Universal Adversarial Perturbations are Not Bugs, They are Features

Philipp Benz* Chaoning Zhang* Tooba Imtiaz In-So Kweon

Korea Advanced Institute of Science and Technology (KAIST)

Abstract

A wide variety of works have explored the reason for the existence of adversarial examples, but there is no consensus on the explanation. We propose to treat the DNN logits as a vector for feature representation and exploit them to analyze the mutual influence of two independent inputs based on the Pearson Correlation Coefficient (PCC). We utilize this vector representation to understand adversarial examples by disentangling the clean images and adversarial perturbations and analyze their influence on each other. Our results suggest a new perspective towards the relationship between images and universal perturbations: Universal perturbations contain dominant features, and images behave like noise to them. This feature perspective leads to a new method for generating targeted UAPs using random source images. We achieve the challenging task of a targeted universal attack without utilizing original training data. Our approach using a proxy dataset achieves comparable performance to the state-of-the-art baselines which utilize the original training dataset.

1. Introduction

Deep neural networks (DNNs) have shown impressive performance in numerous applications, ranging from image classification [7, 28] to motion regression [4, 27]. However, DNNs are also known to be vulnerable to adversarial attacks [22, 20]. Contrary to previous works analyzing adversarial examples [6, 23, 24, 10, 2] as a whole (summation of image and perturbation), we propose to analyze adversarial examples by disentangling image and perturbations and studying their mutual influence. Specifically, we analyze the influence of two independent inputs on each other in terms of contributing to the obtained feature representation when the inputs are combined. We treat the network logit outputs as a means of feature representation. Traditionally, only the most important logit values, such as the highest logit value for classification tasks, are considered while other values are disregarded. We propose that all logit values contribute to the feature representation and therefore treat them as a logit vector. We utilize the Pearson Correlation Coefficient (PCC) [1] to analyze the extent of linear correlation between logit vectors. The PCC values computed between the logit vectors of each independent input and the input combination gives insight on the contribution of the two independent inputs towards the combined feature representation. Our findings show that for a universal attack [14, 15, 9, 25], the adversarial examples (AEs) are strongly correlated to the UAP, while a low correlation is observed between AEs and input images (see Figure 3). This suggests that for a DNN, UAPs dominate over the clean images in AEs, even though the images are visually more dominant. Treating the DNN as a feature extractor, we naturally conclude that the UAP has features that are more dominant compared to the features of the images to attack. Given this insight we extend the observations given by Ilyas *et al*. [8] and claim that "UAPs are features while images behave like noise to them". This is contrary to the general perception that treats the perturbation as noise to images in adversarial examples [6, 14].

The observation that images behave like noise to UAPs motivates the use of proxy images to generate targeted UAPs without original training data. This results in a more practical approach since the training data is generally inaccessible to an attacker [18]. A detailed version of this work is presented in [26] and our contributions can be summarized as follows:

- We propose to treat the DNN logits as a vector for feature representation. These logit vectors can be used to analyze the contribution of features of two independent inputs when summed towards the output. In particular, our analysis results regarding universal attacks reveal that in an AE, the UAP has dominant features, while the image behaves like noise to them.
- We leverage this insight to derive a method using random source images as a proxy dataset to generate targeted UAPs without original training data. To our best knowledge, we are the first to fulfill this challenging task and it achieves comparable performance to the

*indicates equal contribution. Correspondence to pbenz@kaist.ac.kr and chaoningzhang1990@gmail.com

state-of-the-art baselines utilizing the original training dataset.

2. Analysis Framework

Following the common consensus that DNNs are feature extractors, we intend to analyze adversarial examples from such perspective. We assume that all DNN output logit values represent the network response to features in the input. We adopt the logit vector (DNN output before the final softmax layer) to facilitate the analysis of the mutual influence of two independent inputs in terms of their contribution to the combined feature representation. We mainly consider two independent inputs $a \in \mathbb{R}^d$ and $b \in \mathbb{R}^d$, which can be images, Gaussian noise, perturbations, etc., whose corresponding logit vectors are denoted as L_a and L_b, respectively. The summation of these two inputs $c = a + b$, when fed to a DNN, leads to the feature representation L_c. Both inputs a and b contribute partially to L_c. Moreover, it is reasonable to expect that the contribution of each input will be influenced by the other one. Specifically, the extent of influence will be reflected in the linear correlation between the individual logit vector L_a or L_b and L_c. To calculate such correlations, we use the Pearson Correlation Coefficient (PCC).

In statistics, the PCC [1] is a widely adopted metric to measure the linear correlation between two variables. In general, this coefficient is defined as $\text{PCC}_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$. where cov indicates the covariance and σ_X and σ_Y are the standard deviations of vector X and Y, respectively, while the range of PCC value is between -1 and 1. The absolute value indicates the extent to which the two variables are linearly correlated, with 1 indicating perfect linear correlation, 0 indicating zero linear correlation, and the sign indicates whether they are positively or negatively correlated. Treating the logit vector as a variable and the logit values as the observations, the PCC between different logit vectors can be calculated. Comparing PCC_{L_a,L_c} and PCC_{L_b,L_c} can provide insight about the contribution of the two inputs to L_c, with a relatively higher PCC value indicating the more significant contributor.

As a basic example, we show the logit vector analysis of two randomly sampled images from ImageNet [11] in Figure 1. The plot shows a strong linear correlation between L_b and L_c ($\text{PCC}_{L_b,L_c} = 0.88$), while L_a and L_c are practically uncorrelated ($\text{PCC}_{L_a,L_c} = 0.19$). These observations suggest a dominant contribution of input b towards logit vector L_c. As a result, the same label "Wood rabbit" is predicted for c and b.

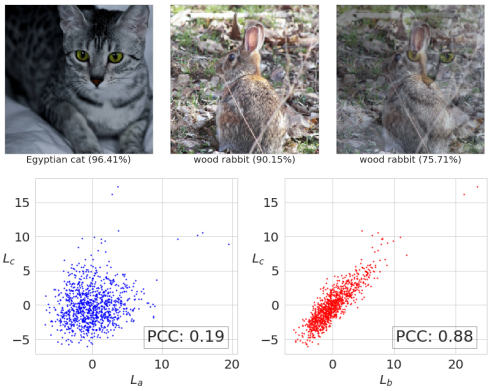

Figure 1. Images and their logit vector analysis. The first row shows the sample images a and b and the resulting image c. The second row shows the plots of logit vector L_c over L_a (left) and L_b (right), with their respective PCC values.

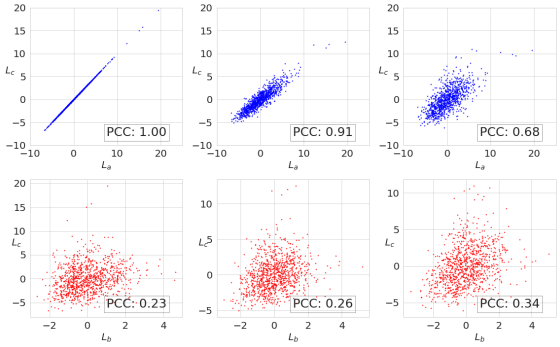

Figure 2. Logit vector analysis for an input image and Gaussian noise $\mathcal{N}(\mu, \sigma)$. The analysis is shown for $\mu = 0$ and $\sigma = 0$ (left), $\sigma = 0.1$ (middle) and $\sigma = 0.2$ (right))

3. Influence of images and perturbations on each other

In this section, we analyze the interaction of clean images with Gaussian noise perturbations, universal perturbations, and image-dependent perturbations. In doing so, input a is the image and input b the perturbation. The analysis is performed on VGG19 pretrained on ImageNet. For consistency, a randomly chosen a (shown in Figure 1, top left) is used for all experiments. Along the same lines, for targeted perturbations, we set 'sea lion' as the target class t.

Analysis of Gaussian Noise. To facilitate the interpretation of our main experiment of performing analysis for perturbations, we first show the influence of noise (Gaussian noise) on images. This Gaussian noise is sampled from $\mathcal{N}(\mu, \sigma)$ with $\mu = 0$ and different standard deviations. The relationship between L_a, L_c is visualized in Figure 2. As expected, by adding zero magnitude Gaussian noise (*i.e.* no Gaussian noise) to the image, L_a and L_c are perfectly linearly correlated ($\text{PCC}_{L_a,L_c} = 1$). If the Gaussian noise

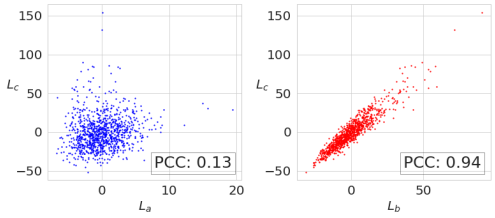

Figure 3. Logit vector analysis for input image (a) and targeted UAP (b).

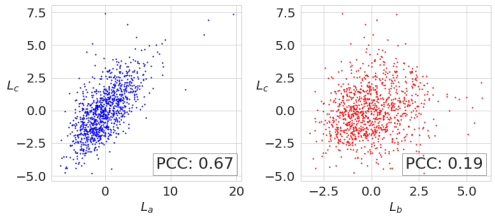

Figure 4. Logit vector analysis for input image (a) and targeted image-dependent perturbation (b). The perturbation was crafted with PGD [13], with target class 'sea lion'

magnitude is increased ($\sigma = 0.1$ for instance), L_a and L_c still show a high linear correlation ($\text{PCC}_{L_a,L_c} = 0.91$). Investigating the relationship between L_b and L_c, a low correlation can be observed for all noise inputs b indicating a low contribution to the final prediction.

Analysis of universal perturbations. The analysis results for the targeted UAP are shown in Figure 3. For the targeted scenario, two major observations can be made: First, PCC_{L_a,L_c} is smaller than PCC_{L_b,L_c}, indicating a higher linear correlation between L_c and L_b than L_c and L_a. In other words, the features of the perturbation are more dominant than that of the clean image. Second, PCC_{L_a,L_c} is close to 0, indicating that the influence of the perturbation on the image is so significant that the clean image features are seemingly unrecognizable to the DNN. In fact, comparing the logit analysis of L_a and L_c in Figure 3 with that of Gaussian noise and image in Figure 2 (bottom), a striking similarity is observed. This offers a novel interpretation of targeted universal perturbations: *Targeted universal perturbations themselves (independent of the images to attack) behave like features, while images behave like noise to them.* For the untargeted UAP we observed a similar behavior as for the targeted universal perturbations (PCC_{L_a,L_c} smaller than PCC_{L_b,L_c}). However the dominance of the non-targeted perturbation is not as significant as that of targeted perturbation.

Analysis of image-dependent perturbations The logit vector analysis results for a targeted image-dependent perturbation is reported in Figure 4. Contrary to the universal perturbations, image-dependent perturbations are weakly correlated to c and have a noise-like behavior (Figure 2). However, the image gets misclassified even though the image features appear to be more dominant than the perturbation. This is because the image features are more strongly corrupted through the image-dependent perturbation than Gaussian noise. The specific reason for this special behavior is that the image-dependent perturbations are crafted to form features only in combination with the image. Such image-dependent behavior violates our assumption of independent inputs. However, we include these results because they offer additional insight into adversarial examples.

3.1. Why adversarial perturbations exist?

Based on our previous analysis, we arrive at the following explanation for the existence of UAPs: *Universal adversarial perturbations behave like features independent of the images to attack. The image features are corrupted to an extent of being unrecognizable to a DNN, and thus the input images behave like noise to the perturbation features.*

The finding in [9] that universal perturbations behave like features of a certain class aligns well with our statement. Jetley *et al.* argue that universal perturbations exploit the high-curvature image-space directions to behave like features, while our finding suggests that universal perturbations themselves are features independent of the images to attack. Utilizing the perspective of positive curvatures of decision boundaries, Jetley *et al.* adopt the decision boundary-based attack DeepFool [16]. However, our explanation does not explicitly rely on the decision boundary properties but focuses on the occurrences of strong features, robust to the influence of images. We can, therefore, deploy the PGD-algorithm to generate perturbations consisting of target class features similar to [8].

If universal perturbations themselves are features independent of the images to attack, do image-dependent perturbations behave in a similar way? As previously discussed, the analysis results in Figure 4 reveal that image-dependent perturbations themselves are not like features but noise. On the other hand, the original image feature is retained to a high extent. Ilyas *et al.* [8] revealed that image-dependent adversarial examples include the features of the target class. However, as we saw from the previous analysis the isolated perturbation seems not to retain independent features due to its low PCC value, but rather interacts with the image to form the adversarial features.

4. Targeted UAP with Proxy Data

Our above analysis demonstrates that images behave like noise to the universal perturbation feature. Since the images are treated as noise, we can exploit proxy images to generate targeted UAPs without original training data. The proxy image does not need to have any class object belonging to the original training class and the main role of proxy images

Table 1. Results for targeted UAPs trained on four different datasets reported in the targeted fooling ratio (%) obtained over 8 different target classes.

Proxy Data	AlexNet	GoogleNet	VGG16	VGG19	ResNet152
ImageNet [11]	48.6	59.9	75.0	71.6	66.3
COCO [12]	47.2	59.8	75.1	68.8	65.7
VOC [5]	46.9	58.9	74.7	68.8	65.2
Places365 [29]	42.6	60.0	73.4	64.5	62.5

is to make the targeted UAP have strong background-robust target class features.

To achieve the desired objective of a targeted UAP to fool most of the data samples to a certain target class, most naively the cross-entropy loss function $\mathcal{L}_{\text{CE}}$ can be utilized. Since cross-entropy loss holistically incorporates logits of all classes, this loss function leads to overall lower fooling ratios. This behavior can be resolved by using a loss function that only aims to increase the logit of the target class. Since we consider universal perturbations, to balance the objective between different samples in training, we clamp the logit values as follows:

$$\mathcal{L}^t_{CL1} = \max(\max_{i \neq t} \hat{C}_i(x_v + v) - \hat{C}_t(x_v + v), -\kappa) \quad (1)$$

where κ indicates the confidence value and x_v are samples from the proxy dataset. In this case, the proxy data can be either a random source dataset or the original training data, depending on data availability. Note that similar techniques of clamping the logits have also been used in [3], however, their motivation is to obtain minimum-magnitude (image-dependent) perturbation. However, using this loss function $\mathcal{L}^t_{CL1}$, while the target logit is increased, the logit values of $\max \hat{C}_i(x_v + v)$ are decreased simultaneously during the training process. This effect is undesirable for generating a UAP with strong target class features since other classes except the target classes will be included in the optimization, which might have negative effects on the gradient update. To prevent manipulation of logits other than the target class, we exclude non-target class logit values in the optimization step, such that non-target class logit values are only used as a reference value for clamping the target class logit. We indicate this loss function as $\mathcal{L}^t_{CL2}$. We further provide a loss function resembling $\mathcal{L}^t_{CL2}$ for the generation of non-targeted UAPs.

$$\mathcal{L}^{nt} = \max(\hat{C}_{gt}(x_v + v) - \max_{i \neq gt} \hat{C}_i(x_v + v), -\kappa) \quad (2)$$

In the special case of crafting non-targeted UAPs, the proxy dataset has to be the original training dataset.

4.1. Results

We generate the targeted UAPs for four different datasets, the ImageNet training dataset as well as three proxy datasets. As the proxy datasets, we use two object detection datasets and one scene recognition dataset (MS-COCO [12], Pascal VOC [5], Places365 [29]). Two major observations can be made: First, a significant difference can not be observed for the three different proxy datasets. Moreover, there is only a marginal performance gap between training with the proxy datasets and training with the original ImageNet training data. The results support our assumption that the influence of the input images on targeted UAPs is like noise.

Table 2. Comparison of the proposed method to other methods. The results are divided into universal attacks with access to the original ImageNet training data (upper) and data-free methods (lower). The metric is reported in the fooling ratio (%))

Method	AlexNet[1]	GoogleNet	VGG16	VGG19	ResNet152
UAP [14]	93.3	78.9	78.3	77.8	84.0
GAP [19]	-	82.7	83.7	80.1	-
Ours(ImageNet [11])	**96.17**	**88.94**	**94.30**	**94.98**	**90.08**
FFF [18]	80.92	56.44	47.10	43.62	-
AAA [21]	89.04	75.28	71.59	72.84	60.72
GD-UAP [17]	87.02	71.44	63.08	64.67	37.3
Ours (COCO [12])	89.9	**76.8**	**92.2**	**91.6**	**79.9**
Ours (VOC [5])	89.9	76.7	**92.2**	90.5	79.1
Ours (Places365 [29])	**90.0**	76.4	92.1	91.5	78.0

To the best of our knowledge, this is the first work to achieve targeted UAP without original training data, thus we can only compare our performance with previous works on related tasks. Other previous works report the (non-targeted) fooling ratio and we compare our performance with them in Table 2. We distinguish between methods with and without data availability. To compare with the methods with data-availability we trained a non-targeted UAP on ImageNet utilizing our introduced non-targeted loss function from Equation 2. We note that our approach achieves superior performance than both UAP [14] and GAP [19]. For the case without access to the original training dataset, we show the performances for different proxy datasets to generate the UAP and report the average number over the performance over 8 target classes. Note that our method is still targeted UAP but we use the non-targeted metric to evaluate the performance. This setting is in favor of other methods since ideally, we could report the best performance of a certain target class.

5. Conclusion

By treating the DNN logit output as a vector we analyze the mutual influence between image and perturbations and reveal that universal perturbations behave like features, and images behave like noise to them. This insight is somewhat contrary to the wide perception to treat the perturbation as "noise" to the images. Based on this understanding, we propose to generate targeted UAPs by exploiting a proxy dataset instead of the original training data.

References

[1] TW Anderson. An introduction to multivariate statistical analysis (wiley series in probability and statistics). 2003. 1, 2

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018. 1

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Symposium on Security and Privacy (SP)*, 2017. 4

[4] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010. 4

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[8] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 3

[9] Saumya Jetley, Nicholas Lord, and Philip Torr. With friends like these, who needs adversaries? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1, 3

[10] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017. 1

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 2, 4

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 4

[13] 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. 3

[14] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 4

[15] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. Analysis of universal adversarial perturbations. *arXiv preprint arXiv:1705.09554*, 2017. 1

[16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[17] 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. 4

[18] 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. 1, 4

[19] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[20] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J Black. Attacking optical flow. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[21] Konda Reddy Mopuri, Phani Krishna Uppala, and R Venkatesh Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. In *European Conference on Computer Vision (ECCV)*, 2018. 4

[22] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[23] Pedro Tabacof and Eduardo Valle. Exploring the space of adversarial images. In *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016. 1

[24] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016. 1

[25] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Cd-uap: Class discriminative universal adversarial perturbation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 1

[26] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[27] Chaoning Zhang, Francois Rameau, Junsik Kim, Dawit Mureja Argaw, Jean-Charles Bazin, and In So Kweon. Deepptz: Deep self-calibration for ptz cameras. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020. 1

[28] Chaoning Zhang, Francois Rameau, Seokju Lee, Junsik Kim, Philipp Benz, Dawit Mureja Argaw, Jean-Charles Bazin, and In So Kweon. Revisiting residual networks with nonlinear shortcuts. In *British Machine Vision Conference (BMVC)*, 2019. 1

[29] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 4