EXPLORING AND ENHANCING THE TRANSFERABILITY OF ADVERSARIAL EXAMPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

State-of-the-art deep neural networks are 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 empirically 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 surface for constructing adversarial examples. We then propose a simple but effective strategy to enhance the transferability, whose effectiveness is confirmed by a variety of experiments on both CIFAR-10 and ImageNet datasets.

1 INTRODUCTION

By now, 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, recently Szegedy et al. (2013) showed 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 deployed models without any query (Papernot et al., 2016b; Liu et al., 2017), inducing serious security issue to deep learning system.

The transfer-based attack could be the most dangerous and mysterious, since they surprisingly do not require any input-output query of the target system. Understanding the mechanism 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 (Madry et al., 2018; Fawzi et al., 2015) model interpretability (Dong et al., 2017; Ross and Doshi-Velez, 2018). Therefore, studying the transferability of adversarial example in the context of deep networks is of significant importance.

In this paper, we numerically investigate two classes of factors that might influence the adversarial transferability, and further provide a simple but rather effective strategy for enhancing the transferability. Our contributions are summarized as follows.

- We find 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. Second, we find that adversarial examples generated from a deep model appear less transferable than a shallow model. We also explore the impact of shattered gradients, which reflect the roughness of loss surface. Specifically, we find that the local non-smoothness of loss surface harms the transferability of generated adversarial examples.

- Inspired by previous investigations, We suggest applying the locally averaged gradient instead of the original one to generate adversarial examples. It is dubbed as *smooth gradient*, since the averaging have the smoothing effect which suppresses the local oscillation of the loss surface. Extensive numerical validations justify the effectiveness of our method.

2 Related work

The phenomenon of adversarial transferability was first studied by Szegedy et al. (2013). By utilizing the transferability, Papernot et al. (2016b;a) and proposed a practical black-box attack by training a substitute model with limited queried information. Liu et al. (2017) first studied the targeted transferability and introduced the ensemble-based attacks to improve the transferability. More recently, Dong et al. (2018) showed that momentum can help to boost transferability significantly. Our smoothed gradient attacks that enhance the transferability by utilizing informations in the small neighborhood of the clean example is inspired by the works on shattered gradients (Balduzzi et al., 2017) and model interpretability (Smilkov et al., 2017). Recently, similar strategies are also explored by Athalye et al. (2018a) and Warren He (2018) for white-box attacks, whereas we focus on the transferability. Our method is also related to the work by Athalye et al. (2018b), which introduced the expectation over transformation (EOT) method to increase robustness of adversarial examples. The EOT formulation is similar to our objective (7), but they did not study the transferability. Also our motivation are totally different from theirs.

Meanwhile, there exist several works trying to explain the adversarial transferability. Papernot et al. (2016a) contributed it to the similarity between input gradients of source and target models. By visualization, Liu et al. (2017) suggested that the transferability comes from the similarity of decision boundaries. Our numerical investigations implies that these similarity-based understandings have the intrinsic limitation, which cannot explain the non-symmetric property of adversarial transferability.

3 Background

3.1 Adversarial attack

We use $f(x)$ denote the classifier. In deep learning, it is found that for almost any sample x and its label y^{true}, there exists a small perturbation η that is nearly imperceptible to human, such that

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

We call η adversarial perturbation and correspondingly $x^{adv} := x + \eta$ the 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 this paper, we consider the pure black-box setting. That means 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. On the contrary, the *white-box* attack means the target model is available for adversary.

3.2 Generating adversarial examples

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

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

where J is some loss function measuring the discrepancy between the model prediction and ground truth; $\|\cdot\|_\infty$ is used to quantify the magnitude of the perturbation. Other norm is also possible, but we focus on ℓ_∞ norm in this paper. To improve the strength of adversarial transferability, instead of using a single model, Liu et al. (2017) proposed the *ensemble attack*, which solves the following problem

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

Optimizer We use the following iteration to solve (2) and (3),

$$x_{t+1} = \text{proj}_D \left(x_t + \alpha \ \text{sign}(\nabla_x J(x_t)) \right). \tag{4}$$

where $\mathcal{D} = [0, 255]^d \cap \{x' \mid \|x' - x\| \leq \varepsilon\}$ and α is the step size. We call the attack evolving (4) for T steps iterative gradient sign method (IGSM) attack (Kurakin et al., 2017; Madry et al., 2018). Furthermore, the famous fast gradient sign method (FGSM) is a special case with $\alpha = \varepsilon, T = 1$.

Dong et al. (2018) recently proposed the momentum-based attack as follows

$$\begin{aligned} g_{t+1} &= \mu \, g_t + \nabla J(x_t) / \|\nabla J(x_t)\|_1 \\ x_{t+1} &= \text{proj}_D \left(x_t + \alpha \ \text{sign}(g_t) \right), \end{aligned} \tag{5}$$

where μ is the decay factor of momentum. By using this method, they won the first-place in *NIPS 2017 Non-targeted Adversarial Attack and Targeted Adversarial Attack* competitions. We will call it mIGSM attack in this paper.

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 forming our validation set.

Models (i) For MNIST, we trained fully connected networks (FNN) of D hidden layers, with the width of each layer being 500. (ii) For CIFAR-10, we trained five models: *lenet,resnet20, resnet44, resnet56, densenet*. The test accuracies of them are $76.9\%, 92.4\%, 93.7\%, 93.8\%$ and 94.2%, respectively. (iii) For ImageNet, the pre-trained models provided by PyTorch are used. 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, $\{(x_i^{adv}, y_i^{\text{true}})\}_{i=1}^N$, we calculate their *Top-1 success rates* (%) fooling a given model $f(x)$ by $\frac{100}{N} \sum_{i=1}^N 1[\text{argmax}_i \, f_i(x_i^{adv}) \neq y_i^{\text{true}}]$. 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: $\frac{100}{N} \sum_{i=1}^N 1[\text{argmax}_i \, f_i(x_i^{adv}) = y_i^{\text{target}}]$. The corresponding Top-5 rates can be computed in a similar way.

5 How Model-specific Factors Affect Transferability

Previous study on adversarial transferability mostly focused on the influence of attack methods (Liu et al., 2017; Dong et al., 2018; Tramèr et al., 2017; Kurakin et al., 2017). However it is not clear how the choice of source model affects the success rate transferring to target models. In this section, we investigate this issue from three aspects including architecture, test accuracy and model capacity.

5.1 Architecture

We first 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 one-step and multi-step attacks are performed on ImageNet dataset. Table 1 presents the experiment results, and the choice of hyper-parameter is detailed in the caption.

We can see 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 15.9%. The lack of symmetry implies that previous similarity-based explanations of adversarial transferability is quite limited.

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

Table 1: Top-1 success rates(%) of FGSM and IGSM attacks. 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 ($T = 5, \alpha = 5, \varepsilon = 15$). The dashes corresponds to the white-box cases, which are omitted.

	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	-

Another interesting observation is that success rates between models with similar architectures are always 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 from any other architectures.

5.2 Model Capacity and Test Accuracy

We first study this problem on ImageNet dataset. A variety of models are used as source models to perform both FGSM and IGSM attacks against *vgg19_bn*, with results 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 with powerful attack capability concentrate in the bottom left corner, whereas for those models with either large test error or large number of parameters, the fooling rates are much lower. We also tried other models with results shown in Appendix , and the results show no difference.

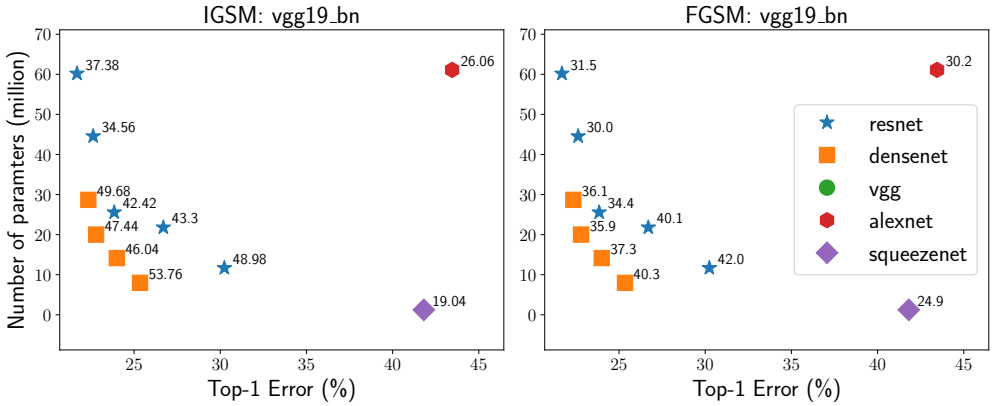


Figure 1: Top-1 success rates of IGSM ($T = 20, \alpha = 5, \varepsilon = 15$) and FGSM($\varepsilon = 15$) attacks against *vgg19_bn* for various models. The annotated value is the success rate transferring to the *vgg_19*. Here, the models of vgg-style have been removed to exclude the influence of architecture similarity. For the same color, the different points corresponds networks of different widths.

We suspect that the impact of test accuracy is due to that the decision boundaries of high-accuracy models are similar, since they all approximate the ground-truth decision boundary of true data very well. On the contrary, the model with low accuracy must has a decision boundary relatively different from the high-accuracy models. Here the targeted network *vgg19_bn* is a high-accuracy model. Therefore it is not surprising to observe that high-accuracy models tend to exhibit stronger attack capability. This phenomenon implies a kind of data-dependent transferability, which is different from architecture-dependent transfers observed in the previous section.

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. This observation suggests us not to use deep models as the source models when performing transfer-based attacks, although we have not fully understand this phenomenon.

Table 2: Each cell (S,T) denotes the Top-1 success rate of attack from the source model (row) to the target model (column). (a) FGSM attack for MNIST with $\varepsilon = 40$ and D denotes the depth of the fully connected network. (b) FGSM attack for CIFAR-10 with $\varepsilon = 10$.

(a) MNIST

	$D=0$	$D=2$	$D=4$	$D=8$
$D=0$	-	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	-

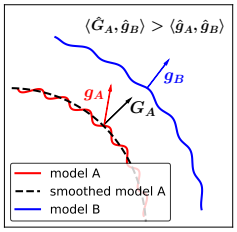
(b) CIFAR-10

	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

6 Non-smoothness of Loss Surface

Smilkov et al. (2017) showed that gradient ∇J could be very noisy and uninformative for visualization, though the model is trained very well. Balduzzi et al. (2017) studied a similar phenomenon that gradients of deep networks are shattered. Both of them implies that the landscape are locally rough . In this section, we study how this non-smoothness affects the transferability.

6.1 Intuition



For simplicity, assume model A and B are the source and target models, respectively and let $g(x) = \nabla_x J(x)$. Previous methods use $g_A(x)$ to generate adversarial perturbations, so the transferability mainly depends on how much sensitivity of g_A for model A can transfer to model B. We hypothesize that the noise hurts the transferability of g_A, as illustrated in right cartoon figure, where three curves denote the level sets of three models. Since both model A and B have very high test accuracy, their level sets should be similar globally, and $J_B(x)$ is probably unstable along g_A. However, as illustrated, 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 gradient G_A, i.e. $\langle \hat{G}_A, \hat{g}_B \rangle > \langle \hat{g}_A, \hat{g}_B \rangle$.

6.2 Justification

To justify the previous arguments, we consider smoothing the loss surface by convolving with a Gaussian, $J_\sigma(x) := \int J(x - x')\phi_\sigma(x')dx'$ with $\phi_\sigma(x)$ being the density of $\mathcal{N}(0, \sigma^2 I)$. The corresponding gradient can be calculated by

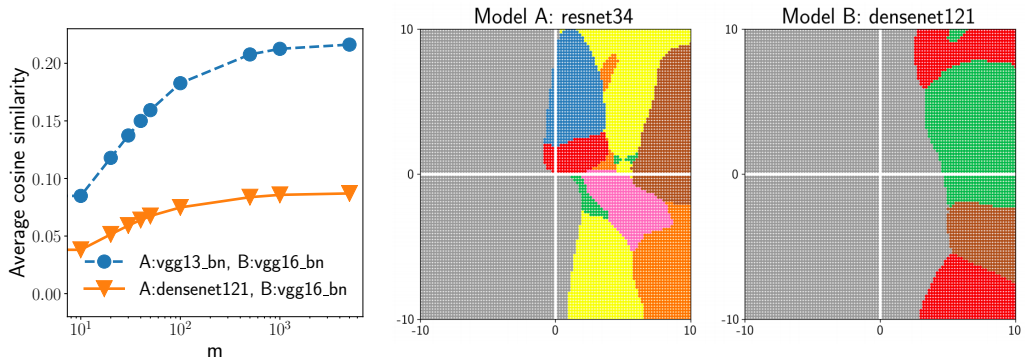
$$G_\sigma(x) = \mathbb{E}_{\xi \sim \mathcal{N}(0,\sigma^2 I)}[g(x + \xi)]. \tag{6}$$

The extent of smoothing is controlled by σ, and please refer to Appendix A to see smoothing effect. We will show that $G_\sigma(x)$ is more transferable than $g(x)$.

Gradient similarity We first quantify the cosine similarity between gradients of source and target models, respectively. Two situations are considered: *vgg13_bn*→*vgg16_bn*, *densenet121*→*vgg13_bn*, which correspond the within-architecture and cross-architecture transfers, respectively. We choose $\sigma = 15$, and the expectation in (6) 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 against the number of samples m. In Figure 2a, as expected, we see that the cosine similarity between G_A and g_B are indeed larger than the one between g_A and g_B. Moreover, the similarity increases with m monotonically, which further justifies that G_A is more transferable than g_A.

Visualization In Figure 2b we visualize the transferability by comparing the decision boundaries of model A (*resnet34*) and model B (*densenet121*). The horizontal axis represents the direction of G_A of *resnet34*, estimated by $m = 1000, \sigma = 15$, and the vertical axis denotes orthogonal direction $h_A := g_A - \langle g_A, \hat{G}_A \rangle \hat{G}_A$. This process means that we decompose the gradient into two terms: $g_A = \alpha G_A + \beta h_A$ with $\langle G_A, h_A \rangle = 0$. 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. The color corresponds to the label predicted by the target model.

It can be easily observed that for model A, a small perturbation in both directions can produce wrong classification. However, when applied to model B, the sensitivities of two directions dramatically change. The direction of h_A becomes extremely stable, whereas to some extent G_A preserves the sensitivity, i.e. G_A do transfer to model B better than h_A. This suggests that for gradient g_A, the noisy part h_A is less transferable than the smooth part G_A. We also tried a variety of other models shown in Appendix C, and the results are the same.



(a) (b)

Figure 2: (a) Cosine similarity between the gradients of source and target models. (b) Visualization of decision boundaries. The origin corresponds to the clean image shown in Figure 9 of Appendix. The same color denotes the small label, and the gray color corresponds to the ground truth label.

7 Smoothed Gradient Attack

7.1 Method

Inspired by the previous investigations, enhancing the adversarial transferability can be achieved by smoothing the loss surface. Then our objective becomes,

$$\begin{aligned} \text{maximize} \quad & J_\sigma(x') := \mathbb{E}_{\xi\sim\mathcal{N}(0,\sigma^2)}[J(x'+\xi)] \\ \text{s.t.} \quad & \|x'-x\| \le \varepsilon. \end{aligned} \tag{7}$$

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

If use the iterative gradient sign method to solve (7), we have the following iteration:

$$\begin{aligned} G_t &= \frac{1}{m}\sum_{i=1}^{m} \nabla J(x_t+\xi_i), \quad \xi_i \sim \mathcal{N}(0,\sigma^2 I) \\ x_{t+1} &= \text{proj}_D\left(x_t + \alpha\, \text{sign}\left(G_t\right)\right), \end{aligned} \tag{8}$$

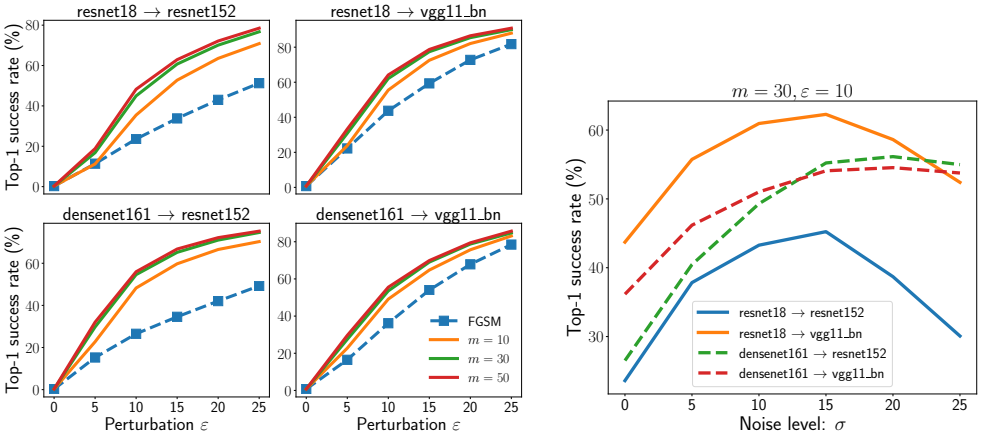
where G_t is a mini-batch approximation of the smoothed gradient (6). Compared to the standard IGSM, the gradient is replaced by a smoothed version, which is endowed with stronger transferability. Therefore we call this method sg-IGSM. The special case $T = 1, \alpha = \varepsilon$, is accordingly called sg-FGSM. Any other optimizer can be used to solve the (7) as well, and we only need to replace the original gradient with the smoothed one.

7.2 Choice of Hyper Parameters

We first explore the sensitivity of hyper parameters m, σ when applying our smoothed gradient technique. We take ImageNet dataset as the testbed, and sg-FGSM attack is examined. To increase the reliability, four attacks are considered here. The results are shown in Figure 3.

We see that sg-FGSM consistently outperforms FGSM for all distortion level, although the improvement varies for different ε. Furthermore, larger m leads to higher success rate due to the better estimation of the smooth part of gradient, and the benefit starts to saturate after $m \ge 30$. For the smoothing factor σ, we find neither too large nor too small value can work well. Overly large σ will introduce a large bias in (8). Extremely small σ is unable to smooth the landscape enough.

Therefore, in the subsequent sections, we will use $m = 20, \sigma = 15$ to estimate the smoothed gradient and only report the result for one distortion level.



(a) (b)

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

7.3 Effectiveness

IGSM attack We first test the effectiveness of our method for single-model based attacks on non-targeted setting. This to say $T = 5$ for sg-IGSM due to $m = 20$. Specifically, we make the number of gradient calculation per sample be fixed 100 for a fair comparison. The results are shown in Table 3. We see that smoothed gradients do enhance the transferability dramatically for all the attacks considered. Please especially note those bold rates, where the improvements have reached about 30%.

Table 3: Top-1 success rates(%) of non-targeted IGSM and sg-IGSM attacks. 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 sg-IGSM ($T = 5, \alpha = 5$). 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 attack In this part, we examine the ensemble-based attack on targeted setting[2]. We test whether the performance can be further improved by employing smoothed gradients. 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. We instead adopt the the Top-5 success rate as our criterion to better reflect the improvement of our method. A variety of model ensembles are examined, and the results are reported in Table 4.

As we can see, it is clear that smoothed gradient attacks outperform the corresponding normal ones by a remarkable large margin. More importantly, the improvement never be harmed compared to single-model case in Table 3, which implies that *smoothed gradient can be effectively combined with ensemble method without compromise*.

Momentum attack In this experiment, three networks of different architectures are selected. As suggested in Dong et al. (2018), we choose $\mu = 1$, and all attacks are iterated for $T = 5$ with step size

[2]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 C for more details 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 4: Top-5 success rates (%) of ensemble-based approaches for IGSM ($T = 20, \alpha = 15$) versus sg-IGSM ($T = 20, \alpha = 15$). The row and column denote the source and target model, respectively.

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

$\alpha = 5$. In Table 5, we report the results of non-targeted attacks of three attacks including mIGSM, sg-IGSM and mIGSM with smoothed gradient (sg-mIGSM). It is shown that our method clearly outperforms momentum-based method for all the cases. Moreover, by combining with the smoothed gradient, the effectiveness of momentum attacks can be further improved significantly.

Table 5: Top-1 success rates (%) of momentum attacks and smoothed gradient attacks. The row and column denote the source and target model, respectively. Each cell contains three rates corresponding to mIGSM, sg-IGSM and sg-mIGSM 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	-

7.4 Robustness

Smoothed gradient attacks can be viewed as generating adversarial examples robust against Gaussian noise perturbation. Therefore, we are interested in how robust is the adversarial example against other image transformations. To quantify the influence of transformations, we use the notion of destruction rate defined by Kurakin et al. (2017). The lower is this rate, the more robust are the adversarial examples.

Densenet121 and *resnet34* are 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 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. We speculate that the robustness can also transfer among different image transforms.

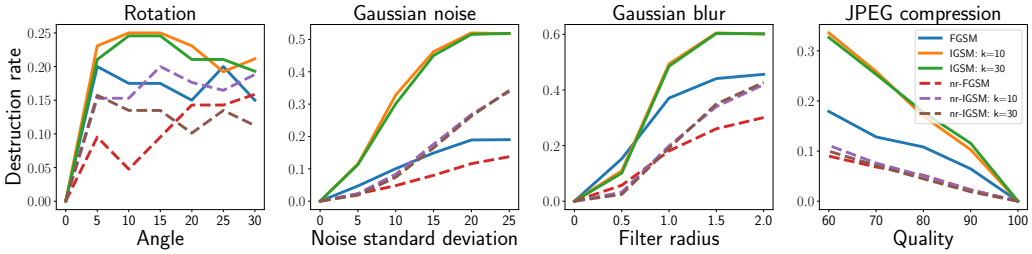


Figure 4: Destruction rates of adversarial examples for various methods. For smoothed gradient attacks, we choose $m = 20, \sigma = 15$. The distortion $\varepsilon = 15$.

8 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 noisy part of gradient hinders the transfer of adversarial examples. Motivated by these understanding, we proposed the smoothed gradient attack which can enhance the transferability of adversarial examples dramatically. Furthermore, the smoothed gradient can be combined with both ensemble and momentum based attacks rather effectively.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 274–283. PMLR, 2018a.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 284–293. PMLR, 2018b.

David Balduzzi, Marcus Frean, Lennox Leary, J. P. Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 342–350. PMLR, 2017.

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

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Fundamental limits on adversarial robustness. In *Proc. ICML, Workshop on Deep Learning*, 2015.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR Workshop*, 2017.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017.

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

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.

Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

Dawn Song Warren He, Bo Li. Decision boundary analysis of adversarial examples. In *International Conference on Learning Representations*, 2018.

A VISUALIZATION OF SMOOTHED GRADIENT

In this section, we provide an visualization understanding how the local average has the smoothing effect. We choose *densenet121* as the model and visualize the saliency map of gradient $\nabla_x J(x)$ and the smoothed versions for varying m. Two images are considered, and the results are shown in Figure 5. We can see that local average can smooth the gradient significantly. Please refer to the work by Smilkov et al. (2017) for more details.

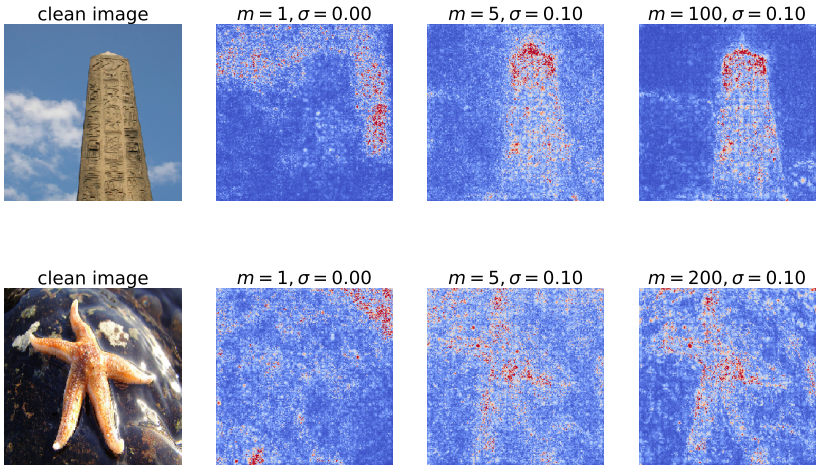


Figure 5: Visualization of gradients. The leftmost is the examined image. The second corresponds to the original gradient, whereas the remaining two images corresponds to the smoothed gradients estimated by different m.

B 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 T and step size α. Here we explore their influence to the quality of adversarial examples generated. The success rates are calculated on $5,000$ images randomly selected according to description of Section 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 4.

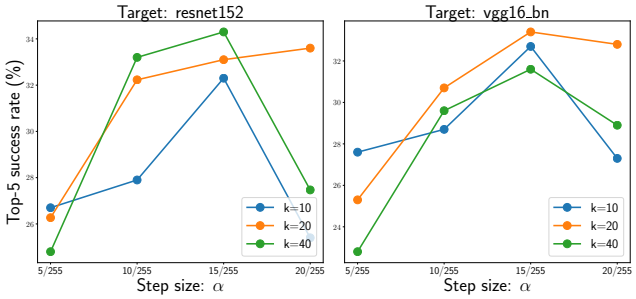


Figure 6: Average success rates over three ensembles for different step size α and number of iteration k. The three ensembles are the same with those in Table 4. Distortion $\varepsilon = 20$.

Figure 6 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 performances. It is worth noting 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.

C ADDITIONAL RESULTS

How Model Capacity and Test Accuracy Affect Transferability In this part, we additionally explore the impact of model capacity and test accuracy by using *resnet152* as our target model and perform transfer-based attacks against from a variety of models and the results shown in Figure 7. The observations are consistent with the observation in Section 5.2.

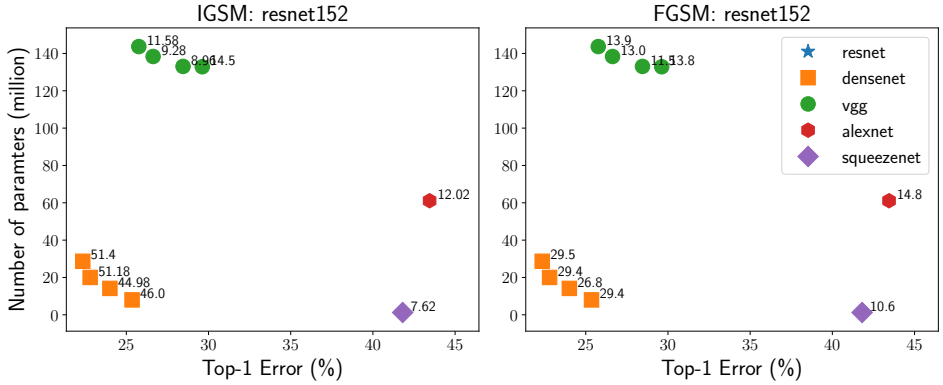


Figure 7: Top-1 success rates of IGSM ($T = 20, \alpha = 5, \varepsilon = 15$) and FGSM($\varepsilon = 15$) attacks against *resnet152* for various models. The annotated value is the success rate transferring to the *resnet152*. Here, the models of resnet-style have been removed to exclude the influence of architecture similarity. For the same color, the different points corresponds networks of different widths.

Visualization of Decision Boundary In this part, we provide additional results on the visualization of the decision boundary. Different Figure 2b, we here consider the sign of two directions, since the attack method actually updates using the sign (g) rather than g. In Figure 8, we show the decision boundary. Each point corresponds to perturbed image:

$$\text{clip}(x + u\text{sign}\left(G_A\right) + v\text{sign}\left(h_A\right), 0, 255).$$

Here the model A is resnet34, and the definitions of G_A and h_A are the same as before. The color denotes the label predicted by the target model, and the gray color corresponds to the ground-truth label. We can see that the smoothed gradient G_A is indeed more transferable than the noise part h_A.

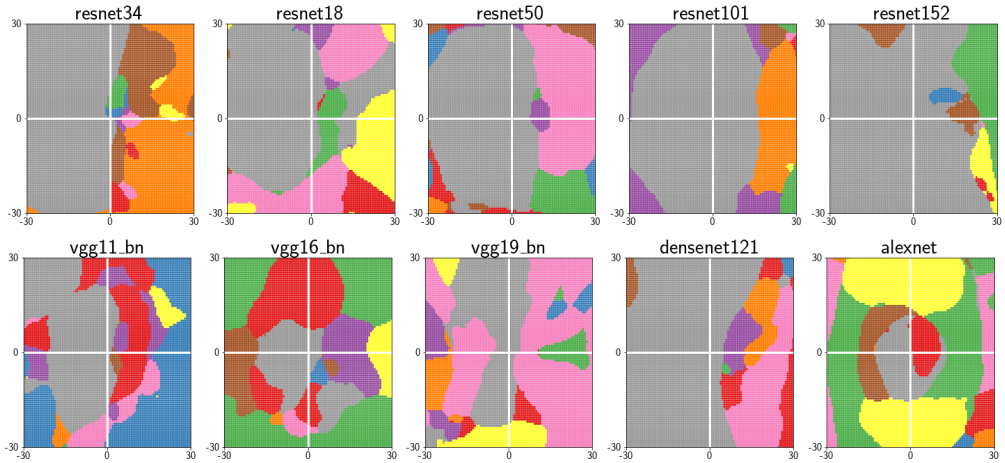


Figure 8: Visualization of decision boundaries. The source model is resnet34, and 9 target models are considered. The horizontal and vertical direction corresponds to u and v, respectively.

Additional Results for the Effectiveness of Smoothed Gradient Attacks In this part, we present some additional results to show the effectiveness of smoothed gradient attack.

Table 6: Top-1 success rates (%) of ensemble-based non-targeted IGSM and sg-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 sg-IGSM ($T = 50, \alpha = 3$). The distortion $\varepsilon = 20$.

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 7: Top-1 success rates (%) of ensemble-based targeted IGSM and sgd-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 sg-IGSM ($T = 20, \alpha = 15$). The distortion $\varepsilon = 20$.

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**

D THE EXAMINED IMAGE FOR VISUALIZATION OF DECISION BOUNDARY

Figure 9: The image used to perform decision boundary visualization. Its ID in ILSVRC2012 validation set is 26, with ground truth label being *table lamp*.