Feature Space Targeted Attacks by Statistic Alignment

Lianli Gao, **Yaya Cheng**, **Qilong Zhang**, **Xing Xu** and **Jingkuan Song***
Center for Future Media, University of Electronic Science and Technology of China
yaya.cheng@hotmail.com, qilong.zhang@std.uestc.edu.cn, jingkuan.song@gmail.com

Abstract

By adding human-imperceptible perturbations to images, DNNs can be easily fooled. As one of the mainstream methods, feature space targeted attacks perturb images by modulating their intermediate feature maps, for the discrepancy between the intermediate source and target features is minimized. However, the current choice of pixel-wise Euclidean Distance to measure the discrepancy is questionable because it unreasonably imposes a spatial-consistency constraint on the source and target features. Intuitively, an image can be categorized as "cat" no matter the cat is on the left or right of the image. To address this issue, we propose to measure this discrepancy using statistic alignment. Specifically, we design two novel approaches called Pair-wise Alignment Attack and Global-wise Alignment Attack, which attempt to measure similarities between feature maps by high-order statistics with translation invariance. Furthermore, we systematically analyze the layer-wise transferability with varied difficulties to obtain highly reliable attacks. Extensive experiments verify the effectiveness of our proposed method, and it outperforms the state-of-the-art algorithms by a large margin. Our code is publicly available at https://github.com/yaya-cheng/PAA-GAA.

1 Introduction

Deep neural networks (DNNs) [He *et al.*, 2016; Huang *et al.*, 2017; Simonyan and Zisserman, 2015; Szegedy *et al.*, 2016] have made impressive achievements in these years, and various fields are dominated by them, *e.g.*, object detection [Redmon *et al.*, 2016]. However, recent works demonstrate that DNNs are highly vulnerable to the adversarial examples [Szegedy *et al.*, 2014; Biggio *et al.*, 2013] which are only added with human-imperceptible perturbations. To find out the insecure "bugs" in the DNNs, many works pay attention to the generation of adversarial examples.

In general, the attack methods can be grouped into three broad categories: white-box, gray-box, and black-box attacks. For the white-box setting [Moosavi-Dezfooli *et al.*, 2016; Carlini and Wagner, 2017], the adversaries can access all information (*e.g.*, the architectures and parameters) of the victim's models. Thus the update directions of the adversarial examples are accurate. For the gray-box setting [Ilyas *et al.*, 2018; Ru *et al.*, 2020], only the output logits or labels are available. Therefore, most of the works craft adversarial examples through a considerable amount of queries. However, in many scenarios, both the white-box and the gray-box attacks are infeasible owing to the opaque deployed models. For the black-box setting, all information of the victim's models is unavailable. Since the decision boundaries of different DNNs are similar, the resultant adversarial examples crafted for the substitute models, *e.g.*, well-trained models, are also practical for others, which is called the transferability of adversarial examples. Most black-box attack methods [Dong *et al.*, 2018; Inkawhich *et al.*, 2019; Gao *et al.*, 2020a; Gao *et al.*, 2020b; Lin *et al.*, 2020] aim at enhancing the transferability of adversarial examples depending on information from the classification layers of the substitute models. However, it is still challenging to improve the success rate of black-box targeted attacks, *i.e.*, induce the victim's models to predict the pre-set target labels.

To tackle the poor effectiveness of black-box targeted attacks, researchers [Sabour *et al.*, 2016; Inkawhich *et al.*, 2019] delve into the feature space targeted attacks, which perturb images by modulating their intermediate feature maps. For example, given a source image, [Inkawhich *et al.*, 2019] first select a single sample of the target label whose intermediate activation is furthest from the source one under Euclidean distance. Then, perturbation is crafted by minimizing the Euclidean distance between the source and target features. However, since Euclidean distance prefers to focus on the spatial gap between two features, it will select the spatially furthest target image rather than the one with the outermost semantic meaning. For instance, considering a source image with a cat on the left and the target label is "dog", under the above setting, the algorithm tends to choose a target image that has a dog on the right instead of on the left. When it comes to the generation of perturbation, what the algorithm needs to do is the semantic meaning alignment between the source and target features and the minimization of the huge spatial discrepancy. Overall, the current choice of pixel-wise Eu-

*corresponding author

clidean distance to measure the discrepancy is questionable, as it unreasonably imposes a spatial-consistency constraint on the source and target features.

To produce spatial-agnostic measurements, we propose two novel approaches called Pair-wise Alignment Attack and Global-wise Alignment Attack, which attempt to measure similarities between features by high-order statistics with translation invariance. With this perspective, we deal with the feature space targeted attacks as the problem of statistic alignment. By aligning the source and target high-order statistics, rather than depending on the Euclidean distance, we can make the two feature maps semantically close without introducing an excessive spatial gap in feature space.

To sum up, our contribution is summarized as three-folds: 1) We point out that the current choice of pixel-wise Euclidean Distance to measure the discrepancy between two features is questionable, for it unreasonably imposes a spatial-consistency constraint on the source and target features. By exploring high-order statistics with translation invariance, two novel methods are proposed: a) Pair-wise Alignment Attack and b) Global-wise Alignment Attack, which deal with feature space targeted attacks as a problem of statistic alignment; 2) To obtain high-reliability results, we systematically analyze the layer-wise transferability. Furthermore, to set all images under the same transfer difficulty, which ranges from the easiest to the hardest, we assign the target labels of the same difficulty level to them and give a comprehensive evaluation of our methods. and 3) Extensive experimental results show the effectiveness of our methods, which outperform the state-of-the-art by 6.92% at most and 1.70% on average in typical setups.

2 Related Works

After the discovery of adversarial examples [Szegedy *et al.*, 2014; Biggio *et al.*, 2013], many excellent works are proposed. Generally, based on different goals, attack methods can be divided into non-targeted attacks and targeted attacks. For non-targeted attacks (*e.g.*, [Xie *et al.*, 2019]), all need to do is fooling DNNs to misclassify the perturbed images. For targeted attacks, the adversaries must let the DNNs predict specific untrue labels for the adversarial examples. [Li *et al.*, 2020] apply Poincaré distance and Triplet loss to regularize the targeted attack process. [Gao *et al.*, 2021] propose staircase sign method to utilize the gradients of the substitute models effectively. The above methods craft adversarial examples by directly using the outputs of the classification layers, *i.e.*, logits (un-normalized log probability).

In addition to these, researchers [Yosinski *et al.*, 2014] observe that distorting the features in the intermediate layers of DNNs can also generate transferable adversarial examples. Based on this, [Inkawhich *et al.*, 2019] generate adversarial examples by minimizing the Euclidean distance between the source and target feature maps. [Inkawhich *et al.*, 2020a] leverage class-wise and layer-wise deep feature distributions of substitute models . [Inkawhich *et al.*, 2020b] extract feature hierarchy of DNNs to boost the performance of targeted adversarial attacks further. However, the above methods need to train specific auxiliary classifiers for each target label, thus suffering from expensive computation costs.

3 Methodology

In this section, we first give some notations of targeted attacks, and the untargeted version can be simply derived. Then we describe our proposed methods, i.e., Pair-wise Alignment Attack and Global-wise Alignment Attack, in Subsection 3.2 and 3.3. The attack process is detailed in Subsection 3.4.

3.1 Preliminaries

Adversarial targeted attacks. This task aims at *fooling* a DNN $\mathcal{F}$ to misclassify perturbed image $\boldsymbol{x}^{\boldsymbol{adv}} = \boldsymbol{x} + \boldsymbol{\delta}$, where $\boldsymbol{x}$ is the original image of label y, $\boldsymbol{\delta}$ is an imperceptible perturbation added on $\boldsymbol{x}$. In our work, ℓ_∞-norm is applied to evaluate the imperceptibility of perturbation, i.e., $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$. Different from the untargeted attacks that only need to let $\mathcal{F}$ will not perform correct recognition, targeted attacks restrict the misclassified label to be y^{tgt}. The constrained optimization of targeted attacks can be written as:

$$\boldsymbol{x}^{\boldsymbol{adv}} = \arg\min \mathcal{L}(\boldsymbol{x}^{\boldsymbol{adv}}, y^{tgt}), s.t. \left\|\boldsymbol{x}^{\boldsymbol{adv}} - \boldsymbol{x}\right\|_\infty \leq \epsilon, \tag{1}$$

where $\mathcal{L}(\cdot,\cdot)$ is the loss function to calculate perturbations.

Perceptions of DNNs. DNNs, especially convolutional neural networks (CNNs), have their patterns to perceive and understand images [Zeiler and Fergus, 2014], which is caused by the mechanism of convolutional layers. As introduced in [Worrall *et al.*, 2017], convolution kernels do not perform a one-time transformation to produce result from the input. Instead, a small region of input is perceived iteratively so that features at every layer still hold local structures similar to that of the input (see Supp. Sec. D). This property of convolution leads to the translation homogeneity of intermediate feature maps. Therefore, measuring only the Euclidean distance between two feature maps will be inaccurate when there are translations, rotations, *etc*.

3.2 Pair-wise Alignment Attack

Given an image $\boldsymbol{x}^{tgt}$ of target label y^{tgt}, a specific intermediate layer l from network $\mathcal{F}$. We use $\boldsymbol{S}^l \in \mathbb{R}^{N_l \times M_l}$ to denote the feature of $\boldsymbol{x}^{\boldsymbol{adv}}$ at layer l of $\mathcal{F}$. Similarly, $\boldsymbol{T}^l \in \mathbb{R}^{N_l \times M_l}$ is the feature of $\boldsymbol{x}^{tgt}$. Specifically, N_l is the number of channels and M_l is the product of the height and width of features.

As described before, since Euclidean distance imposes unreasonable spatial-consistency constraint on $\boldsymbol{S}^l$ and $\boldsymbol{T}^l$, choosing it as the metric leads to redundant efforts on spatial information matching. To handle this, we propose the Pair-wise Alignment Attack ($\mathbf{PAA}$). Assuming that the label information is modeled by highly abstract features, we denote $\boldsymbol{S}^l$ and $\boldsymbol{T}^l$ are under two distributions p and q, which models the label information y and y^{tgt}, respectively. Naturally, an arbitrary feature extracted from $\mathcal{F}$ is treated as a sample set of a series of feature vectors over corresponding distribution.

So the problem is how to utilize these samples to further estimate the difference between p and q. Empirically, source and target sample sets $\Omega \sim p$, $Z \sim q$ are built by splitting $\boldsymbol{S}^l$, $\boldsymbol{T}^l$ into individual vectors, where $\Omega = \{\boldsymbol{s}_{\cdot i}\}_{i=1}^{M_l}$, $Z = \{\boldsymbol{t}_{\cdot j}\}_{j=1}^{M_l}$.

Another way of splitting in where $\Omega = \{\boldsymbol{s}_{i\cdot}\}_{i=1}^{N_l}$, $Z = \{\boldsymbol{t}_{j\cdot}\}_{j=1}^{N_l}$ is analysed in Supp. Sec. C. After that, through measuring the similarity of Ω and Z, the discrepancy between p and q is estimated. Typically, this is a two-sample problem [Gretton *et al.*, 2012].

As introduced in [Gretton *et al.*, 2012], MMD has been explored for the two-sample problem. Let $\mathcal{H}$ be a reproducing kernel Hilbert space (RKHS) with an associated continuous kernel $k(\cdot,\cdot)$. For all $f \in \mathcal{H}$, the *mean embedding* of p in $\mathcal{H}$ is an unique element μ_p which satisfies the condition of $\mathbb{E}_{\omega\sim p} f = \langle f, \mu_p \rangle_{\mathcal{H}}$. Then in our task, $\mathrm{MMD}^2[p,q]$ is defined as the RKHS distance between μ_p and μ_q:

$$\begin{aligned}
&\mathrm{MMD}^2[p,q] = \ \|\mu_p - \mu_q\|_{\mathcal{H}}^2 \\
&= \langle \mu_p, \mu_p \rangle_{\mathcal{H}} + \langle \mu_q, \mu_q \rangle_{\mathcal{H}} - 2\langle \mu_p, \mu_q \rangle_{\mathcal{H}} \\
&= \frac{1}{M_l^2} \sum_{i,j=1}^{M_l} k(\boldsymbol{s}_{\cdot i}, \boldsymbol{s}_{\cdot j}) + \frac{1}{M_l^2} \sum_{i,j=1}^{M_l} k(\boldsymbol{t}_{\cdot i}, \boldsymbol{t}_{\cdot j}) \\
&\quad - \frac{2}{M_l^2} \sum_{i,j=1}^{M_l, M_l} k(\boldsymbol{s}_{\cdot i}, \boldsymbol{t}_{\cdot j}).
\end{aligned} \tag{2}$$

Specifically, MMD is calculated by two kinds of pairs: a) intra-distribution pairs $(\boldsymbol{s}_{\cdot i}, \boldsymbol{s}_{\cdot j})$, $(\boldsymbol{t}_{\cdot i}, \boldsymbol{t}_{\cdot j})$ and b) inter-distribution pair $(\boldsymbol{s}_{\cdot i}, \boldsymbol{t}_{\cdot j})$. Obviously, MMD is not affected by spatial translations, *i.e.*, shifting or rotation will not change the result of equation 2, which is the key difference from Euclidean distance. Furthermore, based on the critical property $\mathrm{MMD}^2[p,q] = 0$ *iff* $p = q$ [Gretton *et al.*, 2012], minimizing equation 2 equals to modulating source feature to target's:

$$\mathcal{L}_{\mathcal{P}}(\boldsymbol{S}^l, \boldsymbol{T}^l) = \mathrm{MMD}^2[p,q]. \tag{3}$$

Since the kernel choice plays a key role in the mean embedding matching [Gretton *et al.*, 2012]. In our experiments, three kernel functions will be studied to evaluate their effectiveness in statistic alignment:

- Linear kernel $\mathbf{PAA}_{\ell}$: $k(\boldsymbol{s}, \boldsymbol{t}) = \boldsymbol{s}^T \boldsymbol{t}$.
- Polynomial kernel $\mathbf{PAA}_{\mathbf{p}}$: $k(\boldsymbol{s}, \boldsymbol{t}) = (\boldsymbol{s}^T \boldsymbol{t} + c)^d$.
- Gaussian kernel $\mathbf{PAA}_{\mathbf{g}}$: $k(\boldsymbol{s}, \boldsymbol{t}) = \exp\left(-\frac{\|\boldsymbol{s}-\boldsymbol{t}\|_2^2}{2\sigma^2}\right)$,

where bias c, power d and variance σ^2 are hyper-parameters. Following [Inkawhich *et al.*, 2019], by randomly sampling images from each label, a gallery is maintained for picking target images. With the help of the gallery, the pipeline of getting $\boldsymbol{x}^{tgt}$ by **PAA** is as follows: Given a source image $\boldsymbol{x}$, we obtain y^{tgt} by using different strategies of target label selection. After that, $\boldsymbol{x}^{tgt}$ is chosen from the corresponding sub-gallery by finding the image with the largest loss $\mathcal{L}_{\mathcal{P}}$. It is worth noting that we adopt the linear-time unbiased estimation of $\mathrm{MMD}^2[p,q]$ from [Gretton *et al.*, 2012] to decrease the space and computation complexity during the selection of the target image $\boldsymbol{x}^{tgt}$.

3.3 Global-wise Alignment Attack

Since Pair-wise Alignment Attack involves time-consuming pair-wise computation, we propose the other efficient approach that achieves comparable performance. Unlike the previous one, Global-wise Alignment Attack (**GAA**) explicitly matches moments of source, and target sample sets Ω, Z. Specifically, we employ two global statistics: a) first-order raw moment (*mean*) and b) second-order central moment (*variance*) to guide the modulation of features. Let $\mu^i_{\boldsymbol{S}^l}$, $\mu^i_{\boldsymbol{T}^l}$, $\sigma^i_{\boldsymbol{S}^l}$, $\sigma^i_{\boldsymbol{T}^l}$ be the mean and variance of the ith channel of $\boldsymbol{S}^l$ and $\boldsymbol{T}^l$, respectively:

$$\mu^i_{\boldsymbol{S}^l} = \frac{1}{M_l} \sum\nolimits_{j=1}^{M_l} (\boldsymbol{S}^l)_{ij}, \ \sigma^i_{\boldsymbol{S}^l} = \mathrm{Var}((\boldsymbol{S}^l)_{i\cdot}), \tag{4}$$

$$\mu^i_{\boldsymbol{T}^l} = \frac{1}{M_l} \sum\nolimits_{j=1}^{M_l} (\boldsymbol{T}^l)_{ij}, \ \sigma^i_{\boldsymbol{T}^l} = \mathrm{Var}((\boldsymbol{T}^l)_{i\cdot}). \tag{5}$$

Minimizing the gaps between Ω and Z of these two moments equals to aligning the source and target features globally:

$$\begin{gathered}
\delta_\mu = \left\|\mu^i_{\boldsymbol{S}^l} - \mu^i_{\boldsymbol{T}^l}\right\|, \delta_\sigma = \left\|\sigma^i_{\boldsymbol{S}^l} - \sigma^i_{\boldsymbol{T}^l}\right\|, \\
\mathcal{L}_{\mathcal{G}}(\boldsymbol{S}^l, \boldsymbol{T}^l) = \delta_\mu + \delta_\sigma.
\end{gathered} \tag{6}$$

The reasons for performing Global-wise Alignment are: 1) the two moments are practical to estimate the distribution on a dataset, just like what batch-normalization does; and 2) when the architectures of DNNs go deeper, these two moments will contain more complicated traits to represent different distributions [Li *et al.*, 2018]. Similar as **PAA**, **GAA** also chooses the target image from the gallery by calculating Equation (6).

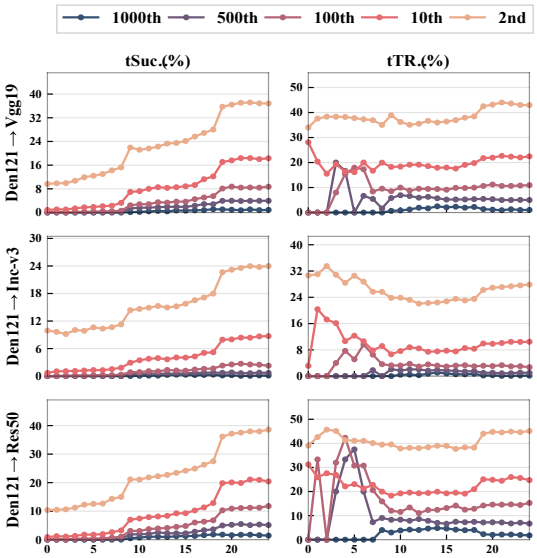

Figure 1: Performance (tSuc and tTR) of $\mathbf{PAA}_{\mathbf{p}}$ w.r.t. $2nd$, $10th$, $100th$, $500th$, and $1000th$ settings. Target label of higher ranking leads to better performance.

3.4 Attack Algorithm

Motivated by MIFGSM [Dong *et al.*, 2018] which using momentum to memorize previous gradients and follow the setting of AA [Inkawhich *et al.*, 2019], we integrate momentum to the pipeline of perturbation generation. Specifically, for two kinds of attacks, *i.e.*, **PAA** and **GAA**, we firstly calculate

	Den121→VGG19		Den121→Inc-v3		Den121→Res50	
	tSuc	tTR	tSuc	tTR	tSuc	tTR
TIFGSM	0.40	0.41	0.08	0.08	0.24	0.24
MIFGSM	1.48	1.50	0.54	0.55	2.44	2.46
AA	1.18	1.61	0.50	0.68	1.78	2.32
GAA	3.20	4.17	0.70	0.91	4.22	5.62
$\mathbf{PAA_g}$	1.60	2.52	0.52	0.73	2.42	3.60
$\mathbf{PAA}_\ell$	3.20	3.97	0.74	0.94	4.40	5.65
$\mathbf{PAA_p}$	**4.38**	**5.56**	**1.16**	**1.45**	**6.08**	**7.95**
	VGG19→Inc-v3		VGG19→Den121		VGG19→Res50	
TIFGSM	0.08	0.08	0.26	0.26	0.12	0.12
MIFGSM	0.30	0.30	**1.16**	1.17	**0.68**	0.69
AA	0.08	0.14	0.38	0.48	0.16	0.24
GAA	0.12	0.20	0.72	1.56	0.44	0.88
$\mathbf{PAA_g}$	0.14	0.30	0.52	1.27	0.32	0.73
$\mathbf{PAA}_\ell$	0.12	0.19	0.34	0.74	0.18	0.49
$\mathbf{PAA_p}$	**0.28**	**0.36**	1.00	**1.87**	0.56	**0.88**
	Inc-v3→VGG19		Inc-v3→Den121		Inc-v3→Res50	
TIFGSM	0.16	0.18	0.08	0.09	0.08	0.09
MIFGSM	0.56	0.56	0.56	0.57	0.54	0.54
AA	0.24	0.66	0.28	1.02	0.24	0.66
GAA	0.60	2.49	0.72	2.38	0.68	2.60
$\mathbf{PAA_g}$	0.34	1.03	0.38	1.24	0.34	1.24
$\mathbf{PAA}_\ell$	0.22	0.71	0.32	1.95	0.32	2.13
$\mathbf{PAA_p}$	**0.70**	**2.55**	**0.86**	**3.37**	**0.82**	**3.10**
	Res50→VGG19		Res50→Inc-v3		Res50→Den121	
TIFGSM	0.32	0.33	0.08	0.08	0.44	0.45
MIFGSM	2.00	2.02	0.92	0.93	3.96	3.99
AA	0.78	2.05	0.54	1.29	1.96	4.93
GAA	2.14	5.28	0.76	1.78	3.92	9.80
$\mathbf{PAA_g}$	0.94	3.15	0.44	1.28	1.68	5.96
$\mathbf{PAA}_\ell$	2.16	5.91	0.62	1.71	3.10	8.54
$\mathbf{PAA_p}$	**4.38**	**8.46**	**1.36**	**2.48**	**7.36**	**14.88**

Table 1: Quantitative comparisons with state-of-the-art attacks under the random sample strategy of target label selection. Ours achieve the best performance in most cases.

gradients step-by-step:

$$\boldsymbol{g}_\nu = \nabla_{\boldsymbol{x}_\nu^{adv}} \mathcal{L}(\boldsymbol{S}_\nu^l, \boldsymbol{T}^l), \tag{7}$$

where ν is the current step during the whole iteration, $\boldsymbol{S}_\nu^l$ is the intermediate feature of the perturbed image $\boldsymbol{x}_\nu^{adv}$ at iteration ν, and $\boldsymbol{x}_0^{adv} = \boldsymbol{x}$. Then the momentum term is accumulated by previous gradients:

$$\beta_{\nu+1} = \mu \cdot \beta_\nu + \frac{\boldsymbol{g}_\nu}{\|\boldsymbol{g}_\nu\|}, \tag{8}$$

where μ refers to the decay factor, β_ν is the momentum term at iteration ν and β_0 is initialized to 0. Finally, under the ℓ_∞-norm constraint, adversarial examples are crafted by performing the above calculations iteratively:

$$\boldsymbol{x}_{\nu+1}^{adv} = \text{clip}_{\boldsymbol{x},\epsilon}(\boldsymbol{x}_\nu^{adv} - \alpha \cdot \text{sign}(\beta_{\nu+1})), \tag{9}$$

where α is a given step size.

4 Experiments

To make comprehensive comparisons with state-of-the-arts, we conduct a series of experiments to evaluate performance. Specifically, baselines include a feature space targeted attack method: AA [Inkawhich *et al.*, 2019] and two FGSM-based methods: MIFGSM [Dong *et al.*, 2018] and TIFGSM [Dong *et al.*, 2019]. Supp. Sec. G and Sec. H give comparisons with other FGSM-based methods.

		$2nd$	$10th$	$100th$	$500th$	$1000th$
GAA	tSuc	28.38	14.38	7.6	3.78	0.78
$\mathbf{PAA_g}$		27.18	11.3	4.28	1.62	0.32
$\mathbf{PAA}_\ell$		33.28	15.56	6.86	3.10	0.86
$\mathbf{PAA_p}$		**37.98**	**21.10**	**11.2**	**5.12**	**1.74**
GAA	tTR	36.15	19.82	10.75	5.26	1.03
$\mathbf{PAA_g}$		34.94	16.56	6.68	2.64	0.58
$\mathbf{PAA}_\ell$		39.39	19.68	9.32	4.27	1.05
$\mathbf{PAA_p}$		**44.90**	**26.01**	**14.66**	**6.74**	**2.15**

Table 2: Transferability (tSuc and tTR) w.r.t. $2nd$, $10th$, $100th$, $500th$, and $1000th$ settings. Formally, different target labels lead to different performance and those of lower-ranking lead to worse performance.

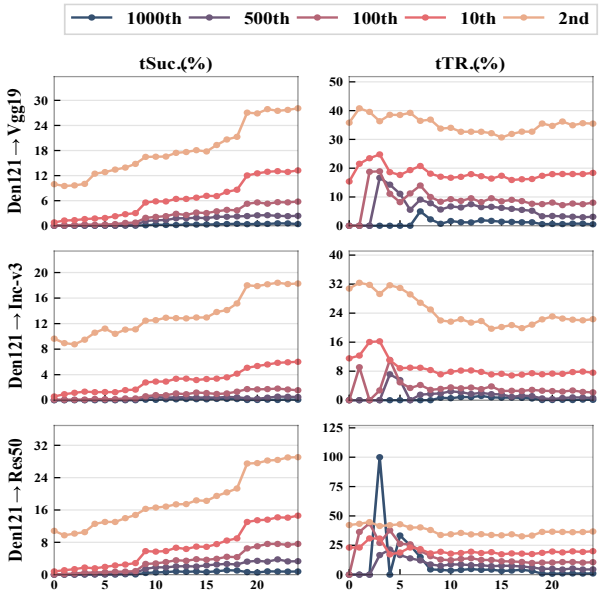

Figure 2: Performance (tSuc and tTR) of **GAA** w.r.t. $2nd$, $10th$, $100th$, $500th$, and $1000th$ settings. Target label of higher ranking leads to better performance.

ImageNet models. For a better evaluation of transferability, four ImageNet-trained models with different architectures are chosen: VGG-19 with batch-normalization (VGG19) [Simonyan and Zisserman, 2015], DenseNet-121 (Den121) [Huang *et al.*, 2017], ResNet-50 (Res50) [He *et al.*, 2016], Inception-v3 (Inc-v3) [Szegedy *et al.*, 2016].

Dataset. Attacking images that have already been misclassified is pointless. Hence for each of all 1000 labels in the ImageNet validation set, we randomly select five images (5,000 in total) to perturb, which are correctly classified by all the networks we considered.

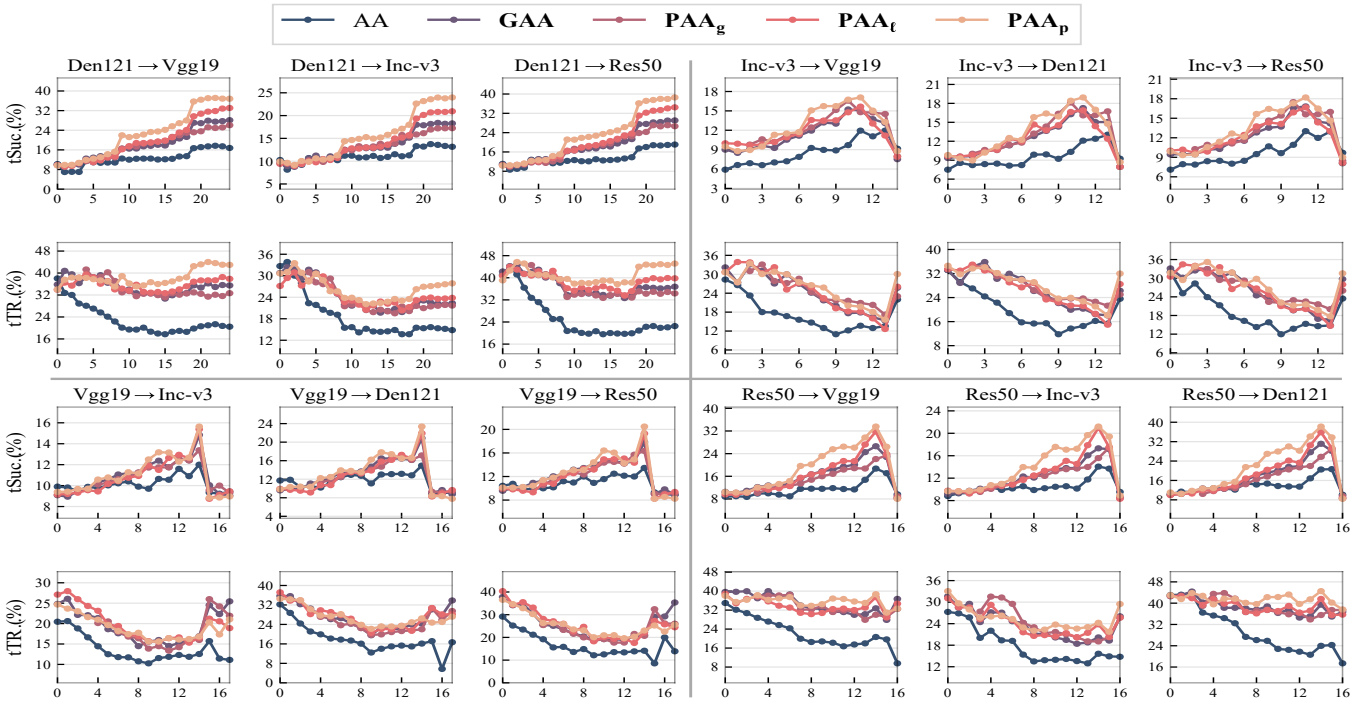

Figure 3: tSuc and tTR performance w.r.t. relative layer depth for multiple transfer scenarios. The figure is split into four phases: upper left, upper right, bottom left, and bottom right, corresponding to black-box attacks transferring from Den121, Inc-v3, VGG19, and Res50. All of our proposed methods outperform AA in most cases, which indicates the effectiveness of statistic alignment on various layers.

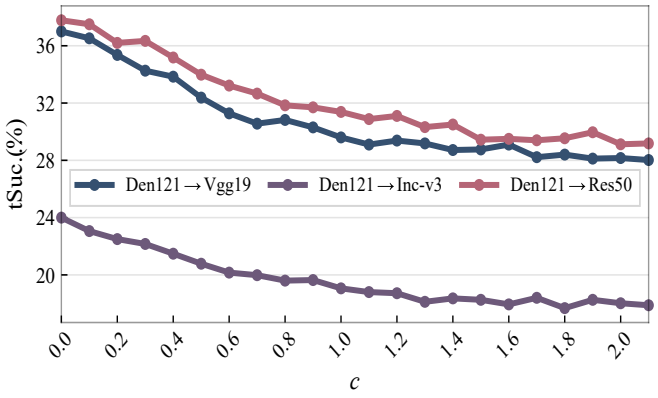

Figure 4: tSuc results w.r.t. bias c for $\mathbf{PAA_p}$ transferring from Den121 (white-box model) to VGG19, Inc-v3, and Res50 (black-box model). We observe the highest results when $c=0$, *i.e.*, polynomial with pure second-order terms.

Layer decoding scheme. Following AA, a scheme for layer decoding is employed to present better which layer is chosen for the attack. Generally, layers are arranged from shallow to deep and numbered by relative layer depths, *e.g.*, layer 0 of Res50 (denoted as Res50$_{[0]}$) is near the input layer, and Res50$_{[16]}$ is closed to the classification layer. Supp. Sec. A details the scheme.

Target label selection. There are two strategies for target label selection: a) random sample adopted in AA. b) choose by ranking. Previous feature space targeted attack methods, *e.g.*, [Inkawhich *et al.*, 2019], gain relatively poor performance. Given the prior knowledge that different target labels involve different transfer difficulties, randomly sampling the target label will lead to fluctuating transfer results (see Supp. Sec. F for more analysis). For instance, given an image of "cat", it is easier to fool a model to predict it as a dog than an airplane. To avoid this, we assign y^{tgt} by ranking. For example, $2nd$ indicates that the label of the second high confidence is chosen to be y^{tgt}. To give an exhaustive comparison, $2nd$, $10th$, $100th$, $500th$, and $1000th$ settings are adopted. We also report results under the random sample strategy to reveal the raw performance.

Implementation details. To make a fair comparison, all methods are set to identical ℓ_∞ constraint $\epsilon=0.07$, the number of iterations $T=20$, and step size $\alpha=\epsilon/T=0.0035$. The gallery size is set to 20×1000. For $\mathbf{PAA_g}$, we set variance σ^2 as the mean of squared ℓ_2 distances of those pairs. For $\mathbf{PAA_p}$, we set bias $c=0$, and only study the case of power $d=2$. For TIFGSM, we adopt the default kernel length as 15. For MIFGSM, we set the decay factor as $\mu=1.0$.

Evaluation metrics. Following AA, we adopt two metrics, *i.e.*, targeted success rate (tSuc) and targeted transfer rate (tTR), to evaluate the transferability of adversarial examples. For tSuc, it equals the percentage of adversarial examples that successfully fool the victim's DNNs. For tTR, given an image set that contains adversarial examples that attack the substitute model successfully, tTR is the ratio that how many examples of this set can fool the black-box model too.

4.1 Comparisons with State-of-the-Art Attacks

In this section, to comprehensively evaluate adversarial examples' transferability, we firstly attack different white-box models using the random sample strategy for target labels, then transfer the resultant adversarial examples to black-box models. For instance, Den121→Res50 indicates that we generate adversarial examples from Den121 and transfer them to Res50. Empirically, attack performance varies according to the choice of layers. Under random sample strategy, VGG19$_{[10]}$, Den121$_{[23]}$, Res50$_{[11]}$ and Inc-v3$_{[8]}$ perform the best, their experimental results are shown in Table 1.

Effectiveness of $\mathbf{PAA}$ with Different Kernel Functions

As demonstrated in Table 1, all pair-wise alignments show their success in feature space targeted attack. Specifically, comparing with Linear kernel and Gaussian kernel, Polynomial kernel brings the best performance, and our $\mathbf{PAA_p}$ outperforms state-of-the-arts by 6.92% at most and 1.70% on average, which shows the effectiveness of our pair-wise alignment. As for the reasons of the performance gains, compared with FGSM-based methods, *i.e.*, TIFGSM and MIFGSM, we exploit the information in the intermediate feature maps to perform highly transferable attacks. Compared with AA, it adopts Euclidean distance for measuring differences so that shows worse performance than ours, demonstrating the effectiveness of our proposed statistic alignment.

Effectiveness of $\mathbf{GAA}$

Although $\mathbf{GAA}$ requires quite simple computations to perform attacks, it still shows convincing performance against all black-box models. Specifically, $\mathbf{GAA}$ outperforms the state-of-the-arts by 3.98% at most and 0.73% on average, which shows the effectiveness of global alignment between statistics from target and source. Moreover, when choosing Den121 and Res50 as white-box models, it shows comparable performance with $\mathbf{PAA}_\ell$. When it becomes VGG19 or Inc-v3, $\mathbf{GAA}$ achieves the second-best results in most cases.

4.2 Ablation Study

Transferability w.r.t. Target Labels

Considering different difficulties of target label y^{tgt}, for $\mathbf{PAA_p}$ and $\mathbf{GAA}$, we study how layer-wise transferability varies with "$2nd$", "$10th$", "$100th$", "$500th$", "$1000th$" setup. As illustrated in Figure 1 and Figure 2, tSuc and tTR w.r.t. relative layer depth under above settings are evaluated. Obviously, the independence of layer-wise transferability from different target labels maintains. In other words, different target labels do not affect the layer-wise transferability trends, although further y^{tgt} away from ground truth y leads to a more challenging transfer-based attack. For case Den121→Res50 under $2nd$, we report the results for the optimal layer of Den121 (Den121$_{[22]}$) in Table 2. Formally, target labels of different ranks lead to different performance , and the lower-ranking leads to worse performance. Specifically, $2nd$ is the best case, $1000th$ refers to the worst case.

Transferability w.r.t. Layers

In this section, transferability w.r.t. relative layer depth under $2nd$ is investigated. Involved methods contain $\mathbf{PAA}_\ell$, $\mathbf{PAA_p}$, $\mathbf{PAA_g}$, $\mathbf{GAA}$, and AA. Specifically, given the white-box and black-box model pair, each subfigure of Figure 3 illustrates performance under different metric w.r.t. relative layer depth. As demonstrated in the figure, compared with the Linear kernel, the Polynomial kernel brings about better attack ability on Res50, Inc-v3, and Dense121 white-box. As for the VGG19 white-box, they achieve comparable results. Furthermore, in most of the chosen layers, all of our methods are superior to the baseline AA by a large margin.

Similar to what is stated in [Inkawhich *et al.*, 2019], given a white-box model, our layer-wise transferability still holds a similar trend regardless of which black-box models we test. Specifically, for Den121, a deeper layer yields more transferability. For Inc-v3, Vgg19, and Res50, the most powerful attack comes from perturbations generated from optimal middle layers. This phenomenon indicates that adversarial examples generated by our optimal layers can be well transferred to truly unknown models. From the experimental results, under $2nd$, we simply adopt VGG19$_{[14]}$, Den121$_{[22]}$, Res50$_{[14]}$, and Inc-v3$_{[11]}$ as our optimal layers.

Transferability w.r.t. Orders

As mentioned above, the Polynomial kernel leads to the most powerful attack. Since larger bias c ($c \geq 0$) results in a greater proportion of lower-order terms in the polynomial, in this section, we study the appropriate value of c under $2nd$ and Den121$_{[22]}$ setup. Specifically, we attack Den121 using $\mathbf{PAA_p}$ parameterized by c ranging from 0.0 to 2.0 with a granularity 0.1. As illustrated in Figure 4, from the monotonically decreasing curves, we can achieve the most effective attack when $c = 0.0$, where tSuc is 37.00%, 24.00%, 37.78% for VGG19, Inc-v3 , and Res50. Once $c = 1.3$ or larger, tSuc maintains stable. The overall average tSuc for VGG19, Inc-v3, Res50 are 30.78%, 19.68%, and 32.12%.

5 Conclusion

In this paper, we propose a novel statistic alignment for feature space targeted attacks. Previous methods utilize Euclidean distance to craft perturbations. However, because of the spatial-related property of this metric, it unreasonably imposes a spatial-consistency constraint on the source and target features. To address this problem, two novel methods, *i.e.*, Pair-wise Alignment Attack and Global-wise Alignment Attack are proposed by employing high-order translation-invariant statistics. Moreover, since randomly selecting target labels results in fluctuating transfer results, we further analyze the layer-wise transferability with different transfer difficulties to obtain highly reliable attacks. Extensive experimental results show the effectiveness of our methods.

Acknowledgements

This work is supported by National Key Research and Development Program of China (No.2018AAA0102200), the National Natural Science Foundation of China (Grant No.61772116, No.61872064, No.62020106008), Sichuan Science and Technology Program (Grant No.2019JDTD0005), The Open Project of Zhejiang Lab (Grant No.2019KD0AB05) and Open Project of Key Laboratory of Artificial Intelligence, Ministry of Education (Grant No.AI2019005).

References

[Biggio *et al.*, 2013] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *ECML/PKDD*, 2013.

[Carlini and Wagner, 2017] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *SP*, 2017.

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

[Dong *et al.*, 2019] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019.

[Gao *et al.*, 2020a] Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. Patch-wise attack for fooling deep neural network. In *ECCV*, 2020.

[Gao *et al.*, 2020b] Lianli Gao, Qilong Zhang, Jingkuan Song, and Heng Tao Shen. Patch-wise++ perturbation for adversarial targeted attacks. *CoRR*, abs/2012.15503, 2020.

[Gao *et al.*, 2021] Lianli Gao, Qilong Zhang, Xiaosu Zhu, Jingkuan Song, and Heng Tao Shen. Staircase sign method for boosting adversarial attacks. *arXiv preprint arXiv:2104.09722*, 2021.

[Gretton *et al.*, 2012] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[Ilyas *et al.*, 2018] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018.

[Inkawhich *et al.*, 2019] Nathan Inkawhich, Wei Wen, Hai (Helen) Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *CVPR*, 2019.

[Inkawhich *et al.*, 2020a] Nathan Inkawhich, Kevin J. Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. In *ICLR*, 2020.

[Inkawhich *et al.*, 2020b] Nathan Inkawhich, Kevin J. Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In *NeurIPS*, 2020.

[Li *et al.*, 2018] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognit.*, 80:109–117, 2018.

[Li *et al.*, 2020] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *CVPR*, 2020.

[Lin *et al.*, 2020] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020.

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

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[Ru *et al.*, 2020] Binxin Ru, Adam D. Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. In *ICLR*, 2020.

[Sabour *et al.*, 2016] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J. Fleet. Adversarial manipulation of deep representations. In *ICLR*, 2016.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[Szegedy *et al.*, 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[Worrall *et al.*, 2017] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *CVPR*, 2017.

[Xie *et al.*, 2019] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019.

[Yosinski *et al.*, 2014] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, 2014.

[Zeiler and Fergus, 2014] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.