Towards Transferable Targeted Attack

Maosen Li[1], Cheng Deng[1]*, Tengjiao Li[1], Junchi Yan[3], Xinbo Gao[1], Heng Huang[2,4]
[1]School of Electronic Engineering, Xidian University, Xi'an 710071, China
[2]Department of Electrical and Computer Engineering, University of Pittsburgh, PA 15260, USA
[3]Department of CSE, and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, China
[4]JD Finance America Corporation, Mountain View, CA 94043, USA
{msli_1, tjli}@stu.xidian.edu.cn, {chdeng, xbgao}@mail.xidian.edu.cn,
yanjunchi@sjtu.edu.cn, heng.huang@pitt.edu

Abstract

An intriguing property of adversarial examples is their transferability, which suggests that black-box attacks are feasible in real-world applications. Previous works mostly study the transferability on non-targeted setting. However, recent studies show that targeted adversarial examples are more difficult to transfer than non-targeted ones. In this paper, we find there exist two defects that lead to the difficulty in generating transferable examples. First, the magnitude of gradient is decreasing during iterative attack, causing excessive consistency between two successive noises in accumulation of momentum, which is termed as noise curing. Second, it is not enough for targeted adversarial examples to just get close to target class without moving away from true class. To overcome the above problems, we propose a novel targeted attack approach to effectively generate more transferable adversarial examples. Specifically, we first introduce the Poincaré distance as the similarity metric to make the magnitude of gradient self-adaptive during iterative attack to alleviate noise curing. Furthermore, we regularize the targeted attack process with metric learning to take adversarial examples away from true label and gain more transferable targeted adversarial examples. Experiments on ImageNet validate the superiority of our approach achieving 8% higher attack success rate over other state-of-the-art methods on average in black-box targeted attack.

1. Introduction

With the great success of deep learning in various fields, the robustness and stability of deep neural networks (DNNs) have attracted more and more attention. However, recent studies have corroborated that almost all of the DNNs are subjected to adversarial example problems [18, 25], which means that in DNNs, by adding some imperceptible disturbances, the original image can be shifted from one side of the decision boundary to the other side, causing discriminant errors [2, 8, 22]. Due to the vulnerability of neural networks in the case of adversarial attack, it also poses a serious security problem for the application of deep neural networks. In this context, numerous adversarial attack methods have been proposed to help evaluate and improve the robustness of the DNNs [4, 12, 23].

Generally, these attack methods can be divided into two categories according to their adversarial specificity: non-targeted attack and targeted attack [26]. The targeted attack expects that the adversarial example is misidentified as specific class. While in non-targeted attack, we expect the prediction of adversarial example can be arbitrary except the original one. Moreover, recent studies have shown that the non-targeted adversarial examples generated by some attack methods have a high cross-model transferability [22, 14], that is, the adversarial examples generated by some known models also have the ability to fool models with unknown architectures and parameters. Attacking such a model only through the transferability without any prior is called black-box attack, which brings more serious security problems to the deployment of DNNs in reality [7, 11, 13].

Although black-box attacks have become a research hotspot, most existing attack methods, such as Carlini & Wagners method [3], fast gradient sign method [8] and series of fast gradient sign based methods, focus on non-targeted attacks and have achieved great success, but they are still powerless for more challenging black-box targeted attacks. By maximizing the probability of the target class, the authors in [11] extend the non-targeted attack methods to the targeted attacks, but this simple extension does not effectively exploit the characteristics of the targeted attack, resulting in the generated adversarial examples not being transferable. Therefore, it is of great significance to develop transferable targeted adversarial examples.

*Corresponding author.

In this paper, we find that the existing black-box targeted attack methods have two serious defects. First, the traditional methods use softmax cross-entropy as a loss function. Thereby, as we will show in Eq. (7), the magnitude of gradient decreases as the probability of target class increases in an iterative attack. Since the added noise is the momentum accumulation of the gradient in each iteration, and the magnitude of the gradient decreases continuously in this process, leading to the historical momentum dominating the noise. Finally, successive noise tends to be consistent in the iterative process, resulting in a lack of diversity and adaptability of noise. We term this phenomenon as *noise curing*. Second, traditional methods only require the adversarial examples to be close to the target class without requiring far away from the original class in the iterative process, which makes generated targeted adversarial examples close to its true class. Therefore, in some cases, the targeted adversarial examples can neither successfully transfer with target label nor fool the model. In order to overcome the two problems, Poincaré space is introduced for the first time as a metric space, where the distances at the surface of the ball grow exponentially as you move toward the surface of the ball (compared to their Euclidean distances), so as to address the phenomenon of noise curing in targeted attacks. We also find that clean examples, which have long been ignored as a useful information in targeted attacks, can help adversarial example away from the original class. With proposed metric learning regularization, we put true label into use by metric method to enforce the adversarial examples away from original prediction during iterative attack, which is helpful for generating transferable targeted examples. In conclusion, the main contributions of our paper are as follows:

1) Rather than treat targeted attack as a simple extension of non-targeted attack, we discover and advocate its special properties different from non-targeted attack, which allow us to develop a new approach to improve the performance of targeted attack models.

2) We formally identify the problem of noise curing in targeted attack that has not been studied before, and also for the first time, introduce Poincaré space as a metric space instead of softmax cross entropy to solve the noise curing problem.

3) We also argue that additional true label information can be exploited to promote targeted adversarial example away from the original class, which is implemented by a new triplet loss. In contrast, ground truth label information has not been considered in existing works.

4) We study the targeted transferability of the existing methods on the Imagenet dataset with extensive experiments. All results show that our method consistently outperforms the state-of-the-art methods in targeted attack.

2. Background

We briefly review on some related adversarial attack methods and provide a brief introduction to Poincaré space.

2.1. Adversarial Attack

In adversarial attack, for a given classifier $f(x) : x \in \mathcal{X} \longrightarrow y \in \mathcal{Y}$ that outputs a label y as the prediction for an input x, adversarial attack aims to find a small perturbation δ, misleading the classifier $f(x^{adv}) \neq y$, where adversarial example $x^{adv} = x + \delta$. The small perturbation δ is constrained by ℓ_∞ norm $\|\delta\|_\infty \leq \epsilon$ in this paper. So the constrained optimization problem can be denoted as:

$$\arg\max_{\delta} J(x+\delta, y), \quad s.t. \ \|\delta\|_\infty \leq \epsilon, \tag{1}$$

where J is often the cross-entropy loss for maximization.

2.1.1 Black-box Attacks

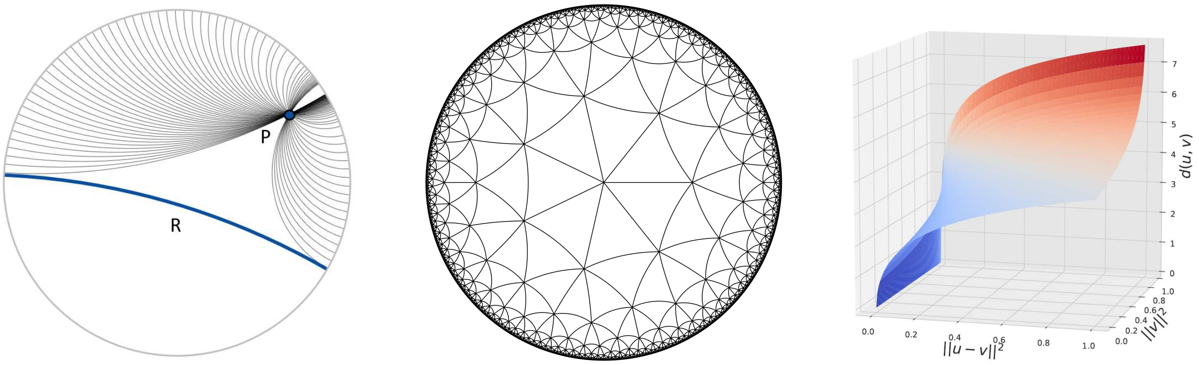
To solve the optimization problem 1, the gradient of the loss function with respect to the input needs to be calculated, termed as *white-box* attacks. For white-box attacks, adversarial examples are first introduced against DNNs [22]. Adversarial examples are generated by using L-BFGS, which is time-consuming and impractical. Then, the fast gradient sign method (FGSM) [8] is proposed, which uses the sign of gradients associated with the inputs to learn adversarial examples. The non-targeted version of FGSM is:

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y)). \tag{2}$$

However, in many cases, we have no access to the gradients of the classifier, where we need to perform attacks in the *black-box* manner. Due to the existence of transferability [17], the adversarial examples generated by the white-box attack can be transformed into the black-box attack. Therefore, in order to enable a powerful black-box attack, a series of methods are proposed to improve transferability. As a seminal work, momentum iterative FGSM (MI-FGSM) [5] is proposed, which integrates the momentum term into the iterative process for attacks to ensure the noise-adding direction more smooth:

$$\begin{aligned} g_{i+1} &= \mu \cdot g_i + \frac{\nabla_x J(x_i^{adv}, y)}{\left\|\nabla_x J(x_i^{adv}, y)\right\|_1}, \\ x_{i+1}^{adv} &= \text{Clip}_{x,\epsilon}\left\{x_i^{adv} + \alpha \cdot \text{sign}(g_{i+1})\right\}, \end{aligned} \tag{3}$$

where μ is the decay factor of the momentum term, and the *Clip* function clips the input values to a specified permissible range *i.e.* $[x-\epsilon, x+\epsilon]$ and $[0,1]$ for images. Compared



(a) Straight lines in Poincaré ball.

(b) Distance in Poincaré ball

(c) Growth of Poincaré distance

Figure 1: (a): The straight lines in Poincaré ball is composed of all Euclidean arcs in the sphere that are orthogonal to the boundary of the sphere and all the diameters of the disk. Parallel lines of a given line R may intersect at point P. (b): High capacity of Poincaré ball model. The length of each line in this figure is the same. (c): The growth of $d(u,v)$ relative to the Euclidean distance and the norm of v, $\|u\|^2 = 0.98$.

with classical FGSM, MI-FGSM is able to craft more transferable adversarial examples. Based on MI-FGSM, diverse inputs method (DI^2-FGSM) [24] transforms images with a probability p to alleviate the overfitting phenomenon. In translation invariant attack method (TI-FGSM) [6], the gradients of the untranslated images $\nabla_x J(x_t^{adv}, y)$ convolved with a predefined kernel K is used to approximate optimizing a perturbation over an ensemble of translated images. These state-of-the-art methods are already capable of generating powerful black-box adversarial examples.

2.1.2 Targeted Attacks

Targeted attacks usually occur in the multi-class classification problem, and are different from non-targeted attack, targeted attack requires target model to output specific target label. The work [14] demonstrates that, although transferable non-targeted adversarial examples are easy to find, targeted adversarial examples generated by prior approaches almost never transfer with their target labels. Therefore, they proposed ensemble-based approaches to generate transferable targeted adversarial examples. The mode extends non-targeted attacks methods to targeted attacks by maximizing the probability of target class [11]:

$$x^{adv} = x - \epsilon \cdot \text{sign}(\nabla_x J(x, y_{tar})), \tag{4}$$

where y_{tar} is target label. However, recent studies [5, 14] have shown that there is still a lack of effective method to generate targeted adversarial examples to fool the black-box model, especially for the models with adversarially trained, which is still a problem to be solved in the future research [5].

2.2. Poincaré Ball

Poincaré ball is one of typical Hyperbolic spaces. Different from Euclid geometries space, in Poincaré ball, as shown in Fig. 1.(a), there are distinct lines through point P that do not intersect line R. The arcs never reach the circumference of the ball. This is analogous to the geodesic on the hyperboloid extending out the infinity, that is, as the arc approaches the circumference it is approaching the "infinity" of the plane, which means the distances at the surface of the ball grow exponentially as you move toward the surface of the ball (compared to their Euclidean distances). Poincaré ball can fit an entire geometry in a unit ball, which means it has higher capacity than Euclid representation. Due to its high representation capacity, Poincaré ball model has attracted more interests in metric learning and representation learning to deal with the complex data distributions in computer vision tasks [1, 15].

All the points of the Poincaré ball are inside a n-dimensional unit ℓ_2 ball, and the distance between two points is defined as:

$$d(u,v) = \text{arccosh}(1+\delta(u,v)), \tag{5}$$

where u and v are two points in n-dimensional Euclid space $\mathbb{R}^n$ with ℓ_2 norm less than one, and $\delta(u,v)$ is an isometric invariant defined as follow:

$$\delta(u,v) = 2\frac{\|u-v\|^2}{\left(1-\|u\|^2\right)\left(1-\|v\|^2\right)}. \tag{6}$$

We can observe from Fig. 1.(b) that the distance of any point to the edge tends to ∞. And as shown in Fig. 1.(c), the growth of Poincaré distance is severe when it gets close to the surface of the ball. This means that the magnitude of the gradient will increase as it moves towards the surface.

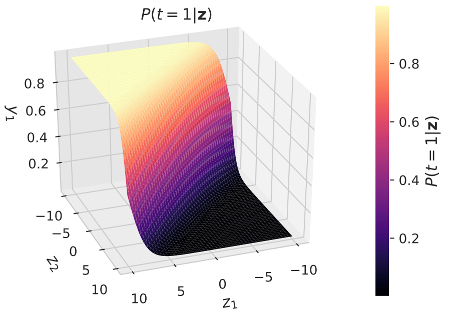


Figure 2: Probabilities of the softmax output $P(t = 1|z)$ in two classes cases $(t = 1, t = 2)$. When $P(t = 1|z)$ approaches to one, it changes slowly with z.

3. Methodology

In this section, we first elaborate the motivation and significance of this paper, then illustrate how to integrate Poincaré distance into iterative FGSM and how to use metric learning approach to regularize iterative attacks.

3.1. Motivations

There are two key differences between targeted attack and non-targeted attack. First, targeted attack has a target, which means, we should find a (local) minimal point for adversarial examples. While for non-targeted attack, the data point only needs to avoid being captured by poor local maxima, and then run away from discriminant boundary. Second, in targeted attack, we should make sure adversarial examples are not only less like original class but also more similar to target class for target model. However, we note that the existing methods do not effectively use these two differences, resulting in poor transferability of the targeted attack.

First, most of the existing methods use cross entropy as the loss function: $\xi(Y, P) = -\sum_i y_i \log(p_i)$, where p_i is prediction probability and y_i is one hot label. For the targeted attack process, derivative of cross entropy loss with respect to softmax input vector **o** can be derived as follow:

$$\frac{\partial L}{\partial o_i} = p_i - y_i. \tag{7}$$

The proof of Eq. (7) is shown in supplementary material.

As shown in Eq. (7), the gradient is linear with p_i, and when p_i is tending to y_i, the gradient is monotone decreasing. So in targeted attack, when iteration goes on, the gradient is tending to vanish. In MI-FGSM, it rescales the gradient to unit ℓ_1 ball to scale the gradients in different iterations to the same magnitude. However, this projection results in the same contribution of the gradient of each iteration to the momentum accumulation, ignoring whether the gradient is obvious in the real situation. And as shown in Fig. 3, in targeted MI-FGSM, even rescaled, the addition noise directions still have a very high cosine similarity in last few iterations due to the accumulation of momentum, which proves the existence of noise curing. This is a good property for non-targeted attack because it helps the data point to runs away from discriminant boundary along fixed direction. However, for targeted attacks when it gets close to minima of target class, the curing noise cannot efficiently find the minima, leading to poor performance in targeted attack. What's worse, as shown in Fig. 2 when output probability of target class p_{tar} approaches to one, due to the saturation of softmax, the gradient changes just a little although softmax input o_i changes a lot. In this case, if the direction of gradient is not proper in the last few iterations, the errors will be accumulated.

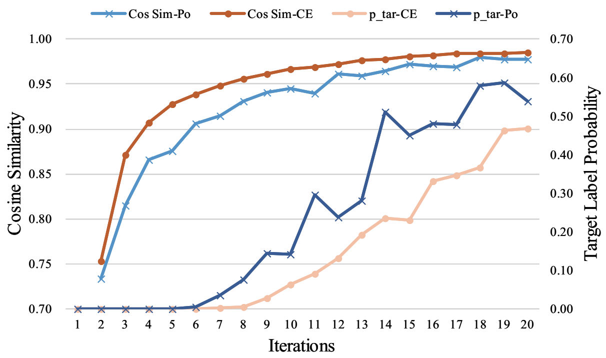


Figure 3: The cosine similarity of two successive perturbations and corresponding target class probability in MI-FGSM and Poincaré attack. To avoid cherry picking, the results are averaged over the first 10 images in the public ImageNet-compatible datasets with 1000 samples.

And the last point of our motivations is that, traditional methods only focus on maximizing the probability of targeted class and ignore whether the adversarial examples are close to the original labels. As the result shown in Fig. 4, although these methods work well in white-box setting, the targeted adversarial examples are hard to separate from corresponding true class. At the same time, original labels have long been ignored. Inspired by this, we want to make use of original labels to generate more powerful targeted adversarial examples.

3.2. Targeted Attack with Poincaré Distance Metric

Based on above analysis, we aim at improving the transferability of targeted adversarial examples by using Poincaré distance metric instead of cross entropy loss.

Note that y, a one hot encoded vector for the labels, has $\sum_i y_i = 1$, which means it is on a unit ℓ_1 ball. When y is a one hot label without smoothing, we have $\|y\|_2 = 1$. Then, the point y is at the edge of the Poincaré ball, which means the distance from any point to this point is $+\infty$. And in

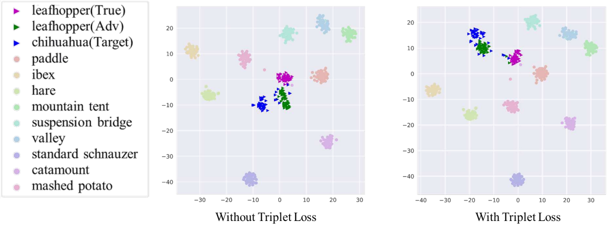


Figure 4: t-SNE visualization of adversarial images from the same true class which are mistakenly classified to target classes. In the absence of triplet loss, it is harder to separate the adversarial examples from corresponding true class.

targeted attack, we are going to reduce the distance between the logits of model and the target class. As we introduce in Sec. 2.2, the closer data point gets to the boundary, the greater the value of the gradient.

But there still exists a serious problem with using Poincaré distance as measure. The fused logits are not satisfied $\|l(x)\|_2 < 1$. So, the logits are normalized by the ℓ_1 distance in this paper. And for one hot target label y, the distance from any point to the target label is $+\infty$, which makes it hard to optimize. To avoid that, we subtract y from a small constant $\xi = 0.0001$ following [16]. The Poincaré distance metric loss

$$\mathcal{L}_{Po}(x, y) = d(u, v) = \operatorname{arccosh}(1+\delta(u, v)), \tag{8}$$

where $u = l_k(x)/\|l_k(x)\|_1$, $v = \max\{y - \xi, 0\}$, and $l(x)$ is fused logits. In this paper, the proposed algorithm generates adversarial examples by integrating multiple models whose logits are fused as:

$$l(x) = \sum\nolimits_{k=1}^{K} w_k l_k(x), \tag{9}$$

where K is the number of ensemble models, $l_k(x)$ indicates the output logits of the k-th model, and w_k is the ensemble weight of k-th model with $w_k > 0$ ($\sum_{k=1}^{K} w_k = 1$). The same setting has been proved to be effective in [5]. In this paper, except for special instructions, all the fused logits we used are the average of those ensemble models.

By using Poincaré metric, the magnitude of gradient grows if and only if data point gets closer to the target label, which is near the surface. This means that the gradient is adaptive, making the direction of the noise more flexible.

3.3. Triplet Loss for Targeted Attack

In targeted attacks, the loss function is often only related to the target label. However, the generated adversarial examples may be too close to the original class, so that some adversarial examples are still classified into the original class by the target model. Therefore, we hope our method could reduce the number of correctly classified adversarial examples and then it may gain more transferable targeted adversarial examples. Driven by such a belief, triplet loss, a classical loss function in metric learning is introduced to targeted attack process. It not only reduces the distance between the output of adversarial example and the target label, but also increases the distance between output of adversarial example and the true label. A typical triplet loss is as:

$$\mathcal{L}_{trip}(x^a, x^p, x^n) = [D(x^a, x^p) - D(x^a, x^n) + \gamma]_+, \tag{10}$$

where $\gamma \geq 0$ is a hyperparameter to define margin between distance metric $D(x^a, x^p)$ and $D(x^a, x^n)$, x^a, x^p, x^n represent anchor, positive and negative examples respectively. The standard triplet loss often uses triplet input $\{x^a, x^p, x^n\}$ for loss computation [10, 19]. But this needs to sample new data, while in targeted attack, one may get just a few images instead of whole dataset, which makes it impossible to sample triplet input from original dataset.

In view of this situation, we decide to use the logits of clean images $l(x_{clean})$, one-hot target label and true label y_{tar}, y_{true} as the triplet input:

$$\begin{aligned} &\mathcal{L}_{trip}(y_{tar}, l(x_i), y_{true}) \\ =&[D(l(x_i), y_{tar}) - D(l(x_i), y_{true}) + \gamma]_+. \end{aligned} \tag{11}$$

Note that the $l(x^{adv})$ is not normalized, so we decide to use the angular distance as distance metric:

$$D(l(x^{adv}), y_{tar}) = 1 - \frac{|l(x^{adv}) \cdot y_{tar}|}{\|l(x^{adv})\|_2 \, \|y_{tar}\|_2}. \tag{12}$$

The use of angular loss excludes the influence of the norm on the loss value. Therefore, adding triplet loss term to the loss function, we get overall loss function:

$$\mathcal{L}_{all} = \mathcal{L}_{Po}(l(x), y_{\text{tar}}) + \lambda \cdot \mathcal{L}_{trip}(y_{tar}, l(x_i), y_{true}). \tag{13}$$

Based on MI-FGSM, we use input diversity method following [24], and demonstrate our algorithm in Algorithm 1.

4. Experiments

Extensive experiments are conducted to evaluate the performance of the proposed method with some state-of-the-art adversarial methods on large-scale ImageNet dataset.

4.1. Experimental Setup

Dataset. In this paper we are aiming to generate transferable targeted adversarial examples on large-scale dataset. Therefore, we use an ImageNet-compatible dataset[1] comprised of 1,000 images to conduct experiments. This dataset is also widely used in adversarial attacks [5, 6].

Networks. As it is less meaningful to attack networks that are already poorly performing, we study 9 state-of-the-art networks on ImageNet, where we consider 6

[1] https://github.com/tensorflow/cleverhans/tree/master/examples/nips17_adversarial_competition/dataset

Algorithm 1 : The overall algorithm

```
Require: $K$ classifier, hyperparameter $\epsilon, \mu$, iterations $T$, ensemble weights $w = [w_1, w_2, \ldots, w_K]$ and a clean input image $x$.
Initialize $\alpha = \epsilon/T; g_0 = 0; x_0^{adv} = x$.
for $i = 0, 1, \ldots, T-1$ do
    Input augmented $x_i$ to $K$ classifier and get logits $l_k(x_i)$,
    Fuse these logits: $l(x_i) = \sum_{k=1}^{K} w_k l_k(x_i)$
    Compute loss function $\mathcal{L}_{all}$ through Eq. (13).
    Obtain the gradient $\nabla_x \mathcal{L}_{all}$;
    Update $g_{i+1}$ by $g_{i+1} = \mu \cdot g_i + \nabla_x \mathcal{L}_{all}$;
    Update $x_{i+1}$ by applying the sign gradient as $x_{i+1} = x_i - \alpha \cdot \text{sign}(g_{i+1})$;
end for
return Targeted adversarial example $x^{adv} = x_T$.
```

normally trained models, *i.e.*, Inception-v3 (Inc-v3) [21], Inception-v4 (Inc-v4) [20], Inception-Resnet-v2 (IncRes-v2), and Resnet-v2-{50,101,152} (Res-{50,101,152}) [9], and three adversarially trained networks [23], *i.e.*, ens3-adv-Inception-v3 (Inc-v3ens3), ens4-adv-Inception-v3 (Inc-v3ens4) and ens-adv-Inception-ResNet-v2 (IncRes-v2ens). All networks are popular in attack tasks and available[2,3].

Parameters. For the parameters of different attackers, we follow the default settings in [5] with the step size $\alpha = \epsilon/T$ and the total iteration number N = 20. We set the maximum perturbation of each pixel to be ϵ = 16, which is still invisible for human observers. For the momentum term, decay factor μ is set to 1 and for the stochastic input diversity, and the probability p is set to 0.7 as in [6]. In translation-invariant methods, we find that the report best kernel length in [6] is not suitable for targeted attack, resulting in worse attack success rate. So, we take kernel length to be 5 for TI-FGSM. In our method, the weight of triplet loss λ is set to 0.01 and margin γ is set to 0.007.

Attacking Methods. We employ two state-of-the-art iteration-based black-box attack methods mentioned in Sec. 2 to evaluate the adversarial robustness, *i.e.*, DI^2-FGSM [24] and TI-FGSM [6]. We will also show other methods as MI-FGSM and more experiments in supplementary material.

4.2. Attacking Naturally Trained Models

Here we present the results when adversarial examples transfer to other unknown naturally trained models. Our ensemble method follows the method proposed in [5], which fuses logit activations of different models. For fairness, we set ensemble weight w to be $1/K$ for all methods. Our adversarial examples are generated on an ensemble of five networks, and tested on the ensemble network (white-box setting) and the hold-out network (black-box setting).

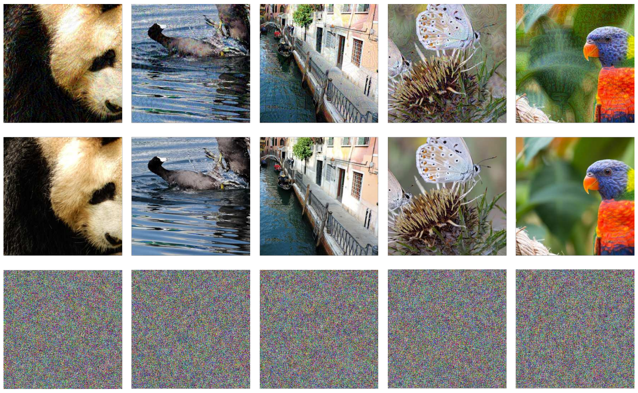


Figure 5: Our adversarial examples (first row) with $\epsilon = 16$ and their corresponding clean images (second row) and addition noise (third row). There shows little visible difference. The plots refer to the first 5 images in the dataset.

The results show both white-box attack success rates and black-box attack success rates in Table 1, where the top of the table shows white-box targeted attack success rates evaluated on ensemble network and the bottom of the table shows black-box targeted attack success rates. It can be observed that under the challenging black-box targeted setting, our method outperforms both DI^2-FGSM and TI-FGSM with a large margin over 8%. Besides, our method outperforms DI^2-FGSM and TI-FGSM by 7.0% and 5.9% in white-box setting on average.

We show some adversarial images generated by our method and their clean counterparts in Fig. 5, which are all generated by hand-out Inception-v3 setting. It can be seen that the differences between adversarial images and clean images are imperceptible to human.

4.3. Attacking Adversarially Trained Models

Adversarial training is known as one of the few defenses against adversarial attacks that withstands strong attacks. The transferability of adversarial examples is largely reduced on the adversarially trained models. Thus generating transferable targeted adversarial examples for black-box adversarially trained models is much more difficult than normally trained models, and is believed as an open issue [5]. For completeness concern, we perform our method and other attack methods on adversarially trained models.

To attack the adversarially trained models in a black-box manner, we include all nine models introduced in Sec. 4.1. We also use the hand-out setting for black-box targeted attack.

The results are shown in Table 2. It can be seen that the adversarially trained models are more robust to adver-

[2] https://github.com/tensorflow/models/tree/master/research/slimmodels

[3] https://github.com/tensorflow/models/tree/master/research/adv_imagenet_models

Model	Attack	-Inc-v3	-Inc-v4	-IncRes-v2	-Res-50	-Res-101	-Res-152	Average
Ensemble	DI^2-FGSM	83.2	82.4	84.2	76.8	79.5	80.8	81.2
	Po	88.5	88.3	85.1	82.0	82.2	85.2	85.2
	Po+Trip	**88.7**	**89.4**	**88.8**	**85.2**	**86.6**	**90.6**	**88.2**
	TI-FGSM	82.1	83.1	82.5	76.5	78.9	79.2	80.4
	Po+TI	**87.4**	85.3	85.7	80.3	82.5	88.1	84.9
	Po+TI+Trip	87.3	**87.6**	**86.4**	**83.3**	**84.4**	**89.3**	**86.3**
Hold-out	DI^2-FGSM	28.0	26.8	25.9	29.5	31.9	32.6	29.1
	Po	37.0	32.6	30.6	36.0	39.7	39.6	35.9
	Po+Trip	**38.3**	**36.6**	**32.0**	**38.5**	**41.2**	**40.6**	**38.0**
	TI-FGSM	29.8	27.5	28.8	29.6	33.7	34.4	30.6
	Po+TI	39.3	36.3	34.6	37.5	40.8	41.3	38.3
	Po+TI+Trip	**39.5**	**36.6**	**35.1**	**39.3**	**43.0**	**42.9**	**39.4**

Table 1: The success rates (%) of targeted adversarial attacks compare to DI^2-FGSM and TI-FGSM. Where "-", "Po" and "Trip" indicates the hold-out network, Poincaré distance and triplet loss respectively. The targeted adversarial examples generated on an ensemble of networks. Result shows the method significantly outperforms all in both ensemble network (white-box setting) and the hold-out network (black-box setting).

Model	Attack	Ensemble	Hold-out
-Inc-v3ens3	DI^2-FGSM	76.0	0.7
	Po	88.4	1.1
	Po+Trip	**90.3**	**1.2**
	TI-FGSM	83.1	13.7
	Po+TI	89.0	17.7
	Po+TI+Trip	**91.9**	**18.1**
-Inc-v3ens4	DI^2-FGSM	73.1	1.1
	Po	86.6	**1.5**
	Po+Trip	**87.5**	**1.5**
	TI-FGSM	82.0	10.4
	Po+TI	90.8	12.9
	Po+TI+Trip	**92.1**	**14.6**
-IncRes-v2ens	DI^2-FGSM	71.8	0.5
	Po	**87.1**	1.0
	Po+Trip	86.4	**1.2**
	TI-FGSM	81.5	6.1
	Po+TI	91.0	7.7
	Po+TI+Trip	**91.4**	**8.4**

Table 2: The success rates (%) on adversarially trained models of targeted adversarial attacks compare to DI^2-FGSM and TI-FGSM against an ensemble of white-box models and a hold-out black-box target model.

sarial examples. Adversarial examples generated by simply using input diversity cannot effectively fool the adversarially trained models. TI-FGSM shows its effectiveness by mitigating the effect of different discriminative regions between models and the adversarially trained models. But our method still outperforms all the other methods. And this result shows fooling adversarially trained models is possible.

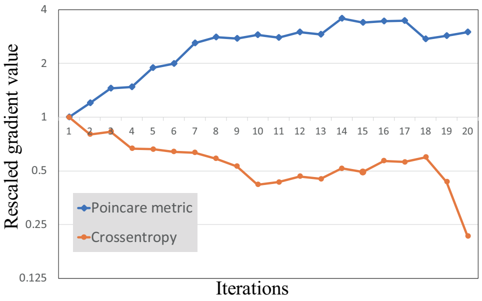


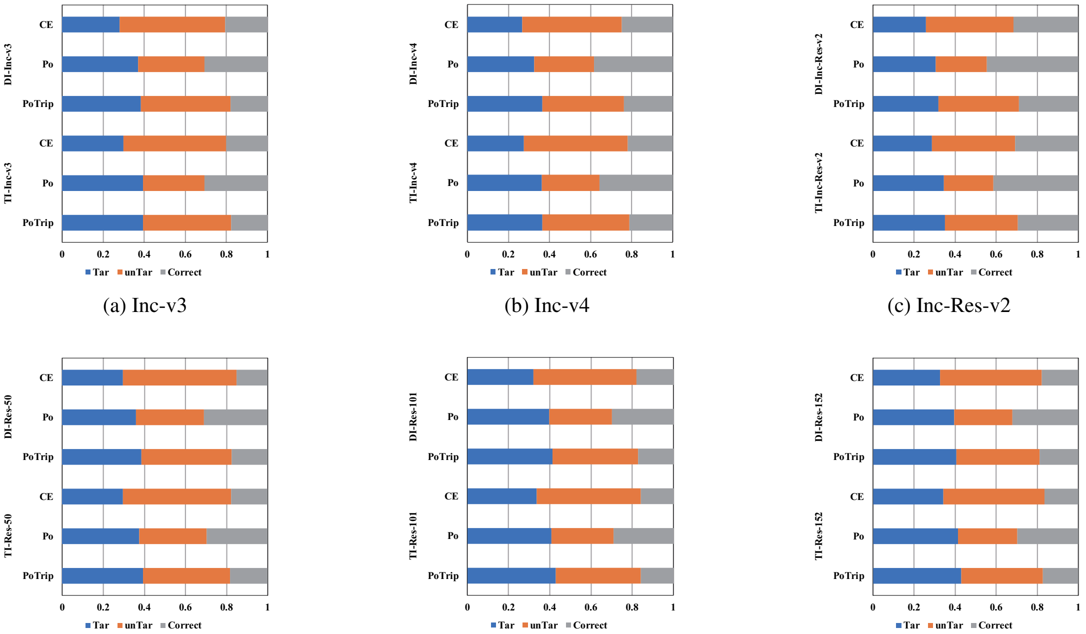
Figure 6: Gradient value over iterations. To avoid cherry picking, results are averaged on the first 10 images in the public ImageNet-compatible datasets with 1000 samples.

4.4. Ablation Study

In this section, we conduct a series of ablation experiments to study the impact of different terms.

Influence of Poincaré metric. As we have shown in Eq. (7), the gradient associated with logits output by models using cross entropy loss suffers from gradient decrease when the output target class probability p_{tar} is tending to 1. Now we'll show our conclusion made from Eq. (7) can be extended to the real situation. The gradient associated with input images is decreasing when iteration continues.

Since the gradient values of Poincaré distance and cross-entropy have a gap, we rescale all the gradient by dividing $\|g_0\|_1$, where g_0 is the gradient of first iteration. Besides, by doing that, the changes of gradient in different iterations are more intuitive to visualize. We use all six normally trained models to produce the gradient, and in order to avoid cherry



(a) Inc-v3

(b) Inc-v4

(c) Inc-Res-v2

(d) Res-50

(e) Res-101

(f) Res-152

Figure 7: Comparison on Tar, unTar and correct percentage, where “CE”, “Po” and “PoTrip” stand for using cross-entropy, Poincaré distance and both Poincaré distance and triplet loss as loss function respectively. Test on the hand-out network.

picking, the first 10 images are used in this experiment. The ℓ_1 distance $\|g_i\|_1$ of rescaled gradient is shown as its gradient value.

As shown in Fig. 6, the gradient of models using cross entropy is decreasing while the gradient of our method is mildly growing, which makes the update direction focus more on current gradient direction when the output is tending to target label. As we all know, momentum gradient descent is able to escape local minimum due to historical accumulation of gradient. What follows is that the similarity of two successive perturbations is growing as the iteration increases, causing noise curing. As shown in Fig. 3, by using Poincaré distance, two successive perturbations are less similar, and then, the adversarial example updating direction is steeper than using cross-entropy one.

Influence of triplet loss. In this experiment, we divide the adversarial examples into three parts, target transfer success samples (Tar), non-target transfer success samples (unTar) and correctly classified samples (correct). The percentage of correctly classified examples suggests whether the adversarial examples are away from true class, while the percentage of target transfer success samples suggests target examples’ transferability. We test both on DI^2 and TI setting. All the settings are the same with Sec. 4.2.

As shown in Fig. 7, though Poincaré attack successfully finds the steeper noise addition direction and generates more transfer targeted adversarial samples, it also brings a drawback that the target class minima may not be far from the discriminant boundary of true class, which leads to more adversarial examples being correctly classified by target model. But with the use of triplet loss, the adversarial examples are away from the true class and it also makes the adversarial examples more transferable.

5. Conclusion

In this paper, we take a different perspective of targeted attack rather than treating targeted attack as an extension of non-targeted attack. Based on the special properties that we discover in targeted attack, a novel method for transferable targeted attack is proposed by using Poincaré distance and triplet loss in this paper. Specifically, our method avoids the noise curing by using high capacity Poincaré space as metric space, and additional true label information is effectively exploited by metric learning based approach. Compared with traditional attacks, the extensive results on ImageNet show that the proposed attack method achieves significantly higher success rates for both black-box models and white-box models in targeted attack.

6. Acknowledgement

ML, CD, TL, JY and XG were partially supported by the Key R&D Program-The Key Industry Innovation Chain of Shaanxi (2018ZDXM-GY-176, 2019ZDLGY03-02-01), the National Key R&D Program of China (2017YFE0104100, 2016YFE0200400, 2018AAA0100704, 2016YFB1001003), NSFC (61972250, U19B2035, U1609220, 61672231), and STCSM (18DZ1112300).

References

[1] Yanhong Bi, Bin Fan, and Fuchao Wu. Beyond mahalanobis metric: cayley-klein metric learning. In *CVPR*, pages 2339–2347, 2015.

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *ECML-PKDD*, pages 387–402. Springer, 2013.

[3] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on AISec*, pages 3–14. ACM, 2017.

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *S&P*, pages 39–57. IEEE, 2017.

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, pages 9185–9193, 2018.

[6] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, pages 4312–4321, 2019.

[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *CVPR*, pages 1625–1634, 2018.

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016.

[10] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *SIMBAD*, pages 84–92. Springer, 2015.

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

[12] Chao Li, Shangqian Gao, Cheng Deng, De Xie, and Wei Liu. Cross-modal learning with adversarial samples. In *NeurIPS*, 2019.

[13] Juncheng Li, Frank Schmidt, and Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *ICML*, pages 3896–3904, 2019.

[14] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017.

[15] Emile Mathieu, Charline Le Lan, Chris J Maddison, Ryota Tomioka, and Yee Whye Teh. Hierarchical representations with poincaré variational auto-encoders. *arXiv preprint arXiv:1901.06033*, 2019.

[16] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, pages 6338–6347, 2017.

[17] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[18] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In *AAMAS*, pages 2040–2042. IFAAMAS, 2018.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015.

[20] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[21] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[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.

[23] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[24] 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*, pages 2730–2739, 2019.

[25] Erkun Yang, Tongliang Liu, Cheng Deng, and Dacheng Tao. Adversarial examples for hamming space search. *IEEE Trans. Cybern.*, 2018.

[26] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 2019.