UNDERSTANDING AND ENHANCING THE TRANSFERABILITY OF ADVERSARIAL EXAMPLES

Lei Wu[1], **Zhanxing Zhu**[2, 3], **Cheng Tai**[2, 3], and **Weinan E**[2, 3, 4]

[1]School of Mathematical Sciences, Peking University, Beijing, China
[2]Center for Data Science, Peking University, Beijing, China
[3]Beijing Institute of Big Data (BIBDR)
[4]Department of Mathematics and PACM, Princeton University, Princeton, NJ, USA
leiwu@pku.edu.cn, zhanxing.zhu@pku.edu.cn, chengtai@pku.edu.cn,
weinan@math.princeton.edu

ABSTRACT

State-of-the-art deep neural networks are known to be vulnerable to adversarial examples, formed by applying small but malicious perturbations to the original inputs. Moreover, the perturbations can *transfer across models*: adversarial examples generated for a specific model will often mislead other unseen models. Consequently the adversary can leverage it to attack deployed systems without any query, which severely hinder the application of deep learning, especially in the areas where security is crucial.

In this work, we systematically study how two classes of factors that might influence the transferability of adversarial examples. One is about model-specific factors, including network architecture, model capacity and test accuracy. The other is the local smoothness of loss function for constructing adversarial examples. Based on these understanding, a simple but effective strategy is proposed to enhance transferability. We call it *variance-reduced attack*, since it utilizes the variance-reduced gradient to generate adversarial example. The effectiveness is confirmed by a variety of experiments on both CIFAR-10 and ImageNet datasets.

1 INTRODUCTION

With the resurgence of neural networks, more and more large neural network models are applied in real-world applications, such as speech recognition, computer vision, etc. While these models have exhibited good performance, recent works (Szegedy et al. (2013); Goodfellow et al. (2014)) show that an adversary is able to fool the model into producing incorrect predictions by manipulating the inputs maliciously. The corresponding manipulated samples are called *adversarial examples*. More seriously, it is found that they have cross-model generalization ability, i.e. the adversarial example generated from one model can fool another different model with a significant probability. We refer to such property as *transferability*. In consequence, hackers can employ this property to attack black-box systems with only limited number of queries (Papernot et al. (2016b); Liu et al. (2016)), inducing serious security issue to deep learning system.

The adversarial vulnerability was first investigated By Szegedy et al. (2013), in which sophisticated L-BFGS was used to generate adversarial examples. Later, a large number of attacks (Goodfellow et al. (2014); Papernot et al. (2016a); Liu et al. (2016); Carlini & Wagner (2016); Chen et al. (2017); Brendel et al. (2017)) are proposed. Among all the categories of attacks, those based on transferability could be the most dangerous and mysterious, since they surprisingly do not require any input-output query of the target system. Understanding the mechanism of adversarial transferability could potentially provide various benefits for modern deep learning models. Firstly, for the already deployed and vulnerable deep neural networks in real systems, it could help to design better strategies to improve the robustness to the transfer-based attacks. Secondly, revealing the mystery behind the transferability of adversarial examples could also extend the existing understandings on modern deep learning, particularly on the effects of model capacity (Fawzi et al. (2015); Madry et al. (2017)) and

model interpretability (Dong et al. (2017); Ross & Doshi-Velez (2017)). Therefore, studying the transferability of adversarial example in the context of deep networks is of significant importance.

In this paper, we particularly focus on investigating two classes of key factors that might influence the adversarial transferability. Inspired by these understandings, we deign a simple but rather effective strategy for enhancing the transferability of adversarial examples. Our contributions are summarized as follows.

- We numerically explore how adversarial transferability relies on the model-specific factors, including the architecture, test accuracy and model capacity. First, it is found that adversarial transfer is not symmetric, i.e. adversarial examples generated from model A can transfer to model B easily does not means the reverse is also natural. This suggests that the explanation based on the similarity of decision boundary is not sufficient, since similarity itself is a symmetric quantity. Second, multi-step attacks seem to outperform one-step attacks in most cases, which is contradictory to the finding in (Kurakin et al. (2016)). Last, we find that adversarial examples generated from a large model appear less transferable than a small model, under the condition they both have good test performance. Interestingly, this finding seems closely related to the previous studies (Madry et al. (2017); Kurakin et al. (2016)) which showed large models are more robust than small ones.

  We also investigate the influence of properties of loss surface. Specifically, our investigation reveals that the local non-smoothness of loss surface harms the transferability of generated adversarial examples. This is consistent with the study of obfuscated gradients (Athalye et al. (2018)) and gradient masking (Tramèr et al. (2017a)).
- Based on previous investigations, we propose a simple but rather effective approach to enhance the transferability. Inspired by the works (Smilkov et al. (2017); Balduzzi et al. (2017)), we suggest applying the locally averaged gradient instead of the original one to generate adversarial examples. We call it *variance-reduced attack*, since the local averaging have the smoothing effect which suppresses the local oscillation of the loss surface. The effectiveness of our method is justified on both CIFAR-10 and ImageNet datasets, where a large number of state-of-the-art architectures are tested. Different from ensemble-based approaches proposed by Liu et al. (2016), our proposal does not require training lots of extra models, which is typically overly expensive in practice. Moreover, we numerically demonstrate that variance-reduced gradient can be combined with ensemble-based and momentum-based (Dong et al.) approaches seamlessly for producing stronger attacks.

2 Related Work

The phenomenon of adversarial transferability was first observed and investigated by Szegedy et al. (2013). Goodfellow et al. (2014) showed that adversarial training can alleviate the transferability slightly, which is recently extended and improved in (Tramèr et al. (2017a)) by incorporating to the adversarial examples generated by a large ensemble. Based on the transferability, Papernot et al. (2016b;a) proposed a practical black-box attack by training a substitute model with queried information. Liu et al. (2016) showed that targeted transfer is much harder than non-targeted one, and additionally introduced the ensemble-based attacks. More recently, Dong et al. showed that momentum can help to boost transferability significantly, and by utilizing this property, they won the first-place in *NIPS 2017 Non-targeted Adversarial Attack and Targeted Adversarial Attack* competitions. It is also worth mentioning the work by Moosavi-Dezfooli et al. (2016), which demonstrated the exists of adversarial perturbations that have cross-sample transferability. Meanwhile, there exist several works trying to explain the adversarial transferability. Papernot et al. (2016a) contributed it to the similarity between data gradients of source and target models. By visualization, Liu et al. (2016) suggested that the transferability comes from the similarity of decision boundaries. Tramèr et al. (2017b) provided some systematic investigation of this similarity. Unfortunately, similarity is symmetric and it cannot explain our finding that transfer is asymmetric.

Our proposed method enhances the transferability by utilizing informations in the small neighborhood of the clean example, which is inspired by the works on shattered gradient (Balduzzi et al. (2017)) and model interpretability (Smilkov et al. (2017)). Recently, similar strategies are also explored by Athalye et al. (2018) and Warren He (2018) for white-box attacks, while we focus on the transferability. Our method is also related to the work by Athalye & Sutskever (2017), which introduced the expectation

over transformation (EOT) method to increase robustness. The EOT formulation is similar to our objective (11), which increases the robustness against Gaussian noise. Chaudhari et al. (2016) suggest that optimizing the landscape smoothed with respect to parameters can lead to solutions generalizing better. Differently, we obtain adversarial perturbation with stronger transferability via smoothing with respect to input x.

3 PRELIMINARIES

3.1 ADVERSARIAL EXAMPLES

We use $f(x) : \mathbb{R}^d \mapsto \mathbb{R}^K$ to denote the function a model represents, where we omit the dependence on the trainable model parameter $\boldsymbol{\theta}$, since it is fixed in this work. For many applications of interest, we always have $d \gg 1$ and $K = O(1)$. According to the local linear analysis in Goodfellow et al. (2014), it is the high dimensionality that makes $f(x)$ vulnerable to the adversarial perturbation. That is, considering the K-category classification problems as an example, for each x, there exists a small perturbation η that is nearly imperceptible to human eyes, such that the i-th output $f_i(x)$ satisfies

$$\operatorname*{argmax}_i f_i(x) = y^{\text{true}}, \quad \operatorname*{argmax}_i f_i(x+\eta) \neq y^{\text{true}}, \tag{1}$$

where y^{true} is the true label of the input x. We call η adversarial perturbation and correspondingly $x^{adv} := x + \eta$ an adversarial example.

The attack (1) is called a non-targeted attack since the adversary has no control over which class the input x will be misclassified to. In contrast, a *targeted attack* aims at fooling the model to produce a wrong label specified by the adversary, i.e.

$$\operatorname*{argmax}_i f_i(x+\eta) = y^{\text{target}}.$$

In the *black-box attack* setting, the adversary has no knowledge of the target model (e.g. architecture and parameters) and is not allowed to query input-output pairs from the model, i.e. the target model is a pure black-box. However the adversary can construct adversarial examples on a local model (also called the source model) that is trained on the same or similar dataset with the target model. Then it deploys those adversarial examples to fool the target model. This is typically referred to as a black-box attack, as opposed to the *white-box* attack whose target is the source model itself.

3.2 MODELS OF GENERATING ADVERSARIAL EXAMPLES

In general, crafting adversarial perturbation can be modeled as the following optimization problem,

$$\begin{aligned} \text{maximize}_{x'} \quad & J(x') := J(f(x'), y^{\text{true}}) \\ \text{s.t.} \quad & \|x' - x\| \leq \varepsilon, \end{aligned} \tag{2}$$

where J is some loss function measuring the discrepancy between the model prediction and ground truth; $\|\cdot\|$ is certain norm metric to quantify the magnitude of the perturbation. For image data, there is also an implicit constraint: $x' \in [0, 255]^d$, with d being the number of pixels. In practice, the common choice of J is the *cross entropy*. Carlini & Wagner (2016) introduced a loss function that directly manipulates the output logits instead of probability, which has been also adopted in many works. As to the measurement of distortion, it should be chosen to encourage imperceptibility (Xiao et al. (2018)). In this paper, the widely used ℓ_∞ norm is considered.

Ensemble-based attacks To improve the strength of adversarial transferability, instead of using a single model, Liu et al. (2016) suggested using a large ensemble consisting of $f_1, f_2, \cdots, f_Q$ as our source model. Specifically, the non-targeted ensemble-based attack is given by

$$\begin{aligned} \text{maximize}_{x'} \quad & J(\sum_{i=1}^{Q} w_i f_i(x'), y^{\text{true}}) \\ \text{s.t.} \quad & \|x' - x\| \leq \varepsilon, \end{aligned} \tag{3}$$

where w_j is the ensemble weight satisfying $\sum_{j=1}^{Q} w_j = 1$. Following (Liu et al. (2016)), we choose $w_j = 1/Q$ in this paper. The targeted counterpart can be derived similarly.

3.3 OPTIMIZER

There are various optimizers to solve problem (2). Instead of using the classic optimizers, such as Adam, SGD, etc. as in (Liu et al. (2016); Carlini & Wagner (2016)), we adopt a more efficient Frank-Wolfe optimizer. In each iteration, we solve the linear approximation of objective function at current solution $\boldsymbol{x}_t$ in a constrained space,

$$\begin{aligned} s_t &= \text{argmax}_{\|s_t\| \leq \varepsilon} J(x_t) + \langle \nabla_x J(x_t), s_t \rangle \\ x_{t+1} &= \text{proj}_D(x_t + \alpha\, s_t) \end{aligned} \tag{4}$$

where $\mathcal{D} = [0, 255]^d \cap \{x' \,|\, \|x' - x\| \leq \varepsilon\}$ and α is the step size. For ℓ_∞ norm, the Eqn (4) has an explicit solution given by

$$x_{t+1} = \text{proj}_D\left(x_t + \alpha\ \text{sign}(\nabla_x J(x_t))\right). \tag{5}$$

The attack by evolving (5) for T steps is called *iterative gradient sign method* (**IGSM**). Kurakin et al. (2016) and Madry et al. (2017) showed this method can generate strong adversarial examples. Furthermore, the famous *fast gradient sign method* (**FGSM**) is a special case with $\alpha = \varepsilon, T = 1$. Kurakin et al. (2016) and Tramèr et al. (2017a) suggested that FGSM can generate more transferable adversarial examples than IGSM. Therefore, both of them are considered in our work.

Momentum-based approaches Dong et al. recently proposed the momentum-based approaches to enhance adversarial transferability. Specifically, the momentum iterative gradient sign method (**m-IGSM**) is given by

$$\begin{aligned} g_{t+1} &= \mu\, g_t + \nabla J(x_t)/\|\nabla J(x_t)\|_1 \\ x_{t+1} &= \text{proj}_D\left(x_t + \alpha\ \text{sign}(g_t)\right), \end{aligned} \tag{6}$$

where μ is the decay factor of momentum.

3.4 EVALUATION OF ADVERSARIAL TRANSFERABILITY

Datasets To evaluate the transferability, three datasets including MNIST, CIFAR-10 and ImageNet are considered. For ImageNet, directly evaluation on the whole ILSVRC2012 validation dataset is too time-consuming. Therefore, in each experiment we randomly select $5,000$ images that can be correctly recognized by all the examined models.

Trained Models (i) For MNIST, we trained fully connected networks (FNN) of D hidden layers, with the width of each layer being 500. For instance, the architecture of FNN width $D = 2$ is $784 - 500 - 500 - 10$. All the networks are trained to achieve 100% on training set, and test accuracies of them are higher than 98%. (ii) For CIFAR-10, we trained five models: *lenet,resnet20, resnet44, resnet56, densenet*, and test accuracies of them are $76.9\%, 92.4\%, 93.7\%, 93.8\%$ and 94.2%, respectively. (iii) As for ImageNet, the pre-trained models are used provided by PyTorch, including *resnet18, resnet34, resnet50, resnet101, resnet152, vgg11_bn, vgg13_bn, vgg16_bn, vgg19_bn, densenet121, densenet161, densenet169, densenet201, alexnet, squeezenet1_1*. The Top-1 and Top-5 accuracies of them can be found on website[1]. To increase the reliability of experiments, all the models have been tested. However, for a specific experiment we only choose some of them to present since the findings are consistent among the tested models.

Criteria Given a set of adversarial pairs, $\{(x_1^{adv}, y_1^{\text{true}}), (x_2^{adv}, y_2^{\text{true}}), \ldots, (x_N^{adv}, y_N^{\text{true}})\}$, we calculate their *Top-1 success rates* (%) fooling a given model $f(x)$ by

$$100 \times \frac{1}{N} \sum_{i=1}^{N} 1[\underset{i}{\text{argmax}}\, f_i(x_i^{adv}) \neq y_i^{\text{true}}]. \tag{7}$$

If $f(x)$ is the model used to generate adversarial examples, then the rate indicates the the white-box attack performance. For targeted attacks, each image x^{adv} is associated with a pre-specified label $y^{\text{target}} \neq y^{\text{true}}$. Then we evaluate the performance of the targeted attack by the following *Top-1 success rate* (%),

$$100 \times \frac{1}{N} \sum_{i=1}^{N} 1[\underset{i}{\text{argmax}}\, f_i(x_i^{adv}) = y_i^{\text{target}}]. \tag{8}$$

[1] http://pytorch.org/docs/master/torchvision/models.html

The corresponding Top-5 rates can be computed in a similar way.

Attacks Throughout this paper the cross entropy [2] is chosen as our loss function J. We measure the distortion by ℓ_∞ norm, both FGSM and IGSM attacks are considered.

4 How Model-specific Factors Affect Transferability

Previous study on transferability mostly focused on the influence of attack methods (Liu et al. (2016); Dong et al.; Tramèr et al. (2017b); Kurakin et al. (2016)). However it is not clear how the choice of source model affects the success rate transferring to target models. In this section, three factors of source model, including architecture, test accuracy and model capacity, are investigated.

4.1 Architecture

Here we explore how the architecture similarity between source and target model contributes to the transferability. This study is crucial since it can provide us guidance to choose the appropriate source models for effective attacks. To this end, three popular architectures including ResNet, DenseNet and VGGNet are considered, and for each architecture, two networks are selected. Both FGSM and IGSM attacks are performed on ImageNet dataset. Table 1 presents the experiment results, and the choice of hyper-parameter is detailed in the caption.

As we can see, in most cases IGSM outperforms FGSM significantly, especially when the source and target models have the similar architecture. This observation contradicts the finding by Kurakin et al. (2016) that multi-step attacks are somewhat less transferable than single-step attacks. However for the attacks from VGGNets to all the ResNets and DenseNets, we do observe IGSM generate adversarial examples weaker than FGSM (blue rates in the figure). So, in general, which method is better should be similarity dependent.

Another finding is that the transfers between two models are non-symmetric, and this phenomenon is more obvious for the models with different architectures. For instance, the success rates of IGSM attacks from *densenet121* to *vgg13_bn* is 58.7%, however the rate from *vgg13_bn* to *densenet121* has only 19.1%. Moreover, the success rates between models with similar architectures is much higher. For example the success rates of IGSM attacks between *vgg13_bn* and *vgg16_bn* are higher than 90%, nearly twice the ones of attacks formed by any other architectures.

Table 1: Top-1 success rates(%) of FGSM and IGSM attacks between pairs of models. The row and column denote the source and target model, respectively. For each cell, the left is the success rate of FGSM ($\varepsilon = 15$), while the right is that of IGSM ($T = 5, \alpha = 5, \varepsilon = 15$).

	resnet18	resnet101	vgg13_bn	vgg16_bn	densenet121	densenet161
resnet18	-	36.9 / 43.4	51.8 / 58.0	45.1 / 51.7	41.1 / 49.2	30.0 / 35.8
resnet101	48.5 / 57.2	-	38.9 / 41.6	33.1 / 40.0	33.2 / 46.9	28.7 / 43.2
vgg13_bn	35.5 / 26.8	14.8 / 10.8	-	58.8 / 90.7	19.1 / 15.9	13.8 / 11.7
vgg16_bn	35.2 / 26.1	15.6 / 11.1	61.9 / 91.1	-	21.1 / 16.8	15.8 / 13.2
densenet121	49.3 / 63.8	34.4 / 50.7	47.6 / 58.7	41.0 / 57.8	-	38.5 / 73.6
densenet161	45.7 / 56.3	33.8 / 54.6	48.6 / 56.0	41.3 / 55.9	43.4 / 78.5	-

4.2 Model Capacity and Test Accuracy

We first study this problem in ImageNet dataset, with *vgg19_bn*[3] chosen as our target model. A variety of models are used as source models to perform both FGSM and IGSM attacks. The results are displayed in Figure 1. The horizontal axis is the Top-1 test error, while the vertical axis is the number of model parameters that roughly quantifies the model capacity. We can see that the models

[2] We also tried the loss described in Carlini & Wagner (2016) but did not find its superiority to cross entropy. The reason might be that hard constraints, instead of soft penalizations, are used in our formulation.

[3] We also tried other nets, and the results show no difference.

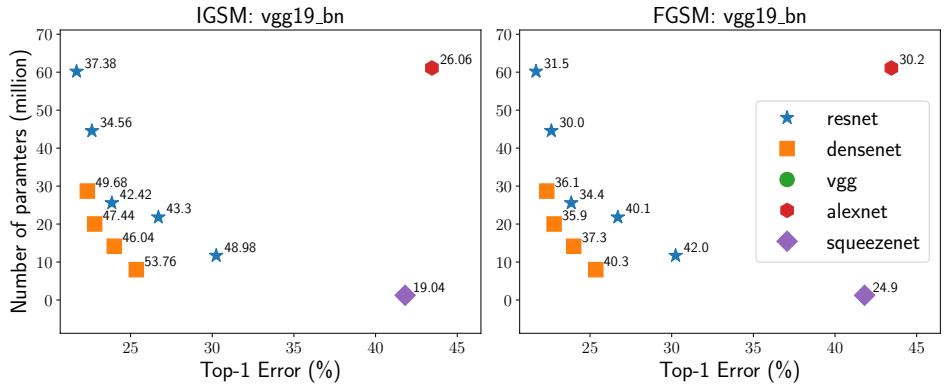


Figure 1: Top-1 success rates of FGSM($\varepsilon = 15$) and IGSM ($k = 20, \alpha = 5, \varepsilon = 15$) attacks against *vgg19_bn* for various models. The annotated value is the percentage of adversarial examples that can transfer to the *vgg_19*. Here, the models of vgg-style have been removed to exclude the influence of architecture similarity.

with powerful attack capability concentrate in the bottom left corner, and the fooling rates are much lower for those models with either large test error or large number of parameters.

The decision boundaries of high-accuracy models should be similar, since all of them approximate the ground-truth decision boundary of data very well. On the contrary, a low-accuracy model must has a decision boundary relatively different from the high-accuracy models. Therefore it is not surprising to observe that **high-accuracy models tend to exhibit stronger attack capability**.

However it is somewhat strange that **adversarial examples generated from deeper model appear less transferable**. To further confirm this observation, we conduct additional experiments on MNIST and CIFAR-10. Table 2 shows the results, which is basically consistent with previous observation, though there exists counter-examples, attacks from *resnet44,resnet56* to *resnet20*. It suggests us not to use deep models as the source models when performing transfer-based attacks, even though we still can not fully understand it. It is also worth to mention that some works (Kurakin et al. (2016); Madry et al. (2017)) observed a dual phenomenon that deeper models are more robust against adversarial perturbations.

Table 2: Each cell (S,T) denotes the Top-1 success rate of attack from source model S to target model T.

(a) MNIST, FGSM attack with $\varepsilon = 40$

	$D=0$	$D=2$	$D=4$	$D=8$
$D=0$	-	62.9	62.9	64.4
$D=2$	52.9	-	48.3	49.4
$D=4$	47.3	43.1	-	44.8
$D=8$	31.2	29.2	29.0	-

(b) CIFAR-10, FGSM attack with $\varepsilon = 10$

	resnet20	resnet44	resnet56	densenet
resnet20	-	70.4	64.0	71.6
resnet44	65.4	-	57.1	65.8
resnet56	66.2	62.9	-	40.3

5 Shattered Gradients

In previous section, we explore how model-specific factors influence the transferability of adversarial examples. In this part, we systematically investigate how the smoothness of $J(x)$ impacts the transferability across models.

For simplicity, assume model A and B are the source and target models, respectively and let $g(x) := \nabla J(x)$ be the gradient. Previous methods use $g_A(x)$ to generate adversarial perturbations, so the transferability mainly depends on how much instability of g_A can transfer to model B. The shattered gradients phenomenon studied in (Balduzzi et al. (2017)) implies that gradient g_A is very noisy,

though the model is trained on training set very well. We hypothesize that the noise hurts the transferability of g_A, as illustrated in cartoon Figure 2(a). Since both model A and B have very high test accuracy, their level sets should be similar globally, and J_B is probably unstable along g_A. However, as illustrated in the figure, the local fluctuation of g_A make the sensitivity less transferable. One way to alleviate it is to smooth the landscape J_A, thereby yielding a more transferable direction G_A, i.e. $\langle \hat{G}_A, \hat{g}_B \rangle > \langle \hat{g}_A, \hat{g}_B \rangle$.

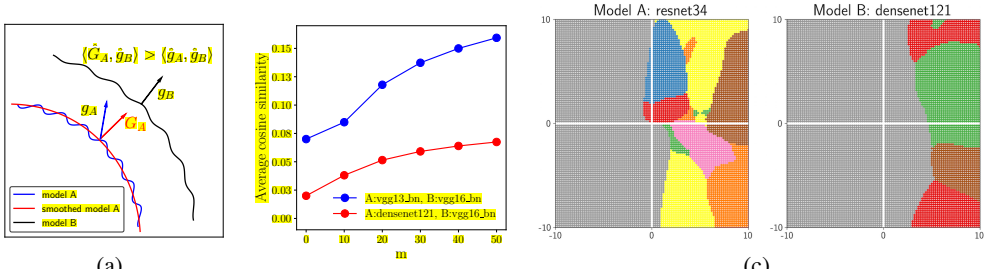


Figure 2: (a) The three solid lines denote the level set of landscapes. The hatted vector denotes the corresponding unit vector. (b) Cosine similarity between the gradients of source and target models. (c) Visualization of decision boundaries. The origin corresponds to the clean image shown in Figure 6 of Appendix.

Inspired by the standard technique from distribution theory, we suggest to smooth the landscape as follows,

$$J_\sigma(x) := \int J(x - x')\psi_\sigma(x')dx', \tag{9}$$

where J_σ is the smoothed landscape, and the smoothing is achieved by convolution with a mollifier, defined as a smooth function satisfying,

$$\psi_\sigma(x) = \sigma^{-d}\psi(x/\sigma) \qquad \int_{\mathbb{R}^d} \psi(x)dx = 1.$$

In this paper, the Gaussian mollifier, i.e. $\psi(x) = \frac{1}{(2\pi)^d} e^{-\|x\|^2/2}$, is used, and the corresponding gradient can be calculated by

$$G_\sigma(x) = \mathbb{E}_{\xi \sim \mathcal{N}(0,\sigma^2)}[g(x + \xi)]. \tag{10}$$

This strategy has been employed by Smilkov et al. (2017) to improve the interpretability of gradient saliency maps, which demonstrates that the local averaged gradient $G_\sigma(x)$ is more informative and interpretable than $g(x)$. Here, we further show that $G_\sigma(x)$ *is more transferable than* $g(x)$ by numerical experiments on ImageNet.

We first quantify the cosine similarity between gradients of source and target models, respectively. Two attacks are considered: *vgg13_bn→vgg16_bn*, *densenet121→vgg13_bn*, which represent the within-architecture and cross-architecture transfer, respectively. We choose $\sigma = 15$, and the expectation in (10) is estimated by using $\frac{1}{m}\sum_{i=1}^{m} g(x + \xi_i)$. To verify the averaged gradients do transfer better, we plot the cosine similarity between source and target model versus the number of samples m. Figure 2(b) display the average similarity calculated from $5,000$ randomly selected samples, which shows that the cosine similarity between G_A and g_B are indeed larger than the one between g_A and g_B. As expected, the similarity increases with m monotonically, which further justifies that G_A is more transferable than g_A.

Next, we visualize the transferability in Figure 2(c) by comparing the decision boundaries of *resnet34* (model A, the source model) and *densenet121* (model B, the target model). The horizontal axis represents the direction of G_A of *resnet34*, estimated by $m = 1000, \sigma = 15$, and the vertical axis denotes orthogonal direction $h_A := g_A - \langle g_A, \hat{G}_A \rangle \hat{G}_A$. Each point in the 2-D plane corresponds to the image perturbed by u and v along each direction,

$$\text{clip}(x + u\,\hat{G}_A + v\,\hat{h}_A, 0, 255),$$

where x is the clean image. It can be easily observed that for *resnet34* (model A), a small perturbation in both directions can produce wrong classification. However, when applied to *densenet121* (model B), a large perturbation along h_A cannot change the classification result, while a small perturbation along G_A can change the prediction easily. This further confirms that local averaging indeed extracts the more transferable part of g_A.

The above analysis implies that the local oscillation of loss surface do harm the transferability of adversarial examples. This is similar to the phenomenon of gradient masking observed in the defense of adversarial example (Tramèr et al. (2017a); Papernot et al. (2016c); Athalye et al. (2018)). Our experiments suggest that gradient masking also exists for normally trained models to some extent, i.e. shattered gradients. The noise contained in the shattered gradient makes the generated adversarial examples less transferable.

6 Variance-reduced Attack

6.1 Method

Previous study suggests to alleviate the shattering of gradients by optimizing the smoothed loss function,

$$\begin{aligned} \text{maximize} \quad & J_\sigma(x') := \mathbb{E}_{\xi\sim\mathcal{N}(0,\sigma^2)}[J(x'+\xi)] \\ \text{s.t.} \quad & \|x'-x\| \le \varepsilon. \end{aligned} \tag{11}$$

Intuitively, this method can also be interpreted as generating adversarial examples that are robust to Gaussian perturbation. We expect that the generated robust adversarial examples can still survive easily in spite of the distinction between source and target model.

Using the iterative gradient sign method to solve (11) yields the following iteration formula:

$$\begin{aligned} G_t &= \frac{1}{m}\sum_{i=1}^{m} \nabla J(x^t+\xi_i), \quad \xi_i \sim \mathcal{N}(0,\sigma^2 I) \\ x_{t+1} &= \text{proj}_D\left(x_t + \alpha\,\text{sign}\left(G_t\right)\right), \end{aligned} \tag{12}$$

where G_t is a mini-batch approximation of $\mathbb{E}_{\xi\sim\mathcal{N}(0,\sigma^2 I)}[\nabla J(x+\xi)]$. Compared to IGSM, the gradient is replaced by an averaged one, which removes the local fluctuation. Therefore we call this method variance-reduced iterative gradient sign method (vr-IGSM). The special case $T=1, \alpha=\varepsilon$, is accordingly called variance-reduced fast gradient sign method (vr-FGSM). For any other optimizer, the corresponding variance-reduced version can be derived similarly. However, in this paper we only consider IGSM and FGSM for simplicity.

6.2 Effectiveness

In this section, we evaluate the effectiveness over both CIFAR-10 and ImageNet datasets.

Table 3: Top-1 success rates (%) of attacks between pairs of models. The row and column denote the source and target model, respectively. For each cell, the left is the success rate of the normal attack the right is that of corresponding variance-reduced attack. In this experiment, distortion $\varepsilon = 10$

(a) FGSM versus vr-FGSM ($\sigma = 15, m = 100$)

	lenet	resnet-20	densenet
lenet	-	29.0 / 28.7	28.7 / 29.0
resnet-20	25.4 / 30.3	-	71.6 / 90.0
densenet	26.0 / 31.9	72.0 / 90.5	-

(b) IGSM($T = 5, \alpha = 4$) versus vr-IGSM($T = 5, \alpha = 4, m = 20, \sigma = 15$)

	lenet	resnet-20	densenet
lenet	-	30.8 / 31.7	30.6 / 31.3
resnet-20	24.9 / 26.9	-	**85.9 / 97.6**
densenet	25.9 / 28.4	**92.9 / 99.0**	-

6.2.1 CIFAR-10

We first consider the adversarial transferability among three models: *lenet*, *resnet-20* and *densenet*. The test accuracies of them are 76.9%, 92.4% and 94.2% respectively. Table 3 shows, in general,

variance-reduced gradient indeed improve the transfer rate consistently. In particular, the transferability between *resnet-20* and *densenet* are enhanced significantly. However the variance-reduced gradient does not help too much for the transfers from *lenet* to *resnet-20* and *densenet*. It is interesting to note that *lenet* has very weak attack capability, which probably is due to its low accuracy. It does not learn the data very well, causing that its decision boundary be very different from the high-accuracy model *resnet20* and *densenet*. This is consistent with our finding in Section 4.2.

6.2.2 IMAGENET

Previous experiments show that our method indeed can improve the success rate. We now turn to the more real dataset, imageNet. To make the experimental result reliable, both one-step and multi-step methods, targeted and non-targeted attacks, single-model and ensemble-model based approaches are examined.

Single-model based approaches Here we test both one-step and multi-step attacks, specifically for multi-step attacks, we make the number of gradient calculation per sample be fixed 100 for a fair comparison. The results of multi-step attacks are shown in Table 4 (readers can refer to Table 7 in Appendix for the result of one-step attack). As we can see, our method enhance the transferability dramatically for all the attacks. Please especially note those bold rates, where the improvement has reached about 30%.

Table 4: Top-1 success rates(%) of non-targeted IGSM and vr-IGSM attacks between pairs of models. The row and column denote the source and target model, respectively. For each cell, the left is the success rate of IGSM ($T = 100, \alpha = 1$), while the right is the that of vr-IGSM ($m = 20, \sigma = 15, T = 5, \alpha = 5$). In this experiment, distortion $\varepsilon = 15$.

	densenet121	resnet152	resnet34	vgg13_bn	vgg19_bn
densenet121	-	**50.1 / 80.6**	59.9 / 87.2	62.2 / 82.2	56.5 /84.3
resnet152	**52.5 / 81.3**	-	57.2 / 85.6	47.7 / 71.1	**42.9 / 72.6**
resnet34	51.5 / 76.4	46.5 / 73.1	-	53.8 / 74.8	49.1 / 74.5
vgg13_bn	24.1 / 49.2	14.3 / 33.5	**25.1 / 54.1**	-	90.6 / 96.4
vgg19_bn	**27.1 / 57.5**	16.7 / 41.6	**27.6 / 60.7**	92.0 / 96.1	-

Ensemble based approaches In this part, we apply the ensemble-based attack proposed by Liu et al. (2016), which is rather effective in generating strong transferable adversarial examples. We examine whether we can further improve its performance by employing our variance-reduced gradient.

Both targeted [4] and non-targeted attacks are considered. For non-targeted attacks, the Top-1 success rates are nearly saturated; for targeted attacks, generating an adversarial example predicted by target models as a specific label is too hard, resulting in a very small success rate. Therefore, we instead adopt the the Top-5 success rate as our criterion to better reflect the improvement of method. A variety of model ensembles are examined, and the results are summarized in Table 5 (The corresponding Top-1 success rate can be found in Appendix).

As we can see, it is clear that variance-reduced attacks outperform the corresponding normal ones by a remarkable large margin for both non-targeted and targeted attacks. More importantly, the improvement never be harmed compared to single-model case, which implies that **variance-reduced gradient can be effectively combined with ensemble method without compromise**.

Momentum-based approaches Momentum-based attacks are recently proposed by Dong et al., which won the first place in *NIPS 2017 Non-targeted Adversarial Attack and Targeted Adversarial Attack* competitions. Here we compare our variance-reduced attacks with the momentum-based attacks. In this experiment, three networks of different architectures are selected. The momentum

[4] Compared to non-targeted attack, we find that a larger step size α is necessary for generating strong targeted adversarial examples. Readers can refer to Appendix A for more detailed analysis on this issue, though we cannot fully understand it. Therefore a much larger step size than the non-target attacks is used in this experiment.

Table 5: Top-5 success rates (%) of ensemble-based approaches. The row and column denote the source and target model, respectively. For each cell, the left is the rate of normal method, while the right is that the variance-reduced counterpart. The corresponding Top-1 success rates can be found in Appendix B (Table 8 and Table 9).

(a) **Non-targeted attacks**: IGSM ($T = 200, \alpha = 1$) versus vr-IGSM ($T = 10, \alpha = 5, m = 20, \sigma = 15$), distortion $\varepsilon = 15$.

Ensemble	densenet121	resnet152	resnet50	vgg13_bn
resnet18+resnet34+resnet101	43.0 / 75.5	54.5 / 81.6	62.6 / 85.4	42.0 / 74.2
vgg11_bn +densenet161	40.5 / 73.5	**18.5 / 56.4**	33.4 / 70.2	68.3 / 85.6
resnet34+vgg16_bn+alexnet	26.5 / 65.2	**15.7 / 55.3**	33.8 / 72.8	77.8 / 89.9

(b) **Targeted attacks**: IGSM ($T = 20, \alpha = 15$) versus vr-IGSM ($T = 20, \alpha = 15, m = 20, \sigma = 15$), distortion $\varepsilon = 20$.

Ensemble	resnet152	resnet50	vgg13_bn	vgg16_bn
resnet101+densenet121	28.1 / 56.8	26.2 / 52.4	7.7 / 23.6	8.1 / 29.7
resnet18+resnet34+resnet101+densenet121	50.4 / 70.4	54.7 / 72.4	23.2 / 44.2	28.1 / 52.6
vgg11_bn+vgg13_bn+resnet18 +resnet34+densenet121	24.3 / 55.8	36.9 / 65.9	-	62.2 / 83.5

decay factor $\mu = 1$ is chosen as suggested in (Dong et al.), and for variance-reduced gradient, we use $m = 20, \sigma = 15$. All attacks are iterated for $T = 5$ with step size $\alpha = 5$. Table 6 reports the Top-1 success rates of non-targeted attacks of three attacks, including momentum-based IGSM (m-IGSM), vr-IGSM and momentum-based vr-IGSM (m-vr-IGSM).

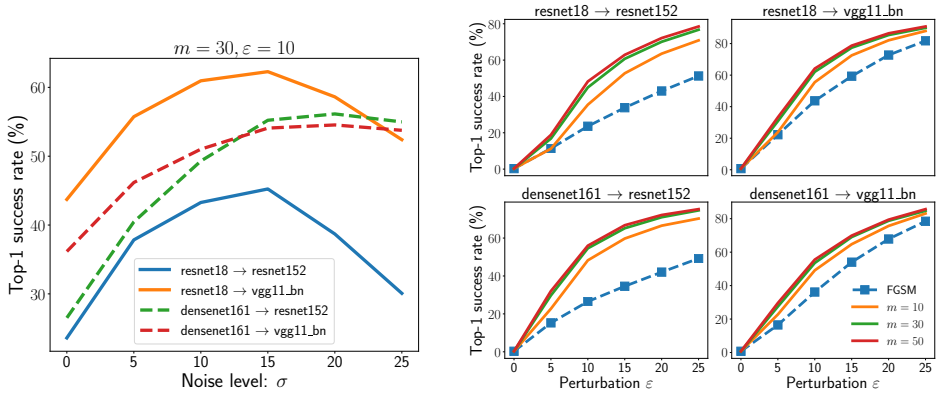
As shown in the table, our method outperforms momentum-based method significantly for all the cases. Furthermore, by combining with the variance-reduced gradient, the effectiveness of momentum-based method is improved significantly.

Table 6: Top-1 success rates(%) of non-targeted attacks. The row and column denote the source and target model, respectively. Each cell contains three rates corresponding m-IGSM, vr-IGSM and m-vr-IGSM, attacks respectively.

	resnet18	densenet121	vgg13_bn
resnet18	-	65.6 / 73.1 / 86.5	70.4 / 77.7 / 86.7
densenet121	72.7 / 84.5 / 91.1	-	68.7 / 80.3 / 86.7
vgg13_bn	43.1 / 58.6 / 74.3	28.4 / 44.7 / 60.9	-

6.3 Influence of hyper parameters

In this part, we explore the sensitivity of hyper parameters m, σ when applying our variance-reduced gradient methods for black-box attacks. We take ImageNet dataset as the testbed, and vr-FGSM attack is examined. To increase the reliability, four attacks are considered here. The results are shown in Figure 3. It is not surprising that larger m leads to higher success rate for any distortion level ε due to the better estimation of the data-dependent direction of the gradient. We find there is an optimal value of σ inducing the best performance. Overly large σ will introduce a large bias in (12). Extremely small σ is unable to smooth the landscape enough. Moreover the optimal σ varies for different source models, and in this experiment it is about 15 for *resnet18*, compared to 20 for *densenet161*.



(a) (b)

Figure 3: **(a)** The sensitivity of the hyper parameter σ. **(b)** Success rates for nr-FGSM attacks with different m.

6.4 Robustness analysis of adversarial examples

Since variance-reduced attacks can be viewed as generating adversarial examples robust against Gaussian noise perturbation. Here we further examine their robustness to generic image transformations. The robustness is particularly important in practice, since it directly affects whether adversarial examples can survive in the physical world (Kurakin et al. (2016); Athalye & Sutskever (2017); Lu et al. (2017)). To quantify the influence of transformations, we use the notion of destruction rate defined by Kurakin et al. (2016),

$$R = \frac{\sum_{i=1}^{N} c(\boldsymbol{x}_i)\left(1 - c(\boldsymbol{x}_i^{adv})\right) c(T(\boldsymbol{x}_i^{adv}))}{\sum_{i=1}^{N} c(\boldsymbol{x}_i)\left(1 - c(\boldsymbol{x}_i^{adv})\right)},$$

where N is the number of images used to estimate the destruction rate, $T(\cdot)$ is an arbitrary image transformation. The function $c(\boldsymbol{x})$ indicates whether $\boldsymbol{x}$ is classified correctly:

$$c(\boldsymbol{x}) := \begin{cases} 1, & \text{if image } \boldsymbol{x} \text{ is classified correctly} \\ 0, & \text{otherwise} \end{cases}$$

And thus, the above rate R describes the fraction of adversarial images that are no longer misclassified after the transformation $T(\cdot)$.

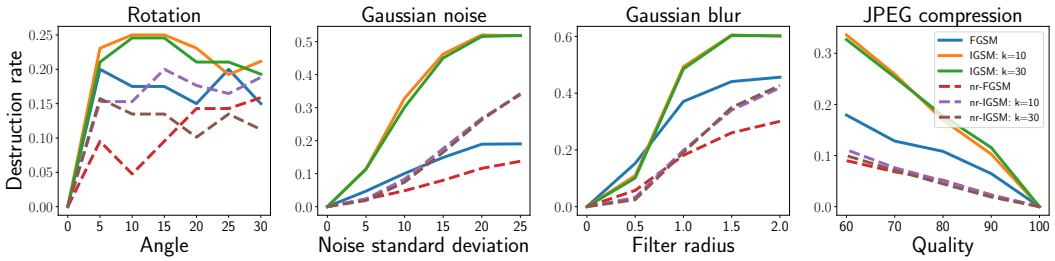


Figure 4: Destruction rates of adversarial examples for various methods. For variance reduced attacks, we choose $m = 20, \sigma = 15$. The distortion $\varepsilon = 15$.

Densenet121 and *resnet34* are randomly chosen as our source and target model, respectively; and four image transformations are considered: rotation, Gaussian noise, Gaussian blur and JPEG compression. Figure 4 displays the results, which show that adversarial examples generated by our methods are much more robust than those generated by vanilla methods. This numerical result is interesting, since we only explicitly increase the robustness against Gaussian noise in generating adversarial examples. This result suggests that the robustness of adversarial examples can also transfer among different image transforms.

7 Discussion and Conclusion

In this paper, we first investigated the influence of model-specific factors on the adversarial transferability. It is found that model architecture similarity plays a crucial role. Moreover models with lower capacity and higher test accuracy are endowed with stronger capability for transfer-based attacks. we second demonstrate that the shattered gradient hinders the transfer of adversarial examples. Motivated by the understanding, we proposed the variance-reduced attack which can enhance the transferability of generated adversarial examples dramatically. Furthermore, the variance-reduced gradient can be combined with both ensemble and momentum based attacks rather effectively. Lastly, we show that, the adversarial examples generated by variance-reduced attacks are much more robust than normal methods.

Our results imply that the robust (at least against Gaussian noise) adversarial examples tend to have stronger cross-model generalization, i.e. transferability. The similar phenomenon is also studied in learning theory (Bousquet & Elisseeff (2002); Roman Novak (2018)), i.e. the stable algorithm and hypothesis always tend to generalize better as well. So it is worth to further investigate if improving the robustness against generic transforms (Athalye & Sutskever (2017)) can enhance the transferability as well.

References

Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. 2018. URL https://arxiv.org/abs/1802.00420.

David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? *arXiv preprint arXiv:1702.08591*, 2017.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2016.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, and Yann LeCun. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26. ACM, 2017.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Xiaolin Hu, Jianguo Li, and Jun Zhu. Boosting adversarial attacks with momentum. *arXiv preprint arXiv:1710.06081*.

Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Fundamental limits on adversarial robustness. *ICML 2015*, 2015.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *arXiv preprint arXiv:1610.08401*, 2016.

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*, 2016a.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint arXiv:1602.02697*, 2016b.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016c.

Daniel A. Abolafia Jeffrey Pennington Jascha Sohl-Dickstein Roman Novak, Yasaman Bahri. Sensitivity and generalization in neural networks: an empirical study. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJC2SzZCW.

Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *arXiv preprint arXiv:1711.09404*, 2017.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

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.

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

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017b.

Dawn Song Warren He, Bo Li. Decision boundary analysis of adversarial examples. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkpiPMbA-.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.

APPENDIX A: INFLUENCE OF STEP SIZE FOR TARGETED ATTACKS

When using IGSM to perform targeted black-box attacks, there are two hyper parameters including number of iteration k and step size α. Here we explore their influence to the quality of adversarial examples generated. The success rates are calculated on $1,000$ images randomly selected according to description of Section 3.4. *resnet152* and *vgg16_bn* are chosen as target models. The performance are evaluated by the average Top-5 success rate over the three ensembles used in Table 5(b).

Figure 5 shows that for the optimal step size α is very large, for instance in this experiment it is about 15 compared to the allowed distortion $\varepsilon = 20$. Both too large and too small step size will yield harm to the attack performances. It is worth to note that with small step size $\alpha = 5$, the large number of iteration provides worse performance than small number of iteration. One possible explanation is that more iterations lead the optimizer to converge to a more overfitted solution. In contrast, a large step size can prevent it and encourage the optimizer to explore more model-independent area, thus more iteration is better.

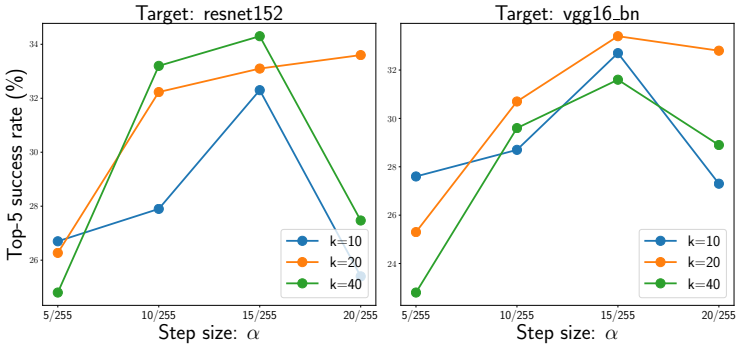


Figure 5: Average success rates over three ensembles for different step size α and number of iteration k. The three ensembles are the same with those in Table 5(b). Distortion $\varepsilon = 20$.

APPENDIX B: SOME ADDITIONAL EXPERIMENTAL RESULTS

Figure 6: The image used to perform decision boundary analysis. Its ID in ILSVRC2012 validation set is 26, with ground truth label being *table lamp*.

Table 7: Top-1 success rates(%) of non-targeted FGSM and vr-FGSM attacks between pairs of models. The row and column denote the source and target model, respectively. The left is the success rate of FGSM, while the right is that of vr-FGSM ($m = 20, \sigma = 15$). In this experiment, distortion $\varepsilon = 15$.

	densenet121	resnet152	resnet34	vgg13_bn	vgg19_bn
densenet121	-	34.4 / 66.7	46.2 / 74.5	53.0 / 72.5	44.9 /71.4
resnet152	39.2 / 67.2	-	45.4 / 71.3	43.3 / 62.4	36.8 / 61.5
resnet34	46.3 / 71.1	38.4 / 66.4	-	54.4 / 70.5	46.7 / 68.8
vgg13_bn	23.0 / 48.4	16.0 / 34.4	28.0 / 53.2	-	54.2 / 84.6
vgg19_bn	28.1 / 58.8	18.7 / 46.4	31.5 / 62.3	62.2 / 87.1	-

Table 8: Top-1 success rates (%) of ensemble-based non-targeted IGSM and nr-IGSM attacks. The row and column denote the source and target model, respectively. The left is the success rate of IGSM ($T = 100, \alpha = 3$), while the right is that of vr-IGSM ($T = 50, \alpha = 3, m = 20, \sigma = 15$). Since targeted attacks are much more difficult, we choose $\varepsilon = 20$.

Ensemble	densenet121	resnet152	resnet50	vgg13_bn
resnet18+resnet34+resnet101	87.8 / **97.8**	94.6 / **98.9**	97.4 / **99.4**	84.1/96.1
vgg11_bn+densenet161	86.8 / 97.2	62.9 / 89.7	80.3 / 94.8	94.9 / 98.4
resnet34+vgg16_bn+alexnet	68.9 / 91.3	54.6 / 87.2	77.9 / 96.2	98.1 / **99.1**

Table 9: Top-1 success rates (%) of ensemble-based targeted IGSM and vr-IGSM attacks. The row and column denote the source and target model, respectively. The left is the success rate of IGSM ($T = 20, \alpha = 15$), while the right is that of vr-IGSM ($T = 20, \alpha = 15, m = 20, \sigma = 15$). Since targeted attacks are harder, we set $\varepsilon = 20$.

Ensemble	resnet152	resnet50	vgg13_bn	vgg16_bn
resnet101+densenet121	11.6 / 37.1	11.9 / 34.5	2.6 / 10.5	2.6 / 14.1
resnet18+resnet34+resnet101+densenet121	30.3 / **55.2**	36.8 / **57.3**	10.8 /**29.1**	12.8/35.0
vgg11_bn+vgg13_bn+resnet18+ resnet34+densenet121	10.1 / 35.1	22.2 / 47.9	-	42.1/**72.1**