Toward Understanding and Boosting Adversarial Transferability From a Distribution Perspective

Yao Zhu, Yuefeng Chen, Xiaodan Li, Kejiang Chen, Member, IEEE, Yuan He, Xiang Tian, Bolun Zheng, Yaowu Chen, and Qingming Huang, Fellow, IEEE

Abstract—Transferable adversarial attacks against Deep neural networks (DNNs) have received broad attention in recent years. An adversarial example can be crafted by a surrogate model and then attack the unknown target model successfully, which brings a severe threat to DNNs. The exact underlying reasons for the transferability are still not completely understood. Previous work mostly explores the causes from the model perspective, e.g., decision boundary, model architecture, and model capacity. Here, we investigate the transferability from the data distribution perspective and hypothesize that pushing the image away from its original distribution can enhance the adversarial transferability. To be specific, moving the image out of its original distribution makes different models hardly classify the image correctly, which benefits the untargeted attack, and dragging the image into the target distribution misleads the models to classify the image as the target class, which benefits the targeted attack. Towards this end, we propose a novel method that crafts adversarial examples by manipulating the distribution of the image. We conduct comprehensive transferable attacks against multiple DNNs to demonstrate the effectiveness of the proposed method. Our method can significantly improve the transferability of the crafted attacks and achieves state-of-the-art performance in both untargeted and targeted scenarios, surpassing the previous best method by up to 40% in some cases.

Index Terms—Adversarial transferability, adversarial attack, black-box attack.

I. INTRODUCTION

D eepl neural networks (DNNs) have achieved great success in many fields, such as face recognition [1], [2], [3], autonomous driving [4], [5], [6], and speaker verification [7], [8], [9]. However, Szegedy et al. [10] and [11] found that the imperceptible adversarial examples can be catastrophic for the DNNs. Even worse, researchers found that adversarial examples can even transfer between the models with different architectures and parameters [11], [12], which allows the attackers to attack unknown target models using adversarial examples generated by the surrogate models. Adversarial transferability has received more and more attention in recent years. On the one hand, such a phenomenon raises severe concerns about the security and safety of DNNs when deployed in real-world scenarios from both academia and industry. On the other hand, exploring the adversarial transferability would benefit many aspects, including understanding the deep learning models, developing stronger defenses and robust models, and evaluating the vulnerability of the modern DNNs [11].

Various understandings of adversarial transferability have been proposed in the past years and led to effective adversarial attacks. Most works explain such transferability from a model perspective, claiming that the decision boundary [12], model architecture [13], [14], and the test accuracy [15], [16] of the surrogate model have a significant influence on the adversarial transferability. These understandings of adversarial transferability from a model perspective motivate various methods to improve adversarial transferability by investigating models’ properties. Some works introduce data augmentation [17], [18], [19] into the generation of adversarial examples or training generators [20], [21] to perform attacks to reduce the reliance on the decision boundary of the surrogate classifier. Wu et al. [13] propose to modify the architecture of the model to enhance the adversarial transferability and Huang et al. [22] propose to fine-tune the adversarial examples using the mid features of the surrogate model. Though these methods are effective in untargeted scenarios, their performance is highly limited in targeted attack scenarios.
To fully understand adversarial transferability, especially in targeted attack scenarios, we propose a novel perspective from the data distribution. Recall the classical assumption in machine learning that the validation data that are independent and identically distributed with the training dataset can be classified correctly by different models, while the out-of-distribution examples can cause difficulty for models to classify [23], [24]. Our hypothesis is also built on such an assumption. To be specific, we denote the distribution of the training dataset as \( p_D(x|y) \), where \( y \) represents the class label, and \( x \) represents the image. Different models tend to predict the validation data that are identically distributed with \( p_D(x|y) \) as \( y \) and can hardly classify the data that is not identically distributed with \( p_D(x|y) \) as \( y \). Therefore, moving the image out of its original distribution causes difficulties for different models to classify this out-of-distribution example, which can enhance the transferability of the untargeted attack. Dragging the image into the target distribution misleads different models to classify the image as the target class, which can enhance the transferability of the targeted attack.

Towards this end, we propose a method named Distribution-Relevant Attack (DRA) to demonstrate our hypothesis. We attempt to push the input image away from its original distribution to generate transferable adversarial examples. However, as we do not have access to the ground truth data distribution, it is technically challenging to push the images away from its original distribution directly.

We borrow the idea from the score-matching generative models [25], [26], [27], [28], which propose to estimate the gradient of the ground truth data distribution \( \nabla_x \log p_D(x|y) \) and generate the image of the certain distribution iteratively using the estimated gradient of the ground truth data distribution through Langevin dynamics [25], [27]. Previous attacks iteratively minimize (maximize) the conditional density of the model \( p_\theta(y|x) \) along the gradient of the conditional density of the model \( \nabla_x \log p_\theta(y|x) \) to perform the untargeted (targeted) attacks. Thus, to estimate the gradient of the ground truth data distribution in the transfer attack scenarios, we fine-tune the surrogate classifier to match the gradient of the conditional density of the model and the gradient of ground truth data distribution. Thereby, the gradient of the fine-tuned model can approximate the gradient of the ground truth data distribution and the process of generating the adversarial examples with the gradient of our fine-tuned models can approximate the process of the Langevin dynamics, which enables us to manipulate the distribution of the image. We name the attack that uses our fine-tuned models to push the image away from the original distribution while generating the adversarial examples Distribution-Relevant Attack (DRA). What’s more, DRA is compatible with existing transfer attacks and can greatly improve the performance of these attacks.

Visually, targeted adversarial perturbation generated by our method, which can drag the image into the target distribution, reflects vivid semantic features of the target class (See Fig. 1: Turning the tree frog to corn). In Fig. 2, we use the out-of-distribution (OOD) detection method Energy [29] to evaluate that our DRA can indeed move the image out of its original distribution, performing better than the normal PGD attack.

We have conducted extensive evaluations and established state-of-the-art performance in both untargeted and targeted attack scenarios, improving the targeted attack success rate by up to 40% in some cases. This work provides new insight into the understanding of adversarial transferability from a data distribution perspective and provides a strong counterpart for future research on defense.

The main contributions of our paper are summarized as follows:

- We provide a new understanding of adversarial transferability from the perspective of data distribution, advocating that adversarial transferability can be enhanced by pushing the images away from its original distribution.
- We introduce a method to match the gradient of the model and the gradient of the data distribution, which enables us to push the image away from its original distribution using the gradient of the model.
- Extensive experiments demonstrate that our DRA outperforms state-of-the-art approaches a lot in both untargeted and targeted attack scenarios (even up to 40% in most cases).

The rest of the paper is organized as follows. Section II summarizes the literature related to adversarial attacks. In Section III, we firstly present some preliminaries and the motivation of our method. Then we introduce the optimization of the distance between the gradient of the model and the gradient of the data distribution. After that, we propose the Algorithm of our DRA, which fine-tunes the original surrogate
model and use the fine-tuned model to generate adversarial examples. In Section IV, we firstly conduct untargeted transfer attack experiments to demonstrate the superiority of DRA against various target models, including both normal models and secured models. Then, we evaluate the effectiveness of DRA in the targeted attack scenario which is more difficult than the untargeted scenario. Section V provides some discussion to further understand our method. Section VI gives some conclusive results.

II. RELATED WORK

In this section, we briefly review the literature related to adversarial attacks.

Deep neural networks (DNNs) obtained by normal training are vulnerable to adversarial examples [10]. This phenomenon has drawn wide concern and affected the deployment of DNNs in many safety-critical fields, such as face recognition [30], [31], [32], medical diagnosis [33], [34], speaker recognition [35], [36], and autonomous driving [37], [38]. According to the access rights to the target model, adversarial attacks can be classified as white-box attacks and black-box attacks.

White-box attacks assume that the attacker can completely access the structure and parameters of the target model. Typical examples of white-box attacks are FGSM [11], BIM [39], PGD [40], DeepFool [41], JSM [42], and CW [43]. The widely used PGD attack builds an adversarial example by performing multi-step gradient updating along the direction of the gradient at each pixel and projecting the perturbation into the specified range.

Black-box attacks assume that the attacker only knows the output of the target model (prediction or confidence), including query-based attacks and transfer attacks. Query-based attacks perform the black-box attack by estimating gradient with queries to the target model [44], [45], [46], [47]. In this paper, we assume that the attacker can only generate adversarial perturbation using a surrogate model without any queries on the target model. This method is much more efficient and is relatively harder to detect by the target system than query-based attacks. Thus, many existing works focused on leveraging the transferability of adversarial examples.

A. Iterative Methods

Liu et al. [12] show that adversarial attacks can be transferred between different models. Iterative methods attack a surrogate model and update the perturbation iteratively using gradient information [13], [17], [18], [48], [49], [50], [51], [52]. The iterative attack methods such as BIM [39] and PGD [40] could achieve good performance in white-box attack scenarios, but they often suffer from low transferability. Recently, many methods have been proposed to improve the adversarial transferability of the iterative methods.

Some methods suggest stabilizing update directions for the iterative algorithms. Dong et al. [48] propose to improve the adversarial transferability by integrating the momentum of gradients into the update of perturbation. Lin et al. [18] adapt Nesterov accelerated gradient into the update of perturbation to enhance the adversarial transferability. Data augmentation, which plays an important role in improving model generalization and mitigating over-fitting, also contributes to adversarial transferability. Xie et al. [17] suggest applying random transformations (resizing and padding) to the input images at each iteration during attacking. Lin et al. [18] take the scale copies of the input images into attack in order to mitigate overfitting on the surrogate model. Wang et al. [19] propose Admix, which attacks the input image admixed with a group of images randomly sampled from other categories. Wang et al. [50] propose a loss to decrease interactions between perturbation units during attacking. There are also some model-specific methods to improve adversarial transferability. Huang et al. [22] fine-tune the adversarial examples by increasing perturbation on a pre-specified layer. Wu et al. [13] propose to reduce the gradients from the residual modules and pay attention to the architectural vulnerability of DNNs. Guo et al. [51] show that properly increasing the linearity of DNNs can enhance adversarial transferability.

These methods perform well in untargeted attacks, but their performance degrades severely in targeted attacks.

B. Generative Methods

The methods generating the targeted adversarial perturbations by generative models always perform better than iterative methods at the expense of training the same number of generative models as the labels [20], [21], [53], [54], [55]. Poursaeed et al. [54] propose to train the generative model against the surrogate classifier via cross-entropy loss. Naseer et al. [20] show that the relativistic cross-entropy loss can improve the performance of the generative model. Naseer et al. [21] propose to match the ‘distribution’ of perturbed images with that of the target class within latent space of the surrogate classifier in generative training so as to reduce the reliance on class-boundary information from the surrogate classifier. This method can successfully imprint the features of the target distribution to the image and achieves satisfactory targeted attack performance but needs to train generators for every class which is nontrivial on large-scale datasets.

Compared with the existing iterative methods, our DRA pays attention to the distribution-relevant information in the surrogate model rather than improving the iterative algorithm or performing data augmentation. DRA overcomes the low transferability of iterative attacks in targeted attack scenarios. The generative methods aim to learn the distribution of the adversarial perturbation, while our DRA focuses on the ground truth distribution. As we show in our experiments, our DRA greatly improves the adversarial transferability and surpasses existing methods in both untargeted and targeted scenarios.

III. METHOD

A. Preliminary

Given a surrogate classifier \( f_\theta \) parameterized by \( \theta \), and image \( x \), label \( y \), total possible classes \( n \), then \( f_\theta(x)[k] \) represents the \( k^{th} \) output of the last layer.
The conditional density $p_\theta(y|x)$ can be expressed as:
$$p_\theta(y|x) = \frac{\exp(f_\theta(x)[y])}{\sum_{k=1}^{l} \exp(f_\theta(x)[k])}. \tag{1}$$

The adversarial perturbation is usually based on the gradient of the classification loss $\mathcal{L}$. The untargeted attack aims to minimize the conditional density $p_\theta(y|x)$ and can be expressed as [11] and [40]:
$$x' = x + \nabla_x \mathcal{L}(f_\theta(x), y) = x - \nabla_x \log p_\theta(y|x), \tag{2}$$
where $x'$ means the adversarial example of the image $x$. The targeted attack aims to maximize the conditional density $p_\theta(y_{\text{target}}|x)$ and can be expressed as:
$$x' = x - \nabla_x \mathcal{L}(f_\theta(x), y_{\text{target}}) = x + \nabla_x \log p_\theta(y_{\text{target}}|x). \tag{3}$$

The goal of the transfer attack is to mislead the target model using the adversarial examples generated by the surrogate model.

B. Motivation

The existing transfer attacks iteratively minimize $p_\theta(y_{\text{label}}|x)$ (untargeted attack) or maximize $p_\theta(y_{\text{target}}|x)$ (targeted attack) of the surrogate model to generate adversarial examples and then use these adversarial examples to attack the target models. However, the existing transfer attacks can hardly perform targeted attacks successfully and lack the explanation why minimizing $p_\theta(y|x)$ of the surrogate model can also fool the target model with different model parameters and architectures from the surrogate model.

In this paper, we propose to understand and improve the adversarial transferability from a data distribution perspective, which builds on the classical assumption in machine learning methods [23], [24] that the deep models can properly classify the validation data that is independent and identically distributed with the training dataset but can hardly classify the out-of-distribution examples. Specifically, the models tend to predict the label of the image that is identically distributed with $p_D(x|y)$ as $y$, but can not handle the out of distribution images properly. We hypothesize that moving the image out of its original distribution can achieve high untargeted adversarial transferability and dragging the image into the target distribution $p_D(x|y_{\text{target}})$ can achieve high targeted adversarial transferability. The challenge comes from how to push the image away from its original distribution as we don’t have access to the ground truth class-conditional data distribution $p_D(x|y)$.

We borrow the idea from the score-matching generative models [25], [26], [27] which propose to estimate the gradient of the ground truth data distribution and then move the initial image from its original distribution $p_D(x|y_0)$ to the target distribution $p_D(x|y)$ iteratively through the Stochastic Gradient Langevin Dynamics (SGLD) [28], [56], [57]:
$$x_t = x_{t-1} + \alpha \cdot \nabla_{x_{t-1}} \log p_D(x_{t-1}|y) + \sqrt{2\alpha} \cdot \epsilon. \tag{4}$$

The $\epsilon \sim \mathcal{N}(0, I)$ and $\alpha$ is a fixed step size. When $\alpha \to 0$ and $T \to \infty$, $x_T$ is exactly an sample from $p_D(x|y)$. Updating the SGLD process along the opposite direction of $\nabla_x \log p_D(x|y)$ can move the image away from the distribution $p_\theta(y|x)$. Based on the above reasoning, the gradient of the data distribution can be used to manipulate the distribution of the input via iterative methods.

In this paper, we propose to match the gradient of the log conditional density $\nabla_x \log p_\theta(y|x)$ (the direction of the normal adversarial attack) and the gradient of the log ground truth class-conditional data distribution $\nabla_x \log p_D(x|y)$. In this way, the adversarial attack can approximate the direction of the gradient of the ground truth class-conditional data distribution.

To be specific, if $\nabla_x \log p_\theta(y|x)$ matches $\nabla_x \log p_D(x|y)$ well, the untargeted attack $x' = x - \eta \cdot \nabla_x \log p_\theta(y_{\text{label}}|x)$ can be regarded as an approximation of the opposite process of SGLD sampling $x_t = x_{t-1} + \alpha \cdot \nabla_{x_{t-1}} \log p_D(x_{t-1}|y_{\text{label}})$, which moves the images out of its original distribution $p_D(x|y_{\text{label}})$. Fig.2 shows that our untargeted attack indeed moves the image out of its original distribution, which causes difficulties for different models to classify this image.

Similarly, if $\nabla_x \log p_\theta(y|x)$ matches $\nabla_x \log p_D(x|y)$ well, our targeted attack $x' = x + \eta \cdot \nabla_x \log p_\theta(y_{\text{target}}|x)$ can be regarded as an approximation of the process of SGLD sampling $x_t = x_{t-1} + \alpha \cdot \nabla_{x_{t-1}} \log p_D(x_{t-1}|y_{\text{target}})$, which drags the image to the target distribution $p_D(x|y_{\text{target}})$. Fig.1 shows that our method can imprint the features of the target distribution into the image and semantically change the tree-frog to corn, which can mislead the models to classify the image as the target class. Compared with the existing transfer attacks, our method aims to intrinsically manipulate the distribution of the image rather than just minimizing or maximizing the classification loss.

In the next subsection, we provide an appealingly simple and generic technique to match the $\nabla_x \log p_\theta(y|x)$ and $\nabla_x \log p_D(x|y)$. The last subsection instructs how to generate high transferable adversarial examples with our method.

C. Decreasing the Distance Between Gradients

In this section, we propose a novel method to decrease the distance between the gradients, which enables us to use the gradient of the model to estimate the gradient of the ground truth data distribution. In this way, adversarial attack can push the image away from its original distribution through Langevin Dynamics (Eq.(4)).

We define the Distance between the gradient of log Conditional density and the gradient of log Ground truth class-conditional data distribution (DCG) as:
$$\text{DCG} \triangleq \mathbb{E}_{p_D(x|y)} \mathbb{E}_{p_\theta(x|y)} \| \nabla_x \log p_\theta(y|x) - \nabla_x \log p_D(x|y) \|_2^2$$
$$= \int \int \| \nabla_x \log p_\theta(y|x) - \nabla_x \log p_D(x|y) \|_2^2 p_D(x|y) p_D(y) dxdy$$
$$= \int \| \nabla_x \log p_D(x|y) \|_2^2 p_D(x|y) p_D(y) dxdy$$
$$+ \int \| \nabla_x \log p_\theta(y|x) \|_2^2 p_D(x|y) p_D(y) dxdy$$
$$-2 \int (\nabla_x \log p_\theta(y|x) \cdot \nabla_x \log p_D(x|y)) p_D(x|y) p_D(y) dxdy.$$  \tag{5}
We omit the integration domain here for simplicity. The first term is a constant which does not depend on the model’s parameters \( \theta \). The middle term can be expressed as 
\[
E_{p_D(y)}\mathbb{E}_{p_D(x|y)} \| \nabla_x \log p_D(x|y) \|_2^2
\]
is tractable since this term does not contain the unknown score of the ground truth distribution. The last term is not directly computable, because the score of the ground truth distribution \( \nabla_x \log p_D(x|y) \) is unknown. Score matching methods \[26\], \[27\], \[58\] eliminate the score of the ground truth distribution using integration by parts. Inspired by these methods, we apply integration by parts to the last term in Eq.(5) as:
\[
\int_a^b p_D(x)dy \int_{x \in \mathbb{R}^n} (\nabla_x \log p_D(x|y))p_D(x|y)dx \\
\left[ I \right]
\int_a^b p_D(x)dy \int_{x \in \mathbb{R}^n} \nabla_x \log p_D(x|y) \cdot \nabla_x p_D(x|y)dx \\
\left[ II \right]
\int_a^b p_D(x)dy \sum_{i=1}^n \int_{x \in \mathbb{R}^n} \nabla_{x_i} \log p_D(x|y) \nabla_{x_i} p_D(x|y)dx \\
\left[ III \right]
\int_a^b p_D(x)dy \sum_{i=1}^n \int_{x \in \mathbb{R}^n} \nabla_{x_i} \log p_D(x|y) \theta |_{\nabla_{x_i} p_D(x|y)}dx_i \\
\left[ IV \right]
\int_a^b p_D(x)dy \sum_{i=1}^n \int_{x \in \mathbb{R}^n} \nabla_{x_i} \log p_D(x|y) \theta |_{\nabla_{x_i} p_D(x|y)}dx_i \\
\end{equation}

where \( \nabla_x \) denotes the Hessian with respect to \( x \). We use \( a \) to represent \(-\infty \) and \( b \) to represent \( +\infty \). \( \nabla_{x_i} \theta \) represents the vector \([x_1, \ldots, x_{i-1}, +M, x_{i+1}, \ldots, x_n] \). \( \nabla_{x_i} p_D(x|y) \) represents the vector \([x_1, \ldots, x_{i-1}, -M, x_{i+1}, \ldots, x_n] \) for equality (I). In equality (I), \( \nabla_x \log p_D(x|y) \theta \) and \( \nabla_x \log p_D(x|y) \) are n-dimensional vectors, and their product result is a scalar. We use the formula: \( u^T \cdot v = \sum_{i=1}^n u_i v_i \) for equality (II), where \( n \) represents the dimension of the data. As for equality (III), we use the integration by parts formula (See Appendix for proof):
\[
\int_{-\infty}^{+\infty} \nabla_{x_i} f(x) \nabla_{x_i} g(x) dx_i = \lim_{M \to \infty} g(x) \nabla_{x_i} f(x) |_{-M}^{+M} - \int_{-\infty}^{+\infty} g(x) \nabla^2_{x_i} f(x) dx_i .
\]

The equality (IV) holds for that we assume \( p_D(x|y) \to 0 \) when \( \|x\|_2 \to \infty \).

Thus, substituting the results of integration by parts into Eq.(5), the DCG loss can be reformulated as:
\[
L_{\text{DCG}} \triangleq E_{p_D(y)}(E_{p_D(x|y)}[\nabla_x \log p_D(x|y) - \nabla_x \log p_D(x|y)]^2_2 \\
+ 2 \cdot E_{p_D(x|y)}[\nabla_x \log p_D(x|y)] \|_{\nabla_x}^2 \log p_D(x|y)] + \text{const.} \quad (8)
\]

We ignore the const in Eq.(8) that does not depend on the model parameters and denote the DCG loss as \( L_{\text{DCG}} \):
\[
L_{\text{DCG}} \triangleq E_{p_D(y)}(E_{p_D(x|y)}[\nabla_x \log p_D(x|y)]^2_2 \\
+ 2 \cdot E_{p_D(x|y)}[\nabla_x \log p_D(x|y)] \|_{\nabla_x}^2 \log p_D(x|y)] . \quad (9)
\]

Computing the Hessian trace term in Eq.9 requires a number of backpropagations that is proportional to the data dimension, which is intractable for high-dimensional data. Hutchinson’s trick \[59\] is a stochastic algorithm to approximate \( \text{tr}(A) \) for any square matrix \( A \). For a distribution of a random vector \( v \) such that \( E_{p(v)}[v^T] = I \), Hutchinson’s trick approximate \( \text{tr}(A) \) as: \( \text{tr}(A) = E_{p(v)}[v^T A v]. \) Hence, using Hutchinson’s trick, we can replace \( \text{tr}(\nabla_x^2 \log p_D(x|y)) \) with \( E_{p_D(x|y)}[v^T \nabla_x^2 \log p_D(x|y)] v \). Thus, we can reformulate the \( L_{\text{DCG}} \) as:
\[
L_{\text{DCG}} \triangleq E_{p_D(y)}(E_{p_D(x|y)}[\nabla_x \log p_D(x|y)]^2_2 \\
+ 2 \cdot E_{p_D(x|y)}[\nabla_x \log p_D(x|y)] v^T \nabla_x^2 \log p_D(x|y)] v . \quad (10)
\]

In practice, we can tune the number of samples \( v \) to trade off the performance of estimation and computational cost. With reference to the existing methods \[25\], \[26\], we sample one random vector \( v \) independently for each input during the training process. The first term in Eq.(10) can be computed by one backpropagation. The second term involves Hessian, but it is in the form of Hessian-vector products, which can be computed within \( O(1) \) backpropagations. Therefore, the computation of Eq.(10) does not depend on the dimension of data and is scalable for training deep models on high-dimensional datasets.

We propose fine-tuning the surrogate model by optimizing the classification loss and the DCG loss jointly during training. The optimization objective can be formulated as:
\[
\min_\theta [L(f_\theta(x), y) + \lambda \cdot L_{\text{DCG}}],
\]

where \( \lambda \) represents the regularization strength.

In this way, we can obtain a fine-tuned surrogate model whose gradient of log conditional density \( \nabla_{x} \log p_D(x|y) \) aligns with the gradient of log ground truth class-conditional
Algorithm 1 Distribution-Relevant Attack DRA: Given Network $f_0$ Parameterized by $\theta$, Regularization Constant $\lambda$, Epochs $T$, Total Batches $M$, Learning Rate $\eta$, the Classification Loss $\mathcal{L}$, the DCG Loss $\mathcal{L}_{DCG}$. Adversarial Perturbation $\delta$, Original Image $x$, $\ell_\infty$ Perturbation Radius $\epsilon$; Step Size $\alpha$; Iterations $N$.

1. **Fine-tuning:**
   
   for $i = 1, 2, ..., T$ do
   
   for $j = 1, 2, ..., M$ do
   
   Updating model parameters:
   
   $\theta = \theta - \eta \cdot (\nabla_{\theta} \mathcal{L}(f_\theta(x_j), y_j) + \lambda \cdot \nabla_{\theta} \mathcal{L}_{DCG})$
   
   end for
   
end for

return Fine-tuned network $f_\theta$

2. **Untargeted Attack:**
   
   Initialize $\delta = \text{Uniform}(-\epsilon, \epsilon)$.

for $i = 1, 2, ..., N$ do

$\delta = \delta + \alpha \cdot \text{sign}(\nabla_{x} \mathcal{L}(f_\theta(x + \delta), y_{\text{label}}))$.

$\delta = \max(\min(\delta, \epsilon), -\epsilon)$

end for

return $\delta$

3. **Targeted Attack:**
   
   Initialize $\delta = \text{Uniform}(-\epsilon, \epsilon)$.

for $i = 1, 2, ..., N$ do

$\delta = \delta - \alpha \cdot \text{sign}(\nabla_{x} \mathcal{L}(f_\theta(x + \delta), y_{\text{target}}))$.

$\delta = \max(\min(\delta, \epsilon), -\epsilon)$

end for

return $\delta$

data distribution $\nabla_{x} \log p_D(x|y)$ better than the original surrogate model. Moreover, we can manipulate the distribution information of the image through the iterative adversarial attack with the fine-tuned model.

**D. Distribution-Relevant Attack**

We named the attack using our distribution-relevant fine-tuned surrogate models as Distribution-Relevant Attack (DRA). **DRA** consists of two steps: fine-tuning the surrogate model to decrease the distance between the gradient of the model and the gradient of the ground truth data distribution, and using the fine-tuned surrogate model to generate adversarial perturbation with the guidance of the approximate gradient of the ground truth data distribution. Alg. 1 details our method.

We jointly optimize the DCG loss $\mathcal{L}_{DCG}$ and the classification loss $\mathcal{L}$ to fine-tune the surrogate model. This optimization process encourages the direction of the gradient of the surrogate model to match the direction of the gradient of the ground truth data distribution.

Our proposed fine-tuned method aims to enable the attackers to push the image away from its original distribution using the gradient of the model. With the fine-tuned surrogate model, we can use most existing transfer attack methods to conduct attacks. We mainly choose the widely used iterative attack method, projected gradient descent (PGD) [39], [40] to generate adversarial examples. Our fine-tuning method is also compatible with other advanced transfer attacks.

1) **Untargeted Attack:** The untargeted attack can be formulated as:

$$
\begin{align*}
    x_n &= x_{n-1} - \eta \cdot \text{sign}(\nabla_{x_{n-1}} \mathcal{L}(f_\theta(x_{n-1}), y_{\text{label}})), \\
    x_n &= \text{clip}(x_n, x_0 - \epsilon, x_0 + \epsilon),
\end{align*}
$$

where $x_0$ is the clean image, $\mathcal{L}$ is the classification loss, $\eta$ is the perturbation step size, and $y_{\text{label}}$ is the original label for the clean image. The **clip** operation aims to make the perturbation bounded in the budget $\epsilon$.

Although the **DRA** and other iterative methods are similar in expression when generating untargeted attacks, the direction of the **DRA** attack is better aligned with the gradient of the data distribution. In other words, **DRA** moves the image out of its original distribution to generate untargeted attacks, while the other iterative methods pay attention to dragging the inputs across the decision boundary of the classifier. As shown in Fig.2, the adversarial examples generated by our **DRA** are regarded as the out-of-distribution examples. Fig.3 shows the difference between the adversarial examples generated by **DRA** and PGD. The adversarial perturbation of the untargeted attack generated by **DRA** concentrates on the semantic features of the image while the adversarial perturbation generated by the PGD attack seems irregular.

2) **Targeted Attack:** Similar to the untargeted **DRA**, the targeted version of **DRA** can be formulated as:

$$
\begin{align*}
    x_n &= x_{n-1} - \eta \cdot \text{sign}(\nabla_{x_{n-1}} \mathcal{L}(f_\theta(x_{n-1}), y_{\text{target}})), \\
    x_n &= \text{clip}(x_n, x_0 - \epsilon, x_0 + \epsilon),
\end{align*}
$$

where attackers aim to make the DNNs misclassify the input as the target class $y_{\text{target}}$. We generate the targeted attack
starting from the mean image¹ and show the perturbation in Fig. 4. The perturbation generated by DRA represents recognizable features of the target distribution, while the perturbation generated by the PGD attack does not show obvious features. The targeted adversarial perturbation, which contains the sufficient features of the target category, may dominate the classification and the original features act like noise with respect to perturbations.

Compared with the prior research on adversarial transferability by improving the attack optimization procedure, we focus on the ground truth data distribution and match the gradient of the surrogate model to the gradient of the data distribution. We use the fine-tuned surrogate model to generate the iterative attack. In the following section, we try to validate the effectiveness of the proposed DRA in both untargeted and targeted attack scenarios quantitatively.

IV. EXPERIMENT

A. Implementation

DRA consists of two steps: fine-tuning the surrogate model to decrease the distance between its gradient and the gradient of the ground truth data distribution and then using the fine-tuned surrogate model to generate adversarial perturbation. We mainly choose the widely used PGD [40] attack to generate perturbation in our DRA in experiments. We also evaluate the compatibility of our method with the existing advanced transfer attacks in subsection D. All experiments in this paper are run on Tesla V100.

1) Fine-Tuning Details: We fine-tune the pre-trained classifiers provided by PyTorch (version 1.8.0) with the SGD optimizer for 20 epochs. The learning rate is 0.001 and decays by a factor of 10 at epochs 10. The size of mini-batch is 32. We set the hyperparameter λ = 6 in our method. We fine-tune the pre-trained classifiers on the training dataset of the ImageNet which is also used to train these classifiers by PyTorch to avoid data leakage problems. The training images are randomly cropped to $3 \times 224 \times 224$. The computation cost of fine-tuning the surrogate model for one epoch requires 8 hours on Tesla V100 using ResNet-50 on ImageNet.

2) Attack Setting: For untargeted attack scenarios, we choose the baseline attack PGD [40] and 7 state-of-the-art transfer attacks: MI attack [48], DI attack [17], TI attack [49], ILA attack [22], SGM attack [13] and IR attack [50]. These methods achieve high adversarial transferability in the untargeted attack scenario, but their performance drops severely in the targeted attack scenario. We consider the state-of-the-art methods specially designed for the targeted attack in targeted attack scenarios, one of which is the generative method TTP [21], and the other is the iteration method Simple [52]. We mainly evaluate these attacks on the randomly selected 5000 ImageNet [60] validation images that are correctly classified by all source models. We also evaluate the performance of our method on ImageNet V2 [61] and CIFAR-10 in Sec.IV-F.

3) Threat Model: We firstly generate adversarial examples using the surrogate models and then use these adversarial examples to attack different target models. As for the attack strength, we follow the standard attack setting for all attack methods [13], [17]. We set the maximum allowable adversarial perturbation as $\epsilon = 16/255$ with respect to a pixel value in $[0, 1]$ by default. In untargeted attack scenarios, we set the step size $\alpha$ to $2/255$ and set the iteration steps to $N = 10$. With reference to [52], the targeted attack needs more iterations to achieve convergence and we set the iteration steps to $N = 300$. The targeted attack success rate results are averaged on 10 different target classes [21].

4) Target Models and Surrogate Models: We conduct experiments on both normal target models and secured target models. For normal target models, we choose 12 convolutional neural networks (CNNs): VGG16 (with batch normalization), VGG19 (with batch normalization) [62], ResNet-50 (RN50), ResNet-152 (RN152) [63], DenseNet-121 (DN121), DenseNet-201 (DN201) [64], 154 layers Squeeze-and-Excitation network (SE154) [65], Wide_ResNet_50_2 (WRN50-2) [66], SqueezeNet1.0 (SQN) [67], ShuffleNet_v2_x1_0 (SFN) [68], Inception V3 (IncV3) [69], Inception V4 (IncV4), and Inception-ResNet V2 (IncRes) [70], and we use the pretrained models in PyTorch [71]. We also choose the officially released ViT-B/16 [72] as the target model. We consider 4 adversarially trained models, including adversarial training with $\ell_2$ perturbation and $\ell_\infty$ perturbation [73], and 3 other robust training methods, including training with Styled ImageNet (SIN) [74], Augmix [75], the mixture of Styled and natural ImageNet (SIN-IN), as secured methods. We choose 5 models as surrogate models: DNNs with skip connection (ResNet-50, ResNet-152), DNNs with dense connection (DenseNet-121, DenseNet-201) and DNNs without skip connection (VGG19).

B. The Evaluation of Untargeted Attack

In this section, we focus on untargeted attacks. We first conduct experiments to compare our DRA with other baseline methods and show the results in Tab. I. For transfer ResNet-50 $\rightarrow$ VGG19, DRA achieves a success rate of 98.26%, which is 45.26% and 15.54% higher than PGD and SGM, respectively. For transfer ResNet-50 $\rightarrow$ IncRes, DRA achieves a success rate of 92.56%, which is 74.96% and 45.78% higher than PGD and IR, respectively. Moreover, our

Fig. 4. Targeted perturbation (unbounded perturbation for visibility) generated from the mean image. The perturbation generated by our DRA can reflect the semantic features of the target category while the perturbation generated by the PGD attack seems more noisy.
TABLE I
TRANSFERABILITY AGAINST NORMAL MODELS: THE SUCCESS RATES OF BLACK-BOX ATTACKS (UNTARGETED) CRAFTED ON RN50, RN152, DN121 AND DN201. THE MAXIMUM STANDARD DEVIATION OF THIS EXPERIMENT IS 0.92% WHICH IS MUCH LESS THAN OUR IMPROVEMENT. THE BEST RESULTS ARE IN BOLD

| Model | Attack | VGG19 | RN50 | RN152 | DN121 | DN201 | SE154 | IncV3 | IncV4 | IncRes | VIT |
|-------|--------|-------|------|-------|-------|-------|-------|-------|-------|--------|-----|
| PGD [40] | 53.00% | 61.26% | 53.62% | 53.56% | 24.78% | 23.86% | 21.96% | 17.60% | 3.82% |
| DI [17] | 75.06% | 81.65% | 81.98% | 74.80% | 52.42% | 42.58% | 44.30% | 27.12% | 7.12% |
| MI [48] | 64.86% | 73.22% | 73.50% | 64.33% | 47.20% | 39.08% | 37.26% | 32.25% | 13.34% |
| ILA [22] | 83.56% | 92.46% | 88.40% | 85.24% | 61.44% | 49.94% | 48.34% | 35.74% | 11.26% |
| SGM [13] | 82.72% | 88.40% | 83.56% | 80.34% | 61.30% | 53.72% | 49.83% | 42.86% | 18.82% |
| IR [50] | 82.46% | 85.24% | 84.35% | 82.10% | 64.20% | 54.60% | 51.05% | 46.78% | 17.76% |
| DRA+PGD | 98.26% | 99.24% | 99.56% | 99.28% | 93.92% | 95.56% | 92.66% | 92.56% | 58.46% |

| Model | Attack | VGG19 | RN50 | RN152 | DN121 | DN201 | SE154 | IncV3 | IncV4 | IncRes | VIT |
|-------|--------|-------|------|-------|-------|-------|-------|-------|-------|--------|-----|
| PGD [40] | 49.32% | 72.72% | 53.44% | 51.00% | 26.32% | 23.50% | 22.58% | 18.72% | 5.10% |
| DI [17] | 74.01% | 88.18% | 79.46% | 77.81% | 57.49% | 50.28% | 47.16% | 35.10% | 10.40% |
| MI [48] | 65.42% | 83.40% | 77.60% | 75.79% | 53.00% | 46.50% | 43.32% | 33.08% | 15.28% |
| ILA [22] | 66.20% | 90.44% | 75.48% | 73.80% | 50.32% | 42.32% | 41.30% | 29.98% | 12.26% |
| SGM [13] | 80.40% | 96.10% | 85.80% | 82.76% | 61.90% | 53.16% | 49.24% | 43.30% | 11.72% |
| IR [50] | 73.20% | 92.70% | 83.43% | 80.60% | 64.00% | 53.60% | 50.30% | 48.00% | 10.24% |
| DRA+PGD | 96.36% | 99.62% | 99.28% | 98.92% | 92.98% | 95.00% | 91.00% | 91.52% | 52.08% |

| Model | Attack | VGG19 | RN50 | RN152 | DN121 | DN201 | SE154 | IncV3 | IncV4 | IncRes | VIT |
|-------|--------|-------|------|-------|-------|-------|-------|-------|-------|--------|-----|
| PGD [40] | 56.78% | 63.22% | 52.76% | 71.98% | 31.46% | 24.92% | 26.82% | 20.64% | 4.62% |
| DI [17] | 73.68% | 79.56% | 74.72% | 89.40% | 53.34% | 53.65% | 47.94% | 37.72% | 7.54% |
| MI [48] | 68.36% | 74.18% | 72.88% | 89.56% | 58.38% | 52.22% | 45.35% | 35.24% | 14.84% |
| ILA [22] | 87.76% | 90.38% | 83.42% | 95.52% | 65.02% | 58.64% | 57.36% | 40.76% | 9.60% |
| SGM [13] | 80.18% | 88.54% | 80.54% | 92.70% | 64.92% | 54.62% | 49.82% | 37.76% | 12.80% |
| IR [50] | 82.56% | 86.14% | 85.20% | 95.30% | 72.20% | 62.22% | 62.10% | 56.00% | 11.38% |
| DRA+PGD | 98.32% | 99.46% | 98.78% | 99.22% | 94.80% | 95.52% | 93.60% | 92.24% | 58.04% |

The existing works always evaluate the adversarial transferability with the perturbation restricted by $\ell_\infty = 16/255$.

The proposed method can not only improve the adversarial transferability from the convolutional neural network (CNN) to CNN but also improve the adversarial transferability from CNN to the vision transformer. When transferring from ResNet-50 to the ViT-B/16 [72], our DRA achieves the success rate of 58.46% which is 39.64% better than the previous best method SGM and 54.64% better than the baseline method PGD. DRA outperforms existing methods by a large margin in all transfer scenarios. We think that generating adversarial perturbation from the perspective of data distribution is a key factor in the success of DRA. As shown in Fig.2 and Fig.3, DRA corrupts the features of the original distribution and moves the data out of its original distribution. Thus, different models cannot give correct predictions.

As shown in Tab. I, the performance of the existing methods is unsatisfactory in some cases. Some works show that the ensemble-based strategy [12] can improve the performance of transfer attacks [13], [50]. Fig. 5 shows the results of the ensemble strategy. We use the ensemble of ResNet-50, ResNet-152 and DenseNet-121 to generate the adversarial attack. The ensemble strategy can enhance the adversarial transferability of different methods, and our DRA still performs the best among the existing methods.

In Fig. 5, we compare our DRA with the existing methods with different attack strengths. DRA achieves the best performance with different attack strengths.

In Tab. II, we provide experiments to show that the adversarial examples generated by DRA can also effectively penetrate the advanced secured models. Augmix [75] is an augmentation-based training in order to make the model robust to natural corruptions. Training on Stylized ImageNet [74] can increase shape bias and decrease bias towards texture.
Adversarial training is considered the most effective way to defend against attacks [73]. As shown in Tab. II, we generate the adversarial perturbation by different methods and then transfer the perturbation to ResNet-50 trained using different methods. The horizontal axis represents different attack strengths. Our DRA surpasses the other methods against different target models.

### C. The Evaluation of Targeted Attack

In this section, we focus on targeted attacks. Generating transferable targeted adversarial examples is much more challenging for current compared with untargeted attacks [12], [76]. Liu et al. [12] show that the decision boundaries for the original class of the image of different models align well with each other, which partially explains why untargeted adversarial perturbation can transfer. However, the decision regions for the other classes of different models are very different, which causes the difficulty of obtaining high targeted adversarial transferability by attacking the surrogate model. It is worth noting that different models tend to give the same predictions for images from the same distribution. As shown in Fig. 4, the targeted perturbation generated by DRA contains the features of the target distribution, which may dominate the prediction.

Tab. III shows that our DRA can surpass the previous best targeted transfer attack TTP [21] by a large margin. For transfer ResNet-50 $\rightarrow$ VGG19, DRA achieves a success rate of 87.80%, which is 17.03% and 6.69% higher than Simple [52] and TTP, respectively. When transferring from the surrogate model to the target model with quite different architecture, the advantages of DRA are more significant. For transfer ResNet-50 $\rightarrow$ IncV3, DRA achieves a success rate of 75.93%, which is 60.43% and 29.46% higher than Simple and TTP, respectively. Meanwhile, we found that Simple [52] performs best when the surrogate and target models have the same architecture. When the architecture of the surrogate and target model is the same, Simple [52] uses the same model to generate perturbation and evaluates the attack performance, which is a white-box attack. Our DRA uses the modified model to generate adversarial perturbation and the target model shares the same architecture with the surrogate model but has different parameters. Thus the attack success rate of DRA is slightly lower than Simple in the white box attack scenario. However, since we focus more on the adversarial transferability between models with different architectures, we argue that this is not a conspicuous drawback of the proposed method.

To evaluate the effectiveness of our DRA comprehensively, we also evaluate its transferability against secured models. Similar to the Sec. IV-B, we consider various types of defense methods (augmented vs. stylized vs. adversarial).

### D. The Compatibility of DRA With Other Attacks

Various techniques have been proposed to improve the transferability of adversarial attacks, such as advanced gradient calculations [13], [18], [48], and input transformations [17], [18]. In the previous subsections, we mainly show that our DRA method can significantly improve the performance of the widely used baseline attack PGD [40]. Here, we delve into the compatibility of our DRA method with other attack methods. In Tab. V, we compare the adversarial transferability of the original adversarial attacks and the DRA version of these attacks. “PGD” means the PGD attack generated with the normal pre-trained surrogate models and “PGD+DRA” means the PGD attack generated by our DRA fine-tuned surrogate models. Our DRA fine-tuning method can significantly improve the performance of different baseline attacks. Moreover, incorporating DRA with the input transformations based attacks (SI, DI) achieves better performance than the
advanced gradient based attacks (MI, NI, SGM). We think the advanced gradient based attacks somewhat change the gradient of the model which may increase the DCG in our fine-tuned models.

E. Transfer-Based Attack on Google Cloud Vision

In this section, we evaluate our DRA attack in the more challenging case, attacking a real-world computer vision system (the Google Cloud Vision API). Most existing works fool real-world computer vision systems with query-based attacks, which require a large number of queries [46], [77], [78]. In contrast, we apply the transfer-based attack to fool the Google Cloud Vision API. Specifically, we use the ResNet-50 as the surrogate model to generate the adversarial examples and then use the Google Cloud Vision API to predict these examples. The API predicts a list of semantic labels along with confidence scores. We measure both the targeted and untargeted transferability. For the untargeted attack, we measure whether or not the ground-truth class appeared in the returned list. For the targeted attack, we measure whether or not the target class appeared in the returned list. Since the predicted label space of Google Cloud Vision API does not precisely correspond to the 1000 ImageNet classes, we treated semantically similar classes as the same, following the setting in [52]. We take the evaluation on randomly selected 500 images that originally yield correct predictions. Fig.7 shows the targeted adversarial examples generated by our method and the top-5 predictions made by the Google Cloud Vision API, the hamster and the arctic fox are misclassified as corn by the Google Cloud Vision API. We compare our method with the original PGD and the previous best iterative transfer attack method Simple [52] in Tab.VI. In particular, DRA achieves the best attack performance compared with previous attack methods. This demonstrates the high practicality of our method which can even attack real-world computer vision systems with a high success rate, e.g., Google Cloud Vision API.

F. Evaluation on the Other Datasets

1) The Evaluation on ImageNet-V2: In this subsection, we evaluate our method on the ImageNet-V2 with

| Model | Attack | VGG19 | DN121 | RN50 | RN152 | WRN50-2 | SQN | SFN | IncV3 |
|-------|--------|-------|-------|------|-------|---------|-----|-----|-------|
| RN50  | DRA+PGD| 92.36 | 89.84 | 83.63 | 85.68 | 75.48   |     |     |       |
| DN121 | DRA+PGD| 93.82 | 86.55 | 84.45 | 80.48 | 75.25   | 87.25 | 82.62 | 70.58 |

TABLE III
Transferability Against Normal Models: The Success Rates of Black-Box Attacks (Targeted) Crafted on VGG19, RN50 and DN121. The Maximum Standard Deviation of This Experiment Is 0.88% Which Is Much Less Than Our Improvement. The Best Results Are in Bold

| Model | Attack | VGG19 | DN121 | RN50 | RN152 | WRN50-2 | SQN | SFN | IncV3 |
|-------|--------|-------|-------|------|-------|---------|-----|-----|-------|
| RN50  | DRA+PGD| 92.36 | 89.84 | 83.63 | 85.68 | 75.48   |     |     |       |
| DN121 | DRA+PGD| 93.82 | 86.55 | 84.45 | 80.48 | 75.25   | 87.25 | 82.62 | 70.58 |

TABLE IV
Transferability Against Secured Models. We Generate Adversarial Perturbation (Targeted) by Different Methods, and Then Transfer the Perturbation to ResNet-50 Trained Using Different Secured Methods Including AUGMIX [75], Stylized ImageNet [74] and Adversarial Examples [73]. The Best Results Are in Bold

| Attack | MN | MN-IN | Augmix | λs=0.05 | λs=0.1 | λs=0.5 | λs=1 |
|--------|---|-------|--------|--------|--------|--------|------|
| Simple | 0.78 | 4.60 | 2.54 | 0.03 | 0.22 | 0.23 | 0.19 |
| TTP | 67.32 | 61.62 | 77.52 | 68.42 | 7.14 | 10.34 | 0.56 |
| DRA+PGD | 80.00 | 89.31 | 79.73 | 82.23 | 41.88 | 48.45 | 15.54 |
| DN121 | 50.48 | 77.32 | 78.86 | 69.08 | 7.08 | 12.78 | 0.65 |
| Simple | 5.27 | 17.78 | 20.37 | 5.70 | 0.31 | 0.33 | 0.24 |
| TTP | 58.83 | 92.92 | 86.74 | 90.22 | 52.52 | 58.43 | 18.01 |
| DRA+PGD | 88.62 | 97.41 | 92.21 | 94.91 | 59.42 | 67.00 | 24.22 |

Fig. 7. We show that the adversarial perturbation generated by our DRA is imperceptible to human observers but can successfully fool the Google Cloud Vision with the target class.
### TABLE V

| Attack Success Rates of Untargeted Black-Box Attacks on ImageNet Crafted on RN50 and DN121. The Maximum Standard Deviation of This Experiment Is 0.96% Which Is Much Less Than Our Improvement. The Best Results Are in Bold |
|---|---|---|---|---|---|---|---|
| Source | Attack | Vgg19 | RN152 | DN121 | DN201 | SB154 | IncV3 | IncV4 | IncRes |
| PGD [40] | 55.00 | 61.26 | 55.62 | 53.36 | 24.78 | 20.86 | 21.96 | 17.60 |
| PGD+DRA | 98.26 | **99.24** | **99.56** | **99.28** | **93.92** | **95.56** | **92.66** | **92.56** |
| MI [48] | 64.86 | 73.22 | 73.30 | 64.33 | 47.20 | 39.08 | 37.35 | 25.26 |
| MI+DRA | 96.36 | 97.88 | 98.68 | 92.28 | 91.26 | 94.06 | 91.94 | 91.78 |
| NT [18] | 79.20 | 84.20 | 81.36 | 78.20 | 55.62 | 47.96 | 50.14 | 39.36 |
| NH+DRA | 96.52 | 98.34 | 99.00 | 98.44 | 91.76 | 93.30 | 90.04 | 90.14 |
| SGM [13] | 82.87 | 88.40 | 83.56 | 80.34 | 61.30 | 53.72 | 49.83 | 42.86 |
| SGM+DRA | 97.52 | 98.58 | 99.00 | 98.40 | 93.04 | 94.56 | 91.42 | 90.42 |
| SI [18] | 75.22 | **86.46** | **83.56** | **79.30** | **64.66** | **42.68** | **41.68** | **31.30** |
| SI+DRA | 96.80 | 98.88 | 99.24 | 98.82 | 92.62 | 95.62 | 92.94 | 93.82 |
| DI [17] | 75.06 | **81.65** | **81.98** | **74.80** | **52.42** | **42.58** | **44.30** | **27.12** |
| DI+DRA | **98.88** | 98.94 | 99.34 | 99.06 | **96.28** | **96.80** | **95.02** | **94.02** |

### TABLE VI

| Untargeted and Targeted Transfer Attack Success Rates (%) of Different Attacks on Google Cloud Vision |
|---|---|---|---|---|---|---|---|
| Source | Attack | Vgg19 | RN50 | RN152 | DN201 | SE154 | IncV3 | IncV4 | IncRes |
| PGD [40] | 56.78 | 58.22 | 43.76 | 65.98 | 23.46 | 21.92 | 23.82 | 17.64 |
| PGD+DRA | 98.32 | **99.46** | **98.78** | **99.22** | **94.80** | **95.52** | **93.60** | **92.24** |
| MI [48] | 68.36 | 74.18 | 67.88 | 79.56 | 55.58 | 49.22 | 42.35 | 32.24 |
| MI+DRA | 96.60 | 98.44 | 97.28 | 98.38 | 93.20 | 93.81 | 91.88 | 89.90 |
| NT [18] | 83.72 | **86.08** | **74.98** | **89.84** | **62.36** | **54.66** | **56.90** | **43.78** |
| NH+DRA | 96.98 | 98.88 | 97.96 | 98.78 | 92.62 | 92.78 | 90.40 | 88.18 |
| SGM [13] | 80.18 | **88.54** | **80.54** | **92.70** | **64.92** | **54.62** | **49.82** | **37.76** |
| SGM+DRA | 92.16 | 94.76 | 90.98 | 93.04 | 82.60 | 79.58 | 76.18 | 70.22 |
| SI [18] | 76.58 | **80.90** | **69.08** | **88.64** | **48.68** | **46.08** | **45.24** | **33.52** |
| SI+DRA | 97.64 | 99.28 | 98.66 | **99.34** | **94.74** | **95.90** | **94.06** | **93.68** |
| DI [17] | 73.68 | 79.56 | 71.72 | 80.40 | 53.34 | 49.65 | 43.94 | 31.72 |
| DI+DRA | **98.94** | 99.34 | 98.42 | 99.18 | **95.90** | **95.76** | **94.78** | **92.16** |

10000 images [61] which are independent of existing pre-trained models. As shown in Tab.VII, our DRA method surpasses the other methods by a large margin. As for transfer from ResNet-50 to Inception-V3, DRA can improve the performance of the baseline method PGD from 59.47% to 99.24%.

2) The Evaluation on CIFAR-10: Following the existing works [13], [50], [52], we focus on addressing the transferability on the ImageNet dataset in the previous subsection. In this subsection, we conduct experiments on another dataset (CIFAR-10) to verify the effectiveness of the DRA algorithm further. We use the ResNet-18 as the surrogate model and choose three target models with different architecture (VGG19, DenseNet-121 and ShuffleNetV2). We set the step size $\alpha$ to $2/255$, and set the iteration steps to $N=10$. As shown in Fig. 8, our DRA shows better transferability than existing transfer attacks. For transfer ResNet-18 → VGG19 with attack strength $\epsilon = 4/255$, DRA can achieve a success rate of 61.97% which is 13.62% higher than SGM [13] and 18.48% higher than PGD [40]. So far, we have evaluated the effectiveness of our DRA on the large-scale datasets (ImageNet) and small-scale datasets (CIFAR-10).

---

**V. DISCUSSION**

**A. Understanding the Superiority of DRA**

The superiority of DRA can be understood from two aspects. First, DRA reduces the dependency on the surrogate model. The existing transfer attacks usually regard the overfitting on the surrogate models as the hindering factor of adversarial transferability and devote to alleviating the overfitting by improving the optimization algorithm. Our proposed method seeks the commonality among different models from the data distribution perspective for that the ground truth data distribution is model-independent. Our method alleviates this over-fitting by aligning the gradient of the model with the gradient of the ground-truth data distribution. In this way, DRA can effectively reduce the dependency on the surrogate model and generate high transferable adversarial examples. Fig.9 illustrates the frequency histograms of Pearson Correlation Coefficient (PCC) [79] between the adversarial perturbations...
Fig. 9. Frequency histogram of the Pearson correlation coefficient (PCC) between adversarial perturbations of the same image generated by the surrogate models ResNet-50 and ResNet-152 (a), ResNet-50 and DenseNet-121 (b), and DenseNet-121 and DenseNet-201 (c). Higher PCC indicates a greater positive correlation. The perturbations generated by our DRA method are closer to each other with different surrogate models than perturbations generated by other methods.

B. Compared With Other Fine-Tuning Methods

Our fine-tuning method aims to match the direction of the adversarial attack to the gradient of the ground truth data distribution, which is different from the other fine-tuning methods that aim to improve the generalization of models. In Tab. VIII, we compare the test accuracy and the adversarial transferability of different ResNet-50 models. “AugMix” means the ResNet-50 fine-tuned with the data processing technique AugMix [75], “AutoAug” means the ResNet-50 fine-tuned with fast AutoAugment [81] and “MEALV2” means the ResNet-50 fine-tuned with the knowledge distillation method MEALV2 [82] that achieves 80%+ Top-1 accuracy on ResNet-50. Our method reduces the test accuracy of the surrogate model but greatly enhances the adversarial transferability. This experiment shows that the generalization (test accuracy) of the model may not be the key factor for adversarial transferability.

C. The Influence of the Hyperparameter

In this subsection, we show that decreasing the distance between the gradient of the surrogate model and the gradient of the ground truth data distribution enhances adversarial transferability, whereas increasing this distance has the opposite effect. As shown in Eq. 11 and our Alg. 1, our proposed method DRA has one hyperparameter \( \lambda \). We optimize the classification loss and the DCG loss jointly during fine-tuning and adjust the strength of DCG loss through the hyperparameter \( \lambda \). Fig. 10 shows the influence of \( \lambda \). If \( \lambda \) is less than 0, the distance between the input-gradients of the surrogate model and the gradients of the ground truth data distribution is increased, and the adversarial transferability
This further validates the superiority of our proposed adversarial images, compared with PGD [40] and SGM [13].

As shown in Fig. 11, though the final latent representations of the adversarial examples generated by the DenseNet-121 (target model) to obtain the final latent model) to generate untargeted adversarial examples and use t-SNE [83]. To be specific, we use the ResNet-50 (surrogate model) to fine-tune models. “λ = 1.0” represents the model fine-tuned with hyperparameter λ = 1. We also present a seagull’s adversarial perturbation generated by different (c) DRA (Ours).

D. t-SNE Visualizations

To find out whether the generated adversarial examples have attacked the target model in the desired way. We visualize the feature embeddings of both the clean images (red) and their corresponding adversarial examples (green) using t-SNE [83]. To be specific, we use the ResNet-50 (surrogate model) to generate untargeted adversarial examples and use the DenseNet-121 (target model) to obtain the final latent representation. Further distance indicates better performance for the untargeted attack. As shown in Fig. 11, though the final latent representations of the adversarial examples generated by SGM show more difference with the clean images than PGD, our method yields features that clearly separate the clean and adversarial images, compared with PGD [40] and SGM [13]. This further validates the superiority of our proposed DRA.

VI. CONCLUSION

In this paper, we propose a novel understanding of adversarial transferability from the data distribution perspective. We find that moving the image out of its original distribution can enhance the untargeted adversarial transferability and dragging the image toward the target distribution contributes to high targeted adversarial transferability. We propose a method named Distribution-Relevant Attack (DRA), which apply a fine-tuned surrogate model to generate more transferable adversarial images with data-distribution relevant information. Technically, we propose to fine-tune the surrogate model with a gradient matching method to match the gradient of the model and the gradient of the data distribution, which enables us to push the image away from its original distribution with the gradient of the model. Extensive experiments demonstrate that the proposed DRA establishes state-of-the-art performance in both untargeted and targeted attack scenarios. Moreover, DRA can also effectively fool the real-world computer vision system (the Google Cloud Vision API). Our finding not only motivates researchers to rethink the adversarial transferability from a data distribution perspective but also provides a strong counterpart for future research on adversarial defense.

PROOF OF THE EQUATION (7)

We assume that \( g(x) \) and \( f(x) \) are differentiable in Eq. (7). We can get the following equation:

\[
\nabla x_i [g(x) \nabla x_j f(x)] = \nabla x_k g(x) \nabla x_j f(x) + g(x) \nabla^2 f(x).
\]

We consider this as a function of \( x_i \) alone, all other variables being fixed. Then, integrating over \( x_i \in \mathbb{R} \), we get the equation:

\[
\lim_{M \to \infty} g(x) \nabla x_j f(x) |^{+M}_{-M} = \int_{-\infty}^{+\infty} \nabla x_i g(x) \nabla x_j f(x) dx_i \quad \text{(14)}
\]

which can prove the Eq. (7). “\( +M \)” represents the vector \( [x_1, \ldots, x_i-1, +M, x_{i+1}, \ldots, x_n] \). “\( -M \)” represents the vector \( [x_1, \ldots, x_{i-1}, -M, x_{i+1}, \ldots, x_n] \).

REFERENCES

[1] R. Raghavendra, K. B. Raja, and C. Busch, “Presentation attack detection for face recognition using light field camera,” IEEE Trans. Image Process., vol. 24, no. 3, pp. 1060–1075, Mar. 2015, doi: 10.1109/TIP.2015.2439591.
[2] J. Guo, X. Zhu, Z. Lei, and S. Z. Li, “Decomposed meta batch normalization for fast domain adaptation in face recognition,” IEEE Trans. Inf. Forensics Security, vol. 16, pp. 3082–3095, 2021, doi: 10.1109/TIFS.2021.3073823.
[3] S. R. Arashloo, “Matrix-regularized one-class multiple kernel learning for unseen face presentation attack detection,” IEEE Trans. Inf. Forensics Security, vol. 16, pp. 4635–4647, 2021, doi: 10.1109/TIFS.2021.3117166.
[4] Z. Sun, S. Balakrishnan, L. Su, A. Bhuyan, P. Wang, and C. Qiao, “Who is in control? Practical physical layer attack and defense for mmWave-based sensing in autonomous vehicles,” IEEE Trans. Inf. Forensics Security, vol. 16, pp. 3199–3214, 2021, doi: 10.1109/TIFS.2021.3076287.
[5] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, and J. M. Dolan, “Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Nov. 2019, pp. 1326–1334.
[6] S. Milz, G. Arbeiter, C. Witt, B. Abdallah, and S. Yogamani, “Visual SLAM for automated driving: Exploring the applications of deep learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2018, pp. 247–257. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018_workshops/w9/html/Milz_Visual_SLAM_for_CVPR_2018_paper.html
