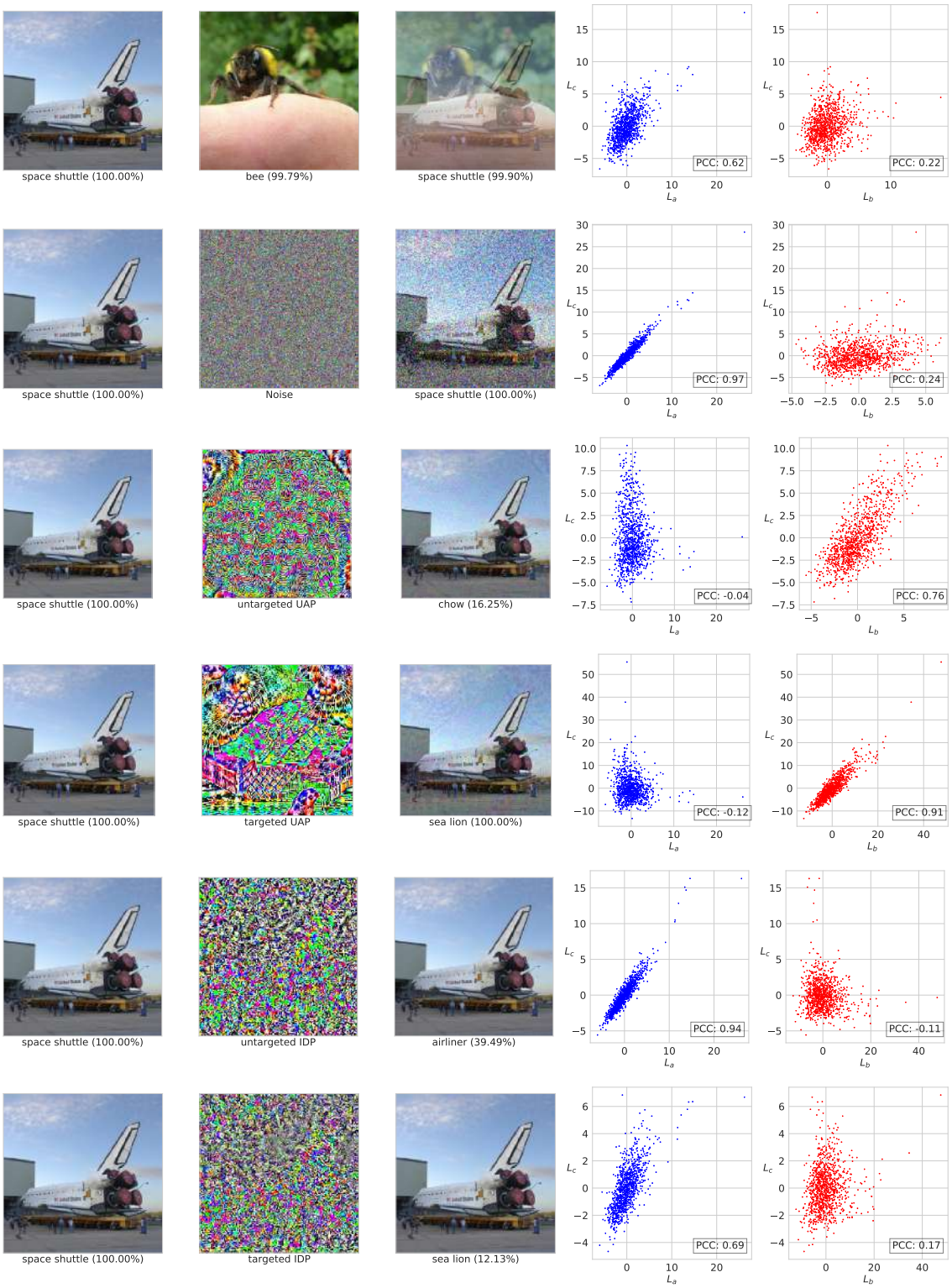

Figure 6. **Analysis result for one sample image 'space shuttle', target network ResNet152, and target class 'sea lion'** (150)**.** Multiple scenarios of input combinations and their respective logit vector analysis are considered. The columns show input a, input b, input $c = a+b$, logit vector analysis of L_c over L_a and that of L_c over L_b. Below each image, the classification label and the softmax output are stated. PGD [1] is used to generate the image-dependant perturbations (IDP). All perturbations shown are amplified for visualization.

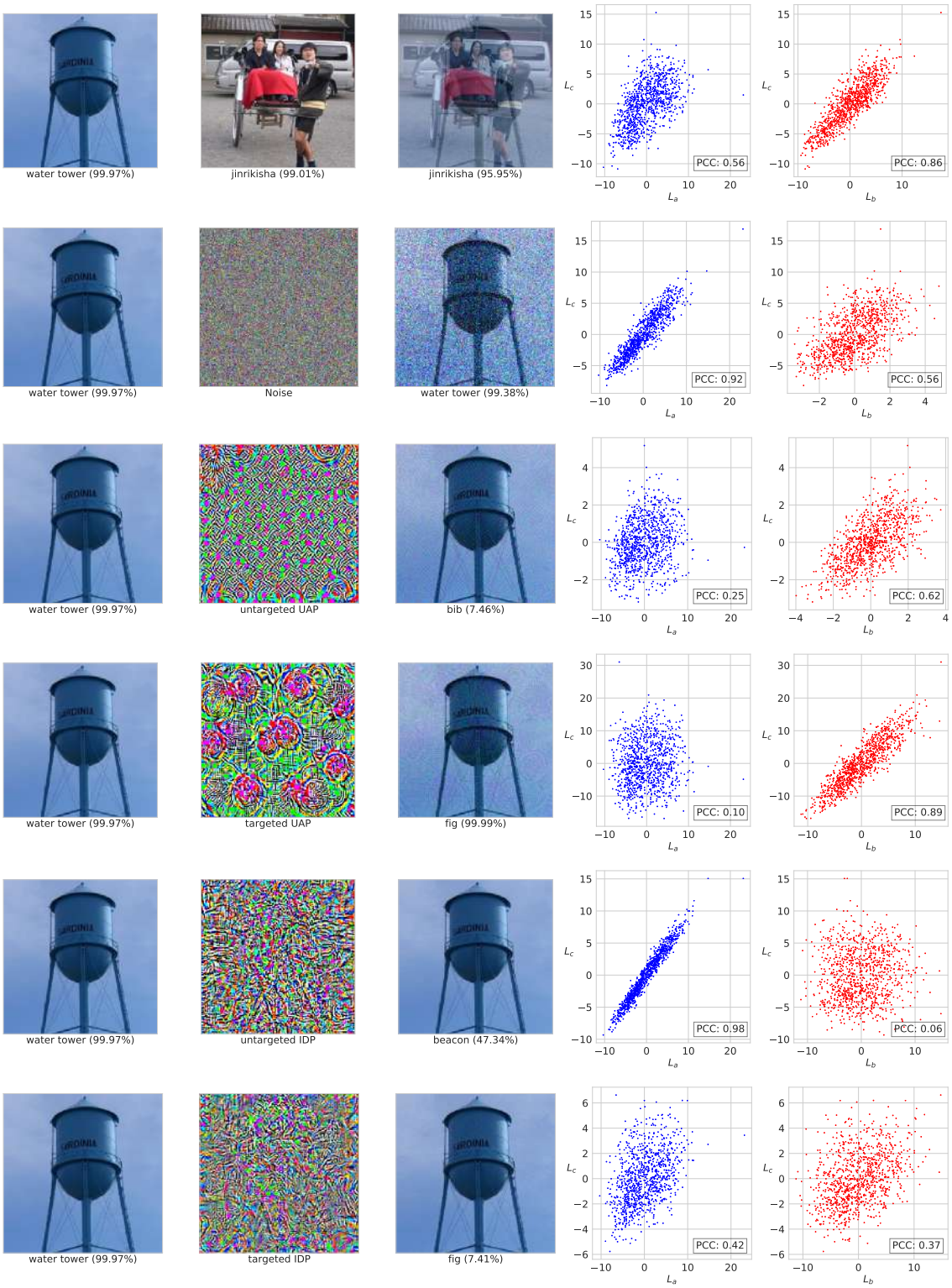

Figure 7. **Analysis result for one sample image 'water tower', target network AlexNet, and target class 'fig'** (952)**.** Multiple scenarios of input combinations and their respective logit vector analysis are considered. The columns show input a, input b, input $c = a + b$, logit vector analysis of L_c over L_a and that of L_c over L_b. Below each image, the classification label and the softmax output are stated. PGD [1] is used to generate the image-dependant perturbations (IDP). All perturbations shown are amplified for visualization.

Table 6. Detailed results for 8 different target classes for networks trained on CIFAR100. The values in each column represent the non-targeted fooling ratio (%) and targeted fooling ratio (%).

	Data	T_1 (6)		T_2 (20)		T_3 (29)		T_4 (39)		T_5 (42)		T_6 (51)		T_7 (86)		T_8 (93)		Avg.	
VGG16	CIFAR100	86.6	67.6	85.2	62.3	83.8	69.1	86.0	64.4	90.5	75.7	85.5	67.4	79.9	53.3	84.9	70.0	85.3 ± 2.8	66.2 ± 6.1
	COCO	86.5	68.8	85.2	61.7	83.9	68.0	79.9	49.0	90.6	75.4	82.3	66.7	79.2	50.5	84.5	69.8	84.0 ± 3.5	63.7 ± 8.8
	VOC	86.6	68.3	82.8	59.2	82.5	66.8	79.1	45.1	89.4	73.1	83.6	66.7	83.9	60.0	83.7	68.5	84.0 ± 2.8	63.4 ± 8.1
	Places365	87.0	68.9	83.6	57.6	82.5	66.5	80.2	47.2	90.2	74.7	81.3	63.5	83.8	58.1	84.5	68.6	84.1 ± 3.0	63.2 ± 8.1
VGG19	CIFAR100	86.0	71.1	83.5	56.1	85.6	73.7	82.2	43.9	92.1	77.8	88.8	70.7	83.2	58.5	85.1	66.6	85.8 ± 3.0	64.8 ± 10.5
	COCO	85.4	69.6	82.5	58.8	85.0	72.1	82.1	49.3	92.5	79.3	84.8	69.3	78.8	56.5	85.6	70.2	84.6 ± 3.7	65.6 ± 9.2
	VOC	86.4	72.5	83.0	58.7	85.8	74.4	79.1	46.8	91.3	78.3	85.7	69.5	80.7	58.4	84.2	66.5	84.5 ± 3.5	65.6 ± 9.7
	Places365	86.7	68.8	82.8	57.8	85.5	71.8	80.2	48.8	92.7	79.1	85.7	71.1	80.9	57.9	84.7	65.9	84.9 ± 3.7	65.1 ± 9.1
ResNet20	CIFAR100	90.0	71.2	87.4	63.7	86.4	68.9	92.3	70.9	91.3	76.1	90.5	77.9	89.3	64.2	88.3	69.7	89.4 ± 1.9	70.3 ± 4.7
	COCO	91.0	72.8	88.2	65.5	88.8	72.9	92.7	74.7	91.6	74.7	90.0	76.4	94.2	75.8	89.6	72.8	90.8 ± 1.9	73.2 ± 3.2
	VOC	89.7	69.7	87.7	63.0	85.6	66.3	91.6	71.6	90.6	72.6	89.9	76.1	94.1	76.9	88.8	71.6	89.7 ± 2.4	71.0 ± 4.4
	Places365	90.7	70.8	89.1	68.6	88.2	67.8	92.6	74.9	92.3	75.5	89.1	76.2	93.5	74.9	90.0	73.9	90.7 ± 1.8	72.8 ± 3.1
ResNet56	CIFAR100	91.2	78.3	86.6	65.7	88.1	72.4	90.6	76.4	92.0	73.6	93.0	80.1	90.0	66.7	89.8	77.8	90.2 ± 1.9	73.9 ± 5.0
	COCO	91.2	78.6	87.8	65.2	90.4	76.1	87.9	61.6	93.4	76.6	92.6	81.8	92.9	72.0	91.2	81.1	90.9 ± 2.0	74.2 ± 6.9
	VOC	91.0	78.7	87.3	66.7	89.7	75.5	89.3	69.3	92.4	75.8	91.7	80.6	92.2	72.9	89.9	75.4	90.5 ± 1.6	74.4 ± 4.3
	Places365	91.2	74.9	88.9	69.0	91.5	78.3	91.9	75.4	93.0	75.8	92.5	81.5	92.4	72.1	91.0	78.7	91.6 ± 1.2	75.7 ± 3.7

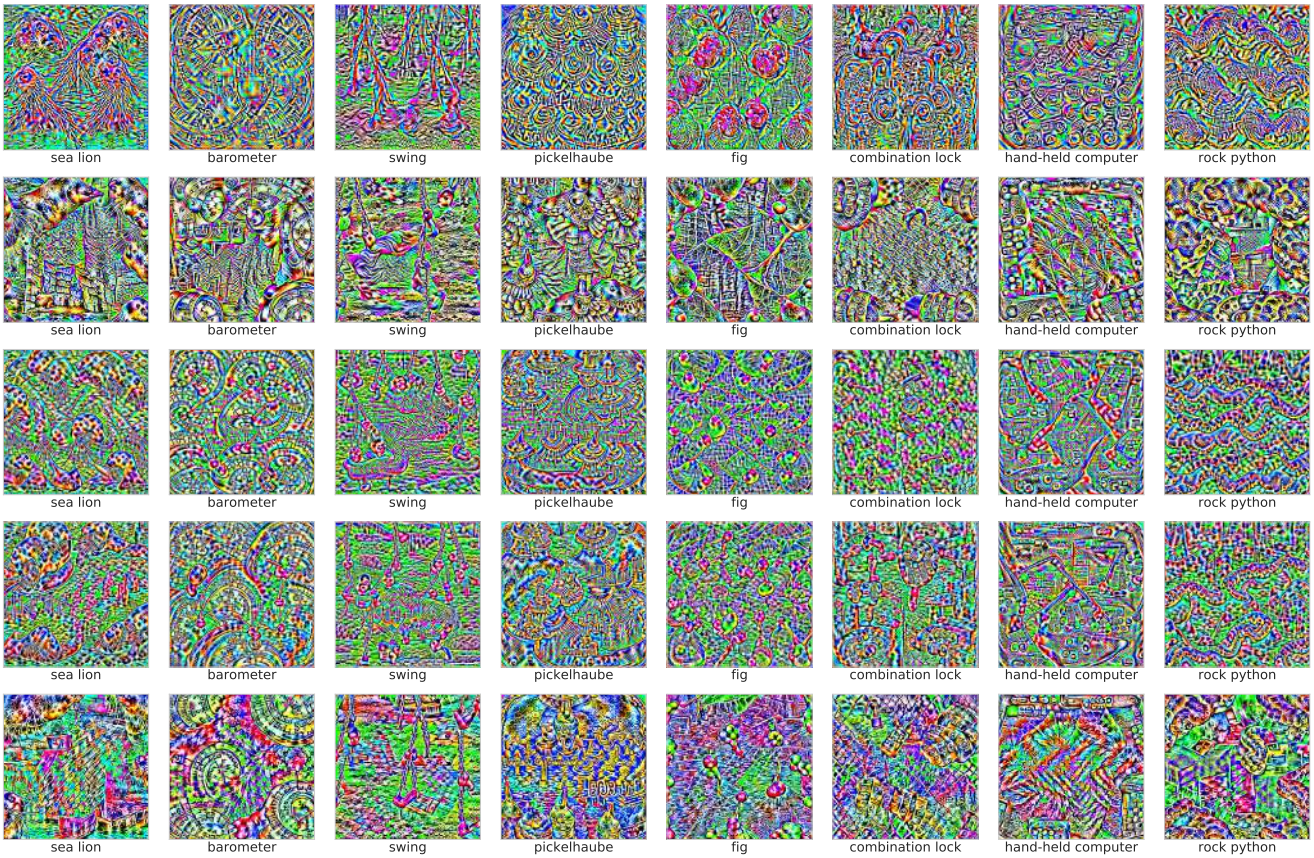

Figure 8. Targeted perturbations crafted for ImageNet pretrained models for different target classes. Each column indicates a different target class and each row a different network architecture. From the top to the bottom row, the order of the networks is: AlexNet, Googlenet, VGG16, VGG19, ResNet152

References

[1] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 5, 6, 7

[2] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar

Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[3] Konda Reddy Mopuri, Aditya Ganeshan, and Venkatesh Babu Radhakrishnan. Generalizable data-free objective for crafting universal adversarial perturbations. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 2

[4] Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. In *British Conference on Machine Vision (BMVC)*, 2017. 2

[5] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2