TOWARDS DATA-FREE UNIVERSAL ADVERSARIAL PERTURBATIONS WITH ARTIFICIAL JIGSAW IMAGES

Chaoning Zhang*, **Philipp Benz***, **Adil Karjauv***, **Jae Won Cho & In So Kweon**
Korea Advanced Institute of Science and Technology (KAIST)
{chaoningzhang1990, mikolez, chojw2017}@gmail.com,
{pbenz, iskweon77}@kaist.ac.kr

ABSTRACT

Adversarial examples constitute a threat against modern deep neural networks (DNNs). Despite numerous explorations on image-dependent adversarial perturbation (DAP), the investigation on universal adversarial perturbation (UAP) is relatively limited. The universal attack can be seen as a more practical attack because the perturbation can be generated beforehand and applied directly during the attack stage. How to generate UAP without access to the training data remains an open problem. In this work, we attempt to address this issue progressively. First, we propose a self-supervision loss to alleviate the need for ground-truth labels with the assumption that it is easier to get access to a training dataset without labels. Second, we attempt to address this issue by utilizing a very small amount of images. Our results show that our simple approach outperforms previous work by a large margin. Third, we attempt to generate a data-free UAP, *i.e*. without access to the training dataset at all. To this end, we propose to utilize artificial jigsaw images as the proxy dataset, and our approach outperforms existing methods by a large margin.

1 INTRODUCTION

The past few years have witnessed significant advancement of Deep neural networks (DNNs) (LeCun et al., 2015) in a wide range of applications; however, they are also widely known to be vulnerable to adversarial examples (Szegedy et al., 2013), *i.e*. human imperceptible perturbation can fool the target DNN. This intriguing phenomenon has inspired active research in adversarial attack and defense techniques (Goodfellow et al., 2015; Madry et al., 2018; Carlini & Wagner, 2017). More strikingly, (Moosavi-Dezfooli et al., 2017) shows that a single perturbation can be generated to attack the model for most images. Due to its image-agnostic nature, it is often termed universal adversarial perturbation (UAP). The existence of UAP is especially worrisome because unlike image-dependent adversarial perturbations (DAP), the UAP can be generated beforehand and then applied directly with a single summation for performing an attack. Such a property makes the UAP a favorable choice for a real-world attack, constituting a concern that cannot be ignored in security-sensitive applications.

Most existing approaches for generating UAPs are dependent on the training dataset. In practice, however, the model owner is unlikely to release their training dataset to the public taking the concern of adversarial attack into account. Our work focuses on addressing the challenge of generating UAP with limited or no dependence on the original training dataset.

2 RELATED WORK

Since the discovery of adversarial examples, numerous works have investigated various attack methods, such as L-BFGS (Szegedy et al., 2013), DeepFool (Moosavi-Dezfooli et al., 2016), C&W (Carlini & Wagner, 2017), FGSM (Goodfellow et al., 2015), I-FGSM or BIM (Kurakin et al., 2017; 2016), MI-FGSM (Dong et al., 2018), PGD attack (Madry et al., 2018), etc. Most of them are dependent on image-specific perturbations, while we summarize the recent development on UAP.

*Equal Contribution

Development of Universal Attacks. (Moosavi-Dezfooli et al., 2017) has first discovered the existence of UAP and applied DeepFool (Moosavi-Dezfooli et al., 2016) iteratively by accumulating the perturbations. Generative Adversarial Perturbations (GAP) were proposed by Poursaeed *et al.* (Poursaeed et al., 2018), using generative models to craft the UAP. In another variant, UAPs are crafted by leveraging the Jacobian matrices of the networks' hidden layers (Khrulkov & Oseledets, 2018). Since a normal UAP attacks all images, it can easily cause suspicion and CD-UAP has been proposed in (Zhang et al., 2020a) attacking images from a predefined group of classes while minimizing the adversarial effect on other classes. (Benz et al., 2020b) extends it to a double targeted UAP that targets the class on both the source and sink side as well as demonstrating its use case as a physical adversarial patch. Assuming no access to the original training data, Fast Feature Fool has been proposed in (Mopuri et al., 2017) to generate data-free UAPs by optimizing the feature change caused by the applied UAP. More follow-up works (Mopuri et al., 2017; 2018; Reddy Mopuri et al., 2018; Liu et al., 2019) have attempted at addressing this data-free challenge. Exploiting proxy dataset, (Zhang et al., 2020b; Benz et al., 2020a) achieves the first data-free targeted UAP and shows that UAPs are not bugs, they are features. This intriguing phenomenon has been partially attributed to the fact that deep classifiers are sensitive to high-frequency content of low magnitude (Zhang et al., 2021a). For defending against UAP, universal adversarial training (UAT) has been proposed in (Shafahi et al., 2020). Class-wise UAT has been introduced in (Benz et al., 2021), demonstrating better robustness against UAP as well as superior accuracy. Moreover, UAP has been found to exist in a wide range of applications (Hendrik Metzen et al., 2017; Li et al., 2019; Neekhara et al., 2019; Abdoli et al., 2019;?; Gao & Oates, 2019; Vadillo & Santana, 2019; Li et al., 2020). See (Zhang et al., 2021b) for a detailed survey on universal adversarial attacks.

3 SELF-SUPERVISED UAP METHOD

UAP task definition. Following (Moosavi-Dezfooli et al., 2017), we adopt X to denote a distribution of images in $\mathbb{R}^d$ and f to is used to define a deep classifier as a function that outputs an predicted label $f(x)$ for each image $x \in X$. The goal of UAP is to seek a single perturbation vector ν, *i.e.* UAP, such that

$$f(x+\nu) \neq f(x) \text{ for most } x \sim X \quad \text{s.t.} \quad ||\nu||_p \leq \epsilon. \tag{1}$$

ν obeys the constraint that its l_p-norm is smaller than a predefined magnitude value ϵ for making it quasi-imperceptible. For consistency we follow prior works (Moosavi-Dezfooli et al., 2017; Poursaeed et al., 2018; Mopuri et al., 2018) and adopt $l_\infty = 10/255$. Unless otherwise specified, the UAP in this work is by default untargetd. Following prior works, to evaluate the UAP effectiveness the metric is adopted to be *fooling ratio* defined as the percentage of the samples that change its prediction under the UAP attack, *i.e.* $\underset{x\sim\mu}{\mathbb{P}}(\hat{k}(x+\nu) \neq \hat{k}(x))$. Unless otherwise specified, we evaluate the generated UAPs on the ImageNet validation dataset.

3.1 SIMPLE-UAP WITH A SELF-SUPERVISION LOSS.

First proposed by Moosavi-Dezfooli *et al.*, the vanilla UAP method (Moosavi-Dezfooli et al., 2017) accumulates image-dependent perturbations to the final universal perturbation. These image-dependent perturbations are iteratively generated via the attack method DeepFool (Moosavi-Dezfooli et al., 2016). For better readability and to differentiate from the term for the generated perturbations, we term the vanilla UAP method DeepFool-UAP. Instead of optimizing the perturbation directly, Poursaeed *et al.* proposed a generator-based method (GAP) (Poursaeed et al., 2018) training a generative network to output a UAP. Compared to the DeepFool-UAP, the generator-based method has the benefit that a batch of images can be used to train the generator network instead of processing each image individually. Combining the benefits of both techniques, we empirically find that the direct optimization of the UAP with batches of images results in a simple yet effective UAP algorithm. We term it *Simple-UAP* and Algorithm 1 outlines its approach. A similar approach has been adopted in (Shafahi et al., 2020) for performing universal adversarial training. Both DeepFool-UAP (Moosavi-Dezfooli et al., 2017) and GAP assume the availability of the original training images as well as their ground-truth labels. However, we argue that in practice the ground-truth labels might not be available. To address this issue, we propose to optimize the UAP in a self-supervised manner. To this end, we propose a novel loss to minimize the cosine similarity (cos) as follows:

$$\mathcal{L} = cos(\hat{k}(x), \hat{k}(x+\nu)) \tag{2}$$

Algorithm 1: Simple-UAP

Input: classifier $\hat{k}$, loss $\mathcal{L}$, batch size m, number of iterations N, allowable magnitude ϵ
Output: perturbation vector ν
$\nu \leftarrow 0$ ▷ initialization
for *iteration* $= 1, \ldots, N$ **do**
  $B \sim \mathcal{X}_\nu$ ▷ samples with $|B| = m$
  $g_\nu \leftarrow \mathop{\mathbb{E}}_{x \sim B}[\nabla_\nu \mathcal{L}(\hat{k}(x), \hat{k}(x+\nu))]$ ▷ Gradient
  $\nu \leftarrow \texttt{Optim}(g_\nu)$ ▷ ν update
  $\nu \leftarrow \min(\epsilon, \max(\nu, -\epsilon))$ ▷ ν clipping
end

Here, for simplicity, we abused the formulation of $\hat{k}(\cdot)$ to indicate the logit vector (before the softmax) *vs* the predicted class (after the softmax, see Eq 1). Intuitively, with this loss the perturbation ν can be optimized such that $\hat{k}(x)$ and $\hat{k}(x+\nu)$ are far from each other, consequently resulting in a change in the class prediction for x.

3.2 Performance comparison

UAP with training dataset. Assuming full availability of the training dataset we first compare the proposed Simple-UAP with the two previously discussed UAP methods (Moosavi-Dezfooli et al., 2017; Poursaeed et al., 2018). The results are shown in Table 1. Our Simple-UAP outperforms DeepFool-UAP and GAP by a large margin, despite not taking the ground-truth labels into account due to the proposed loss function in Eq. 2. Comparing Simple-UAP with GAP, the latter trains significantly more weights and can, therefore, be seen as more complex but achieves inferior performance. On a single GPU, Simple-UAP only takes a few minutes, while UAP (Moosavi-Dezfooli et al., 2017) and GAP (Poursaeed et al., 2018) require multiple hours to craft a UAP, highlighting the efficacy of Simple-UAP.

Table 1: UAP algorithm comparison on the ImageNet validation dataset with the metric of fooling ratio (%). The algorithms are trained on the ImageNet training dataset. The results for DeepFool-UAP (Moosavi-Dezfooli et al., 2017) and GAP (Poursaeed et al., 2018) are reported as in the original papers.

Method	AlexNet	GoogleNet	VGG16	VGG19	ResNet152
DeepFool-UAP (Moosavi-Dezfooli et al., 2017)	93.3	78.9	78.3	77.8	84.0
GAP (Poursaeed et al., 2018)	-	82.7	83.7	80.1	–
Simple-UAP (Ours)	**96.5**	**90.5**	**97.4**	**96.4**	**90.2**

Table 2: Fooling ratio for UAPs crafted with limited data availability of 64 images. Singular Fool proposed by (Khrulkov & Oseledets, 2018) and Simple-UAP is ours.

Network	VGG16	VGG19	ResNet50
Singular Fool	52.0	60.0	44.0
Simple-UAP	**95.4**	**94.1**	**89.8**

UAP with a Small Number of Training Samples. Crafting UAPs with a small number of training samples has been investigated in (Khrulkov & Oseledets, 2018), where the authors show that 64 image samples are sufficient for crafting an effective UAP with a reasonably high fooling ratio. The authors further highlight that DeepFool (Moosavi-Dezfooli et al., 2017) requires around 3000 image samples to achieve equivalent performance. To compare Simple-UAP with (Khrulkov & Oseledets, 2018), only 64 images are used for crafting the perturbation and the comparison results are shown in Table 2.

As demonstrated, Simple-UAP is more effective and efficient than previously proposed UAP generation methods. Thus in the remainder of this work, we adopt it for our analysis to understand the mechanisms behind UAPs. Unless otherwise specified, we adopt a VGG16 network as our target model for performing the analysis.

4 Proposed Data-free UAP

In practice, it is impractical for the attacker to access the training data (Mopuri et al., 2018). To our knowledge, data-free UAPs remain an open challenge. Multiple works (Mopuri et al., 2017; 2018;

Reddy Mopuri et al., 2018; Liu et al., 2019) have attempted to address this challenge; however, the performance is not satisfactory. For example, compared with DeepFool-UAP (Moosavi-Dezfooli et al., 2017) with access to the training data, there is still a large performance gap. We adopt the proposed Simple-UAP as the basic framework for generating UAPs. To address the data-free challenge, artificial jigsaw images are proposed to replace the absent training dataset.

Motivation for Artificial Images. In order to craft effective UAPs without access to original training samples, we aim to approximate their characteristics with artificial images. Due to the domain gap between training samples and artificial images, a performance drop is expected; however, we believe it would still work decently. The reason behind this belief is that the generated UAP should dominate over real images if it can dominate over artificial images. To reduce the performance drop, artificial images should resemble the frequency pattern of training samples. We believe that artificial images need to have two properties: (a) locally smooth; (b) mixed frequency pattern. To prove the concept without losing generality, we propose an artificial jigsaw image as a simple solution to fulfill the above two criteria. Other delicate artificial patterns might lead to superior performance; however, optimizing such patterns is beyond the scope of this work.

Experimental results with jigsaw images. We generate the jigsaw images with random frequency patterns as shown in the appendix in Figure 1. The resulting UAPs are shown in the Appendix in Figure 2. The quantitative results are shown in Table 3. Compared with GD-UAP (Mopuri et al., 2018), PD-UA (Liu et al., 2019) deploys a Monte Carlo sampling method to increase the model uncertainty. Despite its delicate design, the performance improvement is around 5% points, while our simple approach improves GD-UAP (Mopuri et al., 2018) by around 20% points. Overall, our proposed jigsaw solution outperforms the existing approaches by a significant margin. Especially, it is worth mentioning that our data-free approach achieves comparable (marginally better) performance as DeepFool-UAP which utilizes the training dataset. To our knowledge, we are the first to achieve this.

Table 3: Comparison of the proposed method to other data-free methods with the metric of fooling ratio ($\%$). The first row reports the DeepFool-UAP (Moosavi-Dezfooli et al., 2017) that uses the training samples. Thus it is not data-free; however, we use it as a benchmark to indicate the gap between the data-free methods and DeepFool-UAP (Moosavi-Dezfooli et al., 2017) with data. The gap is indicated in the bracket for the "average" column. "Prior" denotes the range prior in (Mopuri et al., 2018).

Method	AlexNet	GoogleNet	VGG16	VGG19	ResNet152	average
DeepFool-UAP (**with data**)	93.3	78.9	78.3	77.8	84.0	82.46
FFF (Mopuri et al., 2017)	80.92	56.44	47.10	43.62	–	–
GD-UAP (w/o Prior) (Mopuri et al., 2018)	84.88	58.62	45.47	40.68	29.78	51.59(−30.57)
GD-UAP (with Prior) (Mopuri et al., 2018)	87.02	71.44	63.08	64.67	37.3	64.40(−17.76)
PD-UA (w/o Prior) (Liu et al., 2019)	–	67.12	53.09	48.95	53.51	–
PD-UA (with Prior) (Liu et al., 2019)	–	–	70.69	64.98	46.39	–
Ours (Jigsaw images)	**91.07**	**87.57**	**89.48**	**86.81**	**65.35**	**84.08**(+**1.60**)

Ablation study results are provided in the Appendix in Table 4 to justify the choice of jigsaw images with variable frequency. Following (Liu et al., 2019), we optimize the UAP on VGG16 and evaluate their performance on VGG16 (white-box), VGG19, ResNet50, ResNet152 and GoogleNet and the results are available in the Appendix in Table 5. The results show that our (regular) UAP also outperforms the existing methods by a non-trivial margin for transferability, demonstrating the efficacy of our approach in black-box scenarios.

5 Conclusion

In this work, we address the challenge of generating UAPs with limited or no dependence on the training dataset. We approach this problem progressively by first alleviating the need for ground-truth labels, then utilizing only a very small number of images. Finally, we propose to use jigsaw images as the proxy dataset for generating UAPs. Our method achieves state-of-the-art performance for the task of data-free UAP.

REFERENCES

Sajjad Abdoli, Luiz G Hafemann, Jerome Rony, Ismail Ben Ayed, Patrick Cardinal, and Alessandro L Koerich. Universal adversarial audio perturbations. *arXiv preprint arXiv:1908.03173*, 2019.

Philipp Benz, Chaoning Zhang, Tooba Imtiaz, and In-So Kweon. Universal adversarial perturbations are not bugs, they are features. *CVPR workshop on Adversarial Machine Learning in Computer Vision*, 2020a.

Philipp Benz, Chaoning Zhang, Tooba Imtiaz, and In So Kweon. Double targeted universal adversarial perturbations. In *ACCV*, 2020b.

Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Universal adversarial training with class-wise perturbations. *ICME*, 2021.

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

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018.

Hang Gao and Tim Oates. Universal adversarial perturbation for text classification. *arXiv preprint arXiv:1910.04618*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *ICCV*, 2017.

Valentin Khrulkov and Ivan Oseledets. Art of singular vectors and universal adversarial perturbations. In *CVPR*, 2018.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR2017 workshop*, 2016.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 2015.

Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. Universal perturbation attack against image retrieval. In *ICCV*, 2019.

Jiguo Li, Xinfeng Zhang, Chuanmin Jia, Jizheng Xu, Li Zhang, Yue Wang, Siwei Ma, and Wen Gao. Universal adversarial perturbations generative network for speaker recognition. In *ICME*, 2020.

Hong Liu, Rongrong Ji, Jie Li, Baochang Zhang, Yue Gao, Yongjian Wu, and Feiyue Huang. Universal adversarial perturbation via prior driven uncertainty approximation. In *ICCV*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017.

Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. In *BMVC*, 2017.

Konda Reddy Mopuri, Aditya Ganeshan, and Venkatesh Babu Radhakrishnan. Generalizable data-free objective for crafting universal adversarial perturbations. *TPAMI*, 2018.

Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. Universal adversarial perturbations for speech recognition systems. *arXiv preprint arXiv:1905.03828*, 2019.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *CVPR*, 2018.

Konda Reddy Mopuri, Phani Krishna Uppala, and R Venkatesh Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. In *ECCV*, 2018.

Ali Shafahi, Mahyar Najibi, Zheng Xu, John P Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *AAAI*, 2020.

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.

Jon Vadillo and Roberto Santana. Universal adversarial examples in speech command classification. *arXiv preprint arXiv:1911.10182*, 2019.

Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Cd-uap: Class discriminative universal adversarial perturbation. In *AAAI*, 2020a.

Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *CVPR*, 2020b.

Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Universal adversarial perturbations through the lens of deep steganography: Towards a fourier perspective. *AAAI*, 2021a.

Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. A survey on universal adversarial attack. *IJCAI*, 2021b.

A APPENDIX

A.1 EXPERIMENTAL RESULTS WITH JIGSAW IMAGES

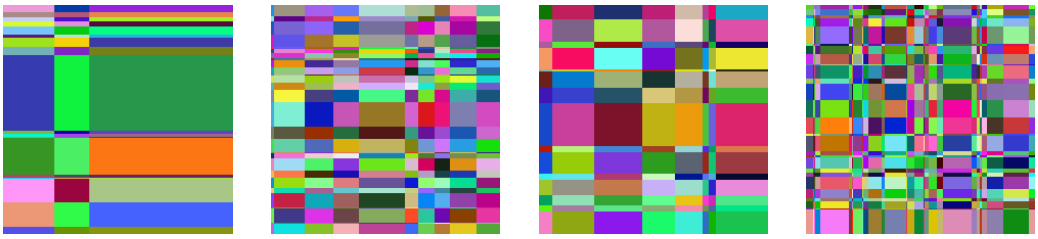

Figure 1: Four examples of jigsaw images

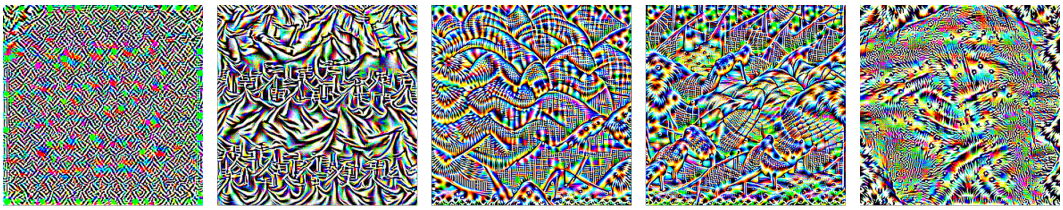

Figure 2: UAPs trained on jigsaw images for different networks.

Table 4: Ablation results for different artificial images.

Artificial images	AlexNet	GoogleNet	VGG16	VGG19	ResNet152	Average
Uniform Random Noise	82.6	40.3	72.3	64.4	47.2	61.4
Gaussian Noise (Mopuri et al., 2018)	89.5	48.7	76.1	75.4	49.9	67.9
Flat Images	80.0	39.9	81.3	79.5	29.9	62.1
Jigsaw with fixed frequency	89.1	78.6	85.6	80.9	62.9	79.4
Jigsaw with variable frequency	**91.1**	**87.6**	**89.5**	**86.8**	**65.4**	**84.1**

A.2 TRANSFERABILITY

Table 5: Transferability results for the proposed data-free UAP method with VGG16 as the source model.

		VGG16	VGG19	ResNet50	ResNet152	GoogleNet
VGG16	PD-UA	53.09	49.30	33.61	30.31	39.05
	GD-UAP+P	51.63	44.07	32.23	28.78	36.79
	UA	48.46	41.97	29.09	24.90	35.52
	GD-UAP	45.47	38.20	27.70	23.80	34.13
	Ours	**89.48**	**76.84**	**44.11**	**38.37**	**48.97**