ENHANCING THE TRANSFERABILITY OF ADVERSARIAL EXAMPLES WITH NOISE REDUCED GRADIENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks provide state-of-the-art performance for many applications of interest. Unfortunately they 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 the other unseen models. Consequently the adversary can leverage it to attack against the deployed black-box systems. In this work, we demonstrate that the adversarial perturbation can be decomposed into two components: model-specific and data-dependent one, and the latter mainly contributes to the transferability. Motivated by this understanding, we propose to use the noise reduced gradient that approximates the data-dependent component, instead of the ordinary gradient, to craft adversarial examples. This new approach can enhance the transferability dramatically, and its effectiveness is justified on various classification models over ImageNet. We also find that low-capacity models have more powerful attack capability than high-capacity counterparts, under the condition that they have similar test performance. All of these valuable findings could potentially provide us insights for designing effective defense approaches against black-box attacks.

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 always able to fool the model to produce incorrect outputs by manipulating the inputs maliciously. The corresponding manipulated samples are called *adversarial examples*. However, how to understand this phenomenon (Goodfellow et al. (2014); Tramèr et al. (2017b)) and how to defend against adversarial examples effectively (Kurakin et al. (2016); Tramèr et al. (2017a); Carlini & Wagner (2017)) are still open questions. Meanwhile it is found that adversarial examples can transfer across different models, i.e. the adversarial examples generated from model S can also fool another model T with a high probability. We refer to such property as *transferability*, which can be leveraged to attack a black-box system (Papernot et al. (2016); Liu et al. (2016)).

The phenomenon of adversarial vulnerability was first introduced and studied in Szegedy et al. (2013). The authors modeled the adversarial example generation as an optimization problem solved by box-constraint L-BFGS, and also attributed the presence of adversarial examples to the strong nonlinearity of deep neural networks. Goodfellow et al. (2014) argued instead that the primary cause of the adversarial instability is the linear nature and the high dimensionality, and the view yielded the fast gradient sign method (FGSM). Similarly based on an iterative linearization of the classifier, Moosavi-Dezfooli et al. (2015) proposed the DeepFool method. In Kurakin et al. (2016); Tramèr et al. (2017a), it was shown that the iterative gradient sign method provides stronger white-box attacks, however, does not work well for black-box attacks. Liu et al. (2016) analyzed the transferability of adversarial examples in detail and proposed *ensemble-based approaches* for effective black-box attacks. In Carlini & Wagner (2017) it was demonstrated that high-confidence adversarial examples that are strongly misclassified by the original model, have stronger transferability.

In addition to crafting adversarial examples for attack, there exist lots of works on devising more effective defense. Papernot et al. (2015) proposed the defensive distillation. Goodfellow et al.

(2014) introduced the adversarial training method, which was examined on ImageNet by Kurakin et al. (2016) and Tramèr et al. (2017a). Lu et al. (2017) utilized image transformation, such as rotation, translation, and scaling, etc, to alleviate the harm of the adversarial perturbation. Instead of making the classifier itself more robust, several works (Li & Li (2016); Feinman et al. (2017)) attempted to detect the adversarial examples, followed by certain manual processing. Unfortunately, all of them can be easily broken by designing stronger and more robust adversarial examples (Carlini & Wagner (2017); Athalye & Sutskever (2017)).

In this work, we explore the possibility to understand and enhance the black-box attacks. We decompose the adversarial perturbation into two components: model-specific and data-dependent one. The model-specific component comes from the model architecture and random initialization, which is noisy and represents the behavior off the data manifold. In contrast, the data-dependent component is smooth and approximates the ground truth along the data manifold. We argue that it is the data-dependent part that mainly contributes the transferability of adversarial perturbations across different models. Based on this view, we propose to construct adversarial examples by employing the data-dependent component of gradient instead of the gradient itself. Since this component is estimated via noise reduction strategy, we call it *noise-reduced gradient (NRG)* method. Its effectiveness is demonstrated by combing it with several methods to perform black-box attacks over ImageNet validation set.

We also explore the relationship between complexity, accuracy and success rate of black-box attacks. It is found that the models with higher accuracy and lower capacity show stronger capability to attack unseen models. Moreover this phenomenon can be explained by our understanding of transferability, and may provide us some guides to attack the remote unseen model.

2 Background

2.1 Adversarial examples

Let us use $f(\boldsymbol{x}) \in \mathbb{R}^d \to \mathbb{R}^K$ denote the model function, which is obtained via minimizing the empirical risk over training set. For simplicity we omit the dependence on the parameter $\boldsymbol{\theta}$, since it is fixed in this paper. For many application of interest, we always have $d \gg 1$ and $K = o(1)$. According to the local linear analysis in Goodfellow et al. (2014), the high dimensionality makes $f(\boldsymbol{x})$ inevitably vulnerable to the adversarial perturbation. That is to say, for each $\boldsymbol{x}$, there exists a small perturbation $\boldsymbol{\eta}$ that is nearly imperceptible by human, such that

$$f(\boldsymbol{x}) = y^{\text{true}} \quad f(\boldsymbol{x} + \boldsymbol{\eta}) \neq y^{\text{true}} \tag{1}$$

We call $\boldsymbol{\eta}$ adversarial perturbation and correspondingly $x^{adv} := \boldsymbol{x} + \boldsymbol{\eta}$ adversarial example. In this work, we mainly study the adversarial examples in the context of deep neural networks, though they also exist in other models, for example, support vector machine (SVM) and decision tree, etc (Papernot et al. (2016)).

Targeted attack This attack aims at fooling the model such that it outputs a not only wrong but also target label that is specified by the adversary, i.e.

$$f(\boldsymbol{x} + \boldsymbol{\eta}) = y^{\text{target}}.$$

In contrast, we call the attack (1) a non-targeted attack.

Black-box attack In this setting, the adversary would like to fool a target model, whose architecture and parameters are not available. Also it is prohibited to query the model, i.e. the target 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.

2.2 Modelling of generating adversarial examples

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

$$\begin{aligned} \text{maximize} \quad & J(f(\boldsymbol{x} + \boldsymbol{\eta}), y^{\text{true}}) \\ \text{s.t.} \quad & \|\boldsymbol{\eta}\| \leq \varepsilon, \end{aligned} \tag{2}$$

where J is a loss function measuring the discrepancy between the prediction and ground truth; $\|\cdot\|$ is the metric to quantify the magnitude of the perturbation. For image data, there is also an implicit constraint: $\boldsymbol{x}^{adv} \in [0, 255]^d$, with d being the number of pixels. In practice, the commonly choice of J is the *cross entropy*. Carlini & Wagner (2016) introduced a loss function that directly manipulates the output logit instead of probability. This loss function is also adopted by many works. As for the measurement of distortion, The best metric should be human-eye, unfortunately which is not available for us. In practice ℓ_∞ and ℓ_2 are the two norms that are commonly used.

Ensemble-based approaches To improve the strength of adversarial examples, instead of using a single model Liu et al. (2016) suggest to use a large ensemble consisting of $f_1, f_2, \cdots, f_Q$ as our source model. We call this ensemble-based approaches, and the corresponding objective can be written as

$$\begin{aligned} \text{maximize} \quad & J\left(\sum_{i=1}^{Q} w_i f_i(\boldsymbol{x}+\boldsymbol{\eta}), y^{\text{true}}\right) \\ \text{s.t.} \quad & \|\boldsymbol{\eta}\| \leq \varepsilon \end{aligned} \tag{3}$$

where w_i are the ensemble weights, $\sum w_i = 1$.

Objectives (2) and (3) are for non-targeted attacks, and the targeted counterpart can be derived similarly.

2.3 Optimizer

There are various optimizers to solve problem (2) and (3). In this paper, we mainly use the normalized-gradient based optimizer.

Fast Gradient Based Method This method (Goodfellow et al. (2014)) attempts to solve (2) by performing only one step iteration,

$$\boldsymbol{x}^{adv} \leftarrow \boldsymbol{x} + \varepsilon\, g(\boldsymbol{x}), \tag{4}$$

where $g(\boldsymbol{x})$ is a normalized gradient vector. For ℓ_∞-attack ,

$$g^\infty(\boldsymbol{x}) = \text{sign}\left(\nabla_x J(f(\boldsymbol{x}); y^{\text{true}})\right) \quad (\textbf{FGSM}); \tag{5}$$

similarly for ℓ_q-attack,

$$g^q(\boldsymbol{x}) = \frac{\nabla_x J(f(\boldsymbol{x}); y^{\text{true}})}{\|\nabla_x J(f(\boldsymbol{x}); y^{\text{true}})\|_q} \quad (\textbf{FGM}). \tag{6}$$

Both of them are called fast gradient based method (FGBM). The key property of FGBM is the efficiency, that is why it is called *fast*. Also it has very good transferability (Kurakin et al. (2016); Tramèr et al. (2017a)) though not optimal. So it is worth considering this simple yet effective optimizer.

Iterative Gradient Method This method (Madry et al. (2017); Kurakin et al. (2016)) performs the projected normalized-gradient ascent

$$\boldsymbol{x}^{t+1} \leftarrow \text{clip}^{\boldsymbol{x}^0, \varepsilon}\left(\boldsymbol{x}^t + \alpha\, g(\boldsymbol{x}^t)\right) \tag{7}$$

for k steps, where $\boldsymbol{x}^0$ is the original clean image; $\text{clip}^{\boldsymbol{x}, \varepsilon}(\cdot)$ is the projection operator to enforce $\boldsymbol{x}^t \in [0, 255]^d \cap \{\boldsymbol{x} \mid \|\boldsymbol{x} - \boldsymbol{x}^0\| \leq \varepsilon\}$ and α is the step size. In analogous to the fast gradient based method, the normalized gradient $g(\boldsymbol{x})$ is chosen as $g^\infty(\boldsymbol{x})$ for ℓ_∞-attack, called iterative gradient sign method (IGSM), and $g^q(\boldsymbol{x})$ for ℓ_q-attack. The fast gradient based method (4) can be viewed as a special case of (7) when $\alpha = \varepsilon, k = 1$.

3 Enhancing the Transferability of Adversarial Examples

3.1 Data-dependent components enable transferability

For now there are few articles to understand why adversarial examples can transfer between models, though it is extremely important for performing black-box attacks and building successful defenses.

To the best of our knowledge, the only two works are (Liu et al. (2016); Tramèr et al. (2017b)), which suggested that the transferability comes from the similarity between the decision boundaries of the source and target models, especially in the direction of transferable adversarial examples. Tramèr et al. (2017b) also claimed that transferable adversarial examples span a contiguous subspace.

To investigate the transfer phenomenon, we ask a further question: *what similarity between the different models A and B that makes the adversarial examples capable of transferring across them?* Since the model A and B have a high performance on the same dataset, they must have learned a similar function *along the data manifold*. However, the behaviour of the models *off the data manifold* can be different. This is determined by the architectures of the models and random initialization before learning, which are data-independent factors. Suppose now we have an adversarial perturbation for model A that can transfer to B. Then we decompose this perturbation into two parts along and off the data manifold, referred to as *data-dependent* and *model-specific* components. We hypothesize that the component along the data manifold mainly contributes to the transfer from A to B, since this component captures the data-dependent information shared between models. The model-specific component contributes little to the transferability due to its different behaviours off the data manifold.

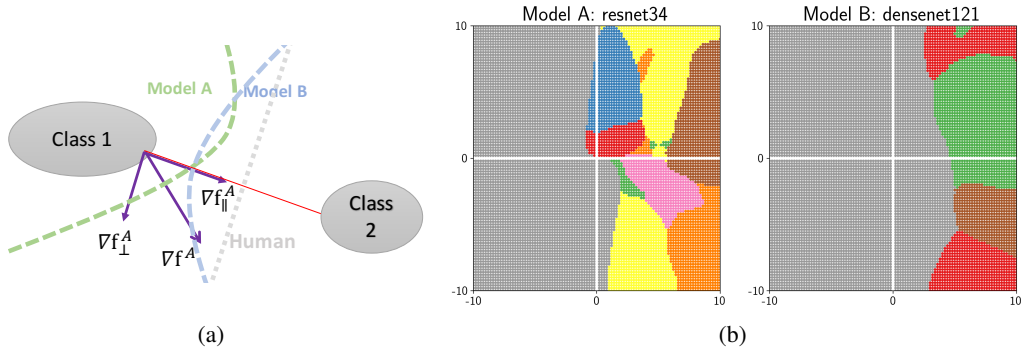


Figure 1: (a) The illustration of the decomposition and transferability of the adversarial perturbation ∇f^A. (b) Decision boundaries visualization. Each point represents image: $\boldsymbol{x} + u\,\nabla f_{\parallel}^A/\|\nabla f_{\parallel}^A\|_2 + v\,\nabla f_{\perp}^A/\|\nabla f_{\perp}^A\|_2$, and the same color indicates the same predicted label. More details can be found in Section 6.3.

Take $\boldsymbol{\eta} = \nabla f^A$ as the adversarial perturbation crafted from model A, we illustrate this explanation in Figure 1. In the left panel, the decision boundaries of two models are similar in the inter-class area. As can be observed, ∇f^A can mislead both model A and B. Then we decompose the perturbation into two parts, a data-dependent component $\nabla f_{\parallel}^A$ and a model-specific one $\nabla f_{\perp}^A$, respectively. Since $\nabla f_{\parallel}^A$ is almost aligned to the inter-class deviation (red line), i.e. along the data manifold, it can attack model B easily. However, The model-specific $\nabla f_{\perp}^A$ contributes little to the transfer from A to B, though it can successfully fool model A with a very small distortion.

In the right panel, we plot the decision boundaries of *resnet34* (model A) and *densenet121* (model B) for ImageNet. The horizontal axis represents the direction of data-dependent component $\nabla f_{\parallel}^A$ estimated by our proposed NRG method (8) from *resnet34*; and the vertical axis depicts the direction of model-specific component $\nabla f_{\perp}^A$. It is easily observed that for model A *resnet34*, a small perturbation in both directions can produce wrong classification. However, when applied to model B *densenet121*, a large perturbation along the $\nabla f_{\perp}^A$ direction cannot change the classification results, while a small perturbation along the data-dependent one can change the prediction easily.

This understanding motivates the our NRG method for enhancing the black-box adversarial attacks by reducing the model-specific noise, as elaborated in the following.

3.2 Noise-reduced gradient methods

Noise-reduced gradient (NRG) The model-specific component $\nabla f_{\perp}$ inherits from the random initialization, so it must be very noisy, i.e. high-frequency (the same observation is also systemat-

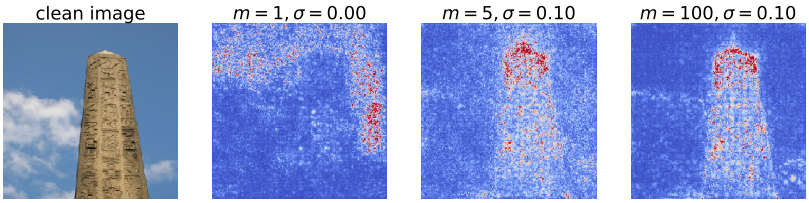


Figure 2: Visualization of the gradient with respect to input for *resnet152*. The leftmost is the original image, and the rest three images are the visualization of the gradients.

ically investigated in Balduzzi et al. (2017)); while $\nabla f_{\|}$ is smooth, i.e. low-frequency, since it encodes the knowledge learned from training data. Therefore, *local average* can be applied to remove the noisy model-specific information, yielding an approximation of the data-dependent component,

$$\nabla f_{\|} \approx \frac{1}{m} \sum_{i=1}^{m} \nabla f(\boldsymbol{x} + \boldsymbol{\xi}_i), \quad \boldsymbol{\xi}_i \sim \mathcal{N}(0, \sigma^2 I_d), \tag{8}$$

where the m is the number of samples chosen for averaging. We call (8) *noise-reduced gradient* (NRG), which captures more data-dependent information than ordinary gradient ∇f. To justify the effectiveness of this method, similar to Smilkov et al. (2017), we visualize NRG for various m in Figure 2. As shown, larger m leads to smoother and more data-dependent gradient, especially for $m = 100$ the NRG can even capture the cloud behind the obelisk.

The noisy model-specific information of ∇f could mislead the optimizer into the solutions that are overfitting to the specific source model, thus generalizing poorly to the other model. Therefore we propose to perform attacks by using $\nabla f_{\|}$ in Eq.(8), instead of ∇f, which can drive the optimizer towards the solutions that are more data-dependent.

Noise-reduced Iterative Sign Gradient Method (nr-IGSM) The iterative gradient sign method mounted by NRG can be written as,

$$\begin{aligned} G^t &= \frac{1}{m} \sum_{i=1}^{m} \nabla J(\boldsymbol{x}^t + \boldsymbol{\xi}_i), \quad \boldsymbol{\xi}_i \sim \mathcal{N}(0, \sigma^2 I_d) \\ \boldsymbol{x}^{t+1} &= \text{clip}^{\boldsymbol{x}^0, \varepsilon} \left(\boldsymbol{x}^t + \alpha \, \text{sign} \left(G^t \right) \right), \end{aligned} \tag{9}$$

The special case $k = 1, \alpha = \varepsilon$ are called noise-reduced fast gradient sign method (nr-FGSM) accordingly. For ℓ_q-attack, the noise-reduced version is similar, replacing the second equation of (9) by

$$\boldsymbol{x}^{t+1} = \text{clip}^{\boldsymbol{x}^0, \varepsilon} \left(\boldsymbol{x}^t + \alpha \, G^t / \|G^t\|_q \right). \tag{10}$$

For any general optimizer, the corresponding noise-reduced counterpart can be derived similarly.

4 Experiment Setting

To justify and analyze the effectiveness of NRG for enhancing the transferability, we use the start-of-the-art classification models over ImageNet dataset. We elaborate the details in the following.

Dataset We use the ImageNet ILSVRC2012 validation set that contains $50,000$ samples. For each attack experiment, we randomly select $5,000$ images that can be correctly recognized by all the models, since it is meaningless to construct adversarial perturbations for the images that target models cannot classify correctly. And for the targeted attack experiments, each image is specified by a randomly wrong label.

Models We use the pre-trained models 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]. For increasing the reliability of experiments, all the models have been used, however for a specific experiment we only choose several of them to save computational time.

Criteria Given a set of adversarial examples, $\{(\boldsymbol{x}_1^{adv}, y_1^{\text{true}}), (\boldsymbol{x}_2^{adv}, y_2^{\text{true}}), \ldots, (\boldsymbol{x}_N^{adv}, y_3^{\text{true}})\}$, We calculate their *Top-1 success rate* fooling a given model $F(\boldsymbol{x})$ by

$$\frac{1}{N}\sum_{i=1}^{N} 1[F(\boldsymbol{x}_i^{adv}) \neq y_i^{\text{true}}]. \tag{11}$$

If F is the model used to generate adversarial examples, then the rate indicates the the white-box attack performance. For targeted attacks, Each image $\boldsymbol{x}^{adv}$ 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*,

$$\frac{1}{N}\sum_{i=1}^{N} 1[F(\boldsymbol{x}_i^{adv}) = y_i^{\text{target}}]. \tag{12}$$

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 two distances: ℓ_∞ norm and scaled ℓ_2 norm, i.e. *root mean square deviation* (RMSD) $\sqrt{\sum_{i=1}^{d} \eta_i^2 / d}$, where d is the dimensionality of inputs. As for optimizers, both FGSM and IGSM are considered.

5 Effectiveness of Noise-Reduced Methods

In this section we demonstrate the effectiveness of our noise-reduced gradient technique by combining it with several commonly-used methods.

Table 1: Top-1 success rates of non-targeted FGSM and nr-FGSM attacks between pairs of models. The cell (S, T) indicates the percentage of adversarial examples generated for model S (row) that successfully fool model T (column). For each cell, the left is the success rate of FGSM; the right is the one of nr-FGSM. In this experiment, distortion $\varepsilon = 15$.

	alexnet	densenet121	resnet152	resnet34	vgg13_bn	vgg19_bn
alexnet	98.3 / 98.1	24.7 / 46.0	**18.0 / 19.3**	34.3 / 37.8	48.6 / 50.2	33.7 / 43.7
densenet121	58.2 / 62.1	91.2 / 98.8	34.4 / **66.7**	46.2 / **74.5**	53.0 / **72.5**	44.9 / **71.4**
resnet152	53.2 / 56.9	39.2 / 67.2	81.4 / 95.9	45.4 / 71.3	43.3 / 62.4	36.8 / 61.5
resnet34	57.3 / **62.5**	46.3 / **71.1**	38.4 / 66.4	94.3 / 98.5	54.4 / 70.5	46.7 / 68.8
vgg13_bn	46.8 / 53.9	23.0 / 48.4	16.0 / 34.4	28.0 / 53.2	96.7 / 98.7	54.2 / 84.6
vgg19_bn	47.4 / 54.6	28.1 / 58.8	18.7 / 46.4	31.5 / 62.3	62.2 / 87.1	91.1 / 98.3

5.1 Single-Model Based Approaches

Fast gradient based method We first examine the combination of noise reduced gradient and fast gradient based methods. The success rates of FGSM and nr-FGSM are summarized in Table 1 (results of FGM ℓ_2-attacks can be found in Appendix C). We observe that, for any pair of black-box attack, nr-FGSM performs better than original FGSM consistently and dramatically. Even the white-box attacks (the diagonal cells) also have improvements. This result implies that the direction of noise-reduced gradient is indeed more effective than the vanilla gradient for enhancing the transferability.

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

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

Table 2: Top-1 success rates of non-targeted IGSM and nr-IGSM attacks between pairs of models. The cell (S,T) indicates the percentage of adversarial examples generated for model S (row) that successfully fool model T (column). For each cell: (1) the left is the success rate of IGSM ($k = 100, \alpha = 1$); (2) the right is the one of nr-IGSM ($m = 20, \sigma = 15, k = 5, \alpha = 5$). In this experiment, distortion $\varepsilon = 15$.

	alexnet	densenet121	resnet152	resnet34	vgg13_bn	vgg19_bn
alexnet	100 / 100	26.9 / 24.6	18.3 / 16.0	38.6 / 37.1	49.2 / 47.3	35.9 / 34.4
densenet121	30.6 / **46.8**	100 / 99.9	50.1 / **80.6**	59.9 / **87.2**	62.2 / **82.2**	56.5 / **84.3**
resnet152	27.4 / 40.7	**52.5 / 81.3**	100 / 100	57.2 / 85.6	47.7 / 71.1	42.9 / 72.6
resnet34	29.7 / 44.5	51.5 / 76.4	46.5 / 73.1	100 / 100	53.8 / 74.8	49.1 / 74.5
vgg13_bn	28.4 / 41.2	24.1 / 49.2	14.3 / 33.5	25.1 / 54.1	100 / 100	90.6 / **96.4**
vgg19_bn	24.9 / 39.2	27.1 / 57.5	16.7 / 41.6	27.6 / 60.7	92.0 / **96.1**	99.9 / 99.9

Iterative gradient sign method One may argue that the above comparison is unfair, since nr-FGSM consumes more computational cost than FGSM, determined by m: number of perturbed inputs used for local average. Here we examine IGSM and nr-IGSM under the same number of gradient calculations. Table 2 presents the results. Except for the attacks from *alexnet*, as we expect, adversarial examples generated by nr-IGSM indeed transfer much more easily than those generated by IGSM. This indicates that the noise-reduced gradient (NRG) does guide the optimizer to explore the more data-dependent solutions.

Some observations By comparing Table 1 with Table 2, we find that large models are more robust to adversarial transfer than small models, for example the *resnet152*. This phenomenon has also been extensively investigated by Madry et al. (2017). It also shows that the transfer among the models with similar architectures is much easier, such as the $90\%+$ transferability between vgg-style networks in Table 2. This implies that the model-specific component also contributes to the transfer across models with similar architectures.

Additionally in most cases, IGSM generates stronger adversarial examples than FGSM except the attacks against *alexnet*. This contradicts the claims in Kurakin et al. (2016) and Tramèr et al. (2017a) that adversarial examples generated by FGSM transfer more easily than the ones of IGSM. However our observation is consistent with the conclusions of Carlini & Wagner (2016): the higher confidence (smaller cross entropy) adversarial examples have in the source model, the more likely they will transfer to the target model. We conjecture that this is due to the inappropriate choice of hyper-parameters, for instance $\alpha = 1, k = \min(\varepsilon + 4, 1.24\varepsilon)$ in Kurakin et al. (2016), which are too small resulting in that the solution has not fit the source model enough (i.e. underfitting). When treated as a target model to be attacked, the *alexnet* is significantly different from those source models in terms of both architecture and test accuracy. And therefore, the multiple iterations cause the IGSM to overfit more than FGSM, producing a lowering fooling rate.

These phenomena clearly indicate that we should not trust the objective in Eq. (2) completely, which might cause the solution overfitted to the source model-specific information, thus leading to poor transferability. Our noise reduced gradient technique regularizes the optimizer by removing the model-specific information from the original gradients, and consequently, it can converge to a more data-dependent solution that has much better cross-model generalization capability.

5.2 Ensemble Based Approaches

In this part, we apply NRG method into ensemble-based approaches described in Eq. (3). Due to the high computational cost of model ensembles, we select $1,000$ images, instead of $5,000$, as our evaluation set.

For non-targeted attacks, both FGSM, IGSM and their noise reduced versions are tested. The Top-1 success rates of IGSM attacks are nearly saturated, so we report the corresponding Top-5 rates in Table 3(a) to demonstrate the improvements of our methods more clearly. The results of FGSM and nr-FGSM attacks can be found in Appendix C.

For targeted attacks, to generate an adversarial example predicted by unseen target models as a specific label, is much harder than generating non-targeted examples. Liu et al. (2016) demonstrated that single-model based approaches are very ineffective in generating targeted adversarial examples. That is why we did not consider targeted attacks in Section 5.1.

Different from the non-targeted attacks, we find that targeted adversarial examples are sensitive to the step size α used in the optimization procedures (6) and (9). After trying lots of α, we find that **a large step size is necessary for generating strong targeted adversarial examples**. Readers can refer to Appendix A for more detailed analysis on this issue, though we still can not fully understand it. Therefore we use a much larger step size compared to the non-target attacks. The Top-5 success rates are reported in Table 3(b).

By comparing success rates of normal methods and our proposed NRG methods in Table 3 for both targeted and non-targeted attacks, we observed that NRG methods outperform the corresponding normal methods by a remarkable large margin in this scenario.

Table 3: Top-5 success rates of ensemble-based approaches. The cell (S, T) indicates the attack performances from the ensemble S (row) against the target model T (column). For each cell: the left is the rate of normal method, in contrast the right is the one of the noise-reduced counterpart. The corresponding Top-1 success rates can be found in Appendix C (Table 7 and Table 8).

(a) **Non-targeted attacks**: IGSM ($k = 200, \alpha = 1$) versus nr-IGSM ($k = 10, \alpha = 5, m = 20, \sigma = 15$), 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 ($k = 20, \alpha = 15$) versus nr-IGSM ($k = 20, \alpha = 15, m = 20, \sigma = 15$), 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**

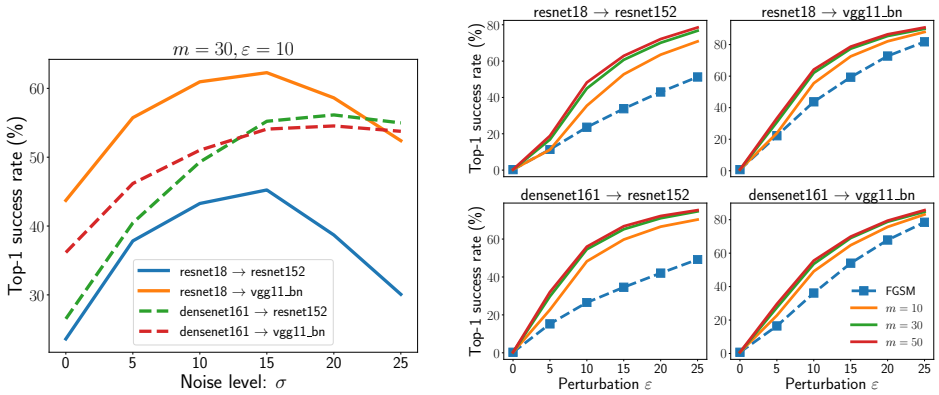
6 Analysis

6.1 Influence of hyper parameters

In this part, we explore the sensitivity of hyper parameters m, σ when applying our NRG methods for black-box attacks. We take nr-FGSM approach as an testbed over the selected evaluation set described in Section 4. Four attacks are considered here, and the results are shown in Figure 3. It is not surprising that larger m leads to higher fooling 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 (8). Extremely small σ is unable to remove the noisy model-specific information effectively, since noisy components of gradients of different perturbed inputs are still strongly correlated for small σ. Moreover the optimal σ varies for different source models, and in this experiment it is about 15 for *resnet18*, compared to 20 for *densenet161*.

6.2 Robustness analysis of adversarial examples

In this part, we preliminarily explore the robustness of adversarial perturbations to image transformations, such as rotation, scaling, blurring, etc. This property is particularly important in practice,



(a) (b)

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

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 in 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 $T(\cdot)$.

Densenet121 and *resnet34* are chosen as our source and target model, respectively; and four image transformations are considered: rotation, Gaussian noise, Gaussian blurring and JPEG compression. Figure 4 displays the results.

The experiment results show that adversarial examples generated by our proposed NRG methods are much more robust than those generated by vanilla methods.

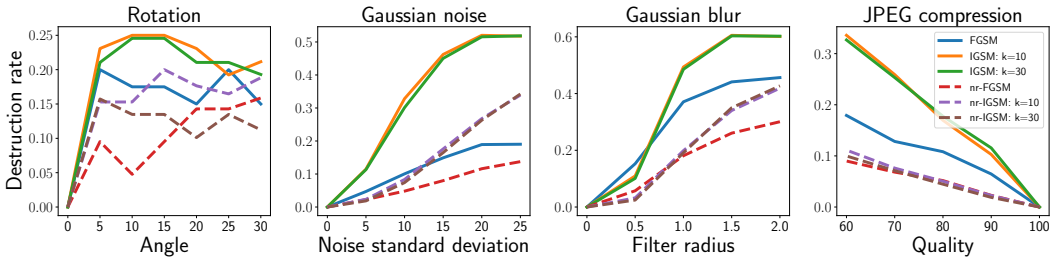


Figure 4: Destruction rates of adversarial examples for various methods. For NRG methods, we choose $m = 20, \sigma = 15$. The distortion ε is set to 15.

6.3 Visualization of decision boundary

In this section we study the decision boundaries of different models to help us understand why NRG-based methods perform better. *Resnet34* is chosen as the source model and nine target models

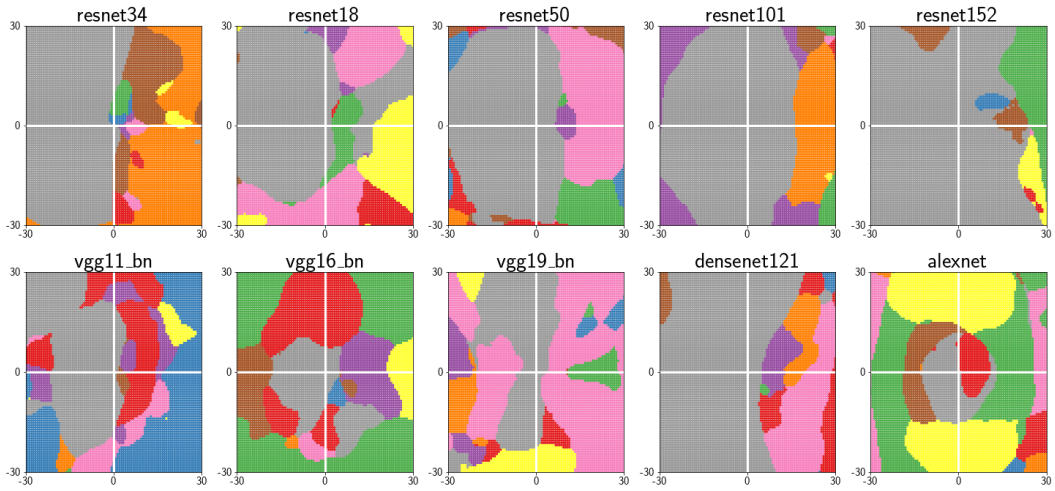


Figure 5: Decision boundaries of various models. The first one is the source model *resnet34* and the others are the target models selected. The horizontal axis represents sign $\left(\nabla f_{\|}\right)$ of *resnet34*, and the vertical axis indicates sign $\left(\nabla f_{\perp}\right)$. The same color indicates the same predicted label. The clean image $\boldsymbol{x}$ used here is *table lamp*, whose ID is 26 in ILSVRC2012 validation set (also shown in Figure 8 of Appendix C).

are considered, including *resnet18, resnet50, resnet101, resnet152, vgg11_bn, vgg16_bn, vgg19_bn, densenet121, alexnet*. The $\nabla f_{\|}$ is estimated by (8) with $m = 1000, \sigma = 15$. Each point (u, v) in this 2-D plane corresponds to the image perturbed by u and v along $\nabla f_{\|}$ and $\nabla f_{\perp}$ direction, respectively, i.e.

$$\text{clip}\left(\boldsymbol{x} + u\,\text{sign}\left(\nabla f_{\|}\right) + v\,\text{sign}\left(\nabla f_{\perp}\right), 0, 255\right) \tag{13}$$

We randomly select one image that can be recognized by all the models examined. The Figure 5 shows the decision boundaries. We also tried a variety of other source models and images, all the plots are similar.

From the aspect of fooling the classifier, the direction of sign $\left(\nabla f_{\|}\right)$ is as sensitive as the direction of sign $\left(\nabla f_{\perp}\right)$ for the source model *resnet34*. However, except *alexnet* all the other target models are much more sensitive along sign $\left(\nabla f_{\|}\right)$ than sign $\left(\nabla f_{\perp}\right)$. This is consistent with the argument in Section 3.1. Removing $\nabla f_{\perp}$ from gradients can also be thought as penalizing the optimizer along the model-specific direction, thus avoiding converging to a source model-overfitting solution that generalizes poorly to the other target models.

Moreover, we find that, along the sign $\left(\nabla f_{\|}\right)$, the minimal distance u to produce adversarial transfer varies for different models. The distances for complex models are significantly larger than those of small models, for instance, the comparison between *resnet152* and *resnent50*. This provides us a geometric understanding of why big models are more robust than small models, as observed in Section 5.1.

6.4 Influence of model capacity and accuracy on attack capability

In earlier experiments (Table 1 and Table 2), we can observe that adversarial examples crafted from *alexnet* generalize worst across models, for example nr-FGSM attack of *alexnet* $\rightarrow$ *resnet152* only achieves 19.3 percent. However, attacks from *densenet121* consistently perform well for any target model, for example 84.3% of nr-IGSM adversarial examples can transfer to *vgg19_bn*, whose architecture is completely different from *densenet121*. This observation indicates that different models can exhibit different performances in attacking the same target model. Now we attempt to find the principle behind this important phenomenon, since the principle can guide us to choose a better local model to generate adversarial examples for attacking the remote black-box system.

We select *vgg19_bn* and *resnet152* as our target model, and use a variety of models to perform both FGSM and IGSM attacks against them. The results are summarized in Figure 6. 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, while fooling rates are very low for those models with large test error and number of parameters[3]. We can obtain an important observation that

the smaller test error and lower capacity a model has, the stronger its attack capability is.

Here we will attempt to explain this phenomenon from our understanding of transferability in Section 3.1. A smaller test error indicates a lower bias for approximating the ground truth along the data manifold. On the other hand, a less complex model might lead to a smaller model-specific component $\nabla f_\perp$, facilitating the data-dependent factor dominate. In a nutshell, the model with small $\nabla f_\perp$ and large $\nabla f_\parallel$ can provide strong adversarial examples that transfer more easily. This is consistent with our arguments presented previously.

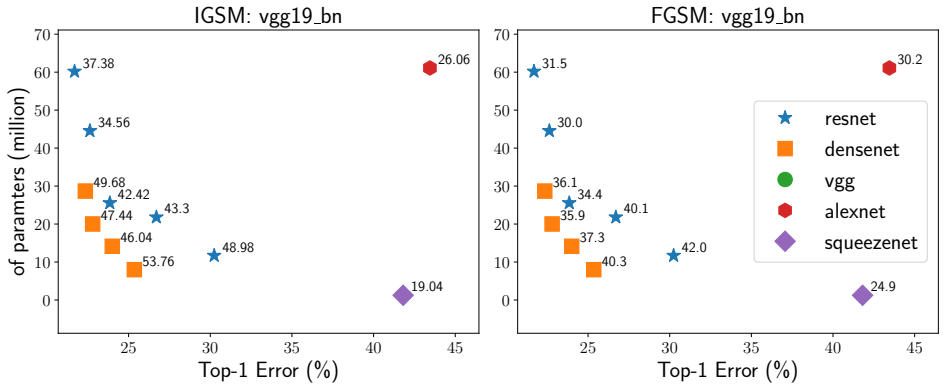


Figure 6: Top-1 success rates of FGSM and IGSM ($k = 20, \alpha = 5$) attacks against *vgg19_bn* for various models. The annotated value is the percentage of adversarial examples that can transfer to the *vgg_19*. Here, the models of vgg-style have been removed, since the contribution from architecture similarities is not in our consideration. The distortion is chosen as $\varepsilon = 15$. The plots of attacking *resnet152* are similar and can be found in Appendix C (Figure 9).

7 Conclusion

In this paper, we have verified that an adversarial perturbation can be decomposed into two components: model-specific and data-dependent ones. And the latter mainly contributes to the cross-model generalization, i.e. adversarial transferability. From this understanding, we proposed the noise-reduced gradient (NRG) based methods to craft adversarial examples, which are much more effective than the previous methods. We also show that the models with lower complexity and higher test accuracy are endowed with stronger capability for black-box attacks.

In the future, we will consider combining NRG-based methods with adversarial training to defend against black-box attacks. The component contributing to the transferability is data-dependent, which is intrinsically low-dimensional, so we hypothesize that black-box attacks can be defensible. On the contrary, the white-box attack origins from the extremely high-dimensional ambient space, thus its defense is much more difficult. Another interesting thread of future works is to learn stable features beneficial for transfer learning by incorporating our NRG strategy, since the reduction of model-specific noise can obtain more accurate information along the data manifold.

[3]Note that the number of model parameters is only an approximate measure of model capacity. For the models with the same number of parameters but different architectures, their capacities might be different. To further investigate the relationship between capacity and attack capability, we fix one particular architecture, e.g. fully connected networks for MNIST, and compare this architecture with different layers, as shown in Appendix B. And the observation is still consistent with that on ImageNet data.

REFERENCES

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

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.

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

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *arXiv preprint arXiv:1705.07263*, 2017.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

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.

Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. *arXiv preprint arXiv:1612.07767*, 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, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *arXiv preprint arXiv:1511.04599*, 2015.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1511.04508*, 2015.

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*, 2016.

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.

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 valuated are calculated on $1,000$ 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 3(b).

Figure 7 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 interesting noting that with small step size $\alpha = 5$, the large number of iteration provides worser performance than small number of iteration. One possible explain is that more iterations lead optimizers to converge to more overfit 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.

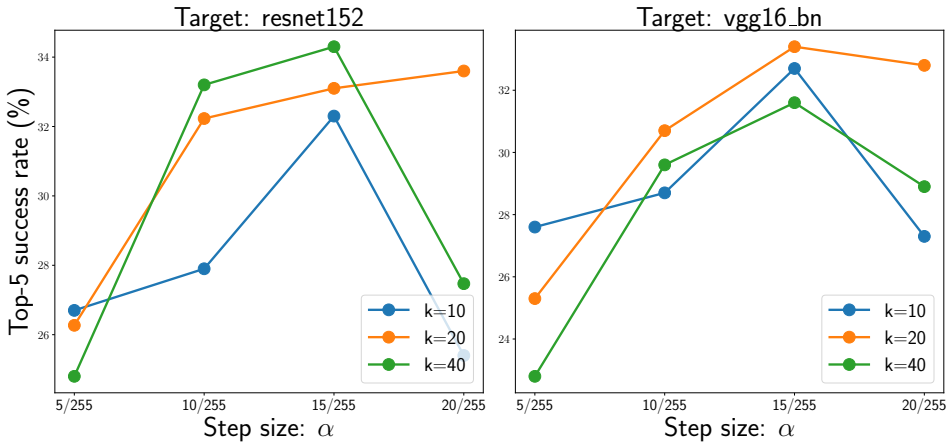


Figure 7: Average success rates over three ensembles for different step size α and number of iteration k. The three ensembles are the same with those in Table 3(b). Distortion $\varepsilon = 20$.

APPENDIX B: MNIST EXPERIMENTS

To further confirm the influence of model redundancy on the attack capability, we conduct an additional experiment on MNIST dataset. We use the fully networks of D layers with width of each layer being 500, e.g. the architecture of model with $D = 2$ is $784 - 500 - 500 - 10$. Models of depth $1, 3, 5, 9$ are considered in this experiment. The Top-1 success rates of cross-model attacks are reported in Table 4.

The results of Table 4 demonstrate the low-capacity model has much stronger attack capability than large-capacity. This is consistent with our observation in Section 6.4.

Table 4: Top-1 success rates of FGSM attacks with $\varepsilon = 40$. Each cell (S,T) indicates the percentage of adversarial examples generated for model S evaluated on model T.

	$D = 1$	$D = 3$	$D = 5$	$D = 9$
$D = 1$	89.37	62.94	62.90	64.42
$D = 3$	52.86	60.32	48.25	49.38
$D = 5$	47.30	43.05	55.84	44.83
$D = 9$	31.19	29.22	29.02	39.60

APPENDIX C: SOME ADDITIONAL EXPERIMENTAL RESULTS

Table 5: Top-1 Success rates of non-targeted FGM and vr-FGM ℓ_2-attacks between pairs of models. The cell (S,T) indicates the percentage of adversarial examples generated for model S(row) that successfully fool model T(column). For each cell: (1) The left is the success rate of FGM; (2) The right rate is the one of vr-FGM. In this experiment, distortion $\varepsilon = 15$.

	alexnet	densenet121	resnet152	resnet34	vgg13_bn	vgg19_bn
alexnet	98.0 / 97.0	27.2 / 30.5	19.9 / 24.6	38.1 / 41.0	47.2 / 46.2	34.5 / 39.0
densenet121	53.5 / 61.6	93.1 / 98.6	34.5 / 68.7	43.6 / 76.2	50.7 / 72.8	44.2 / 74.6
resnet152	50.2 / 54.1	37.9 / 67.9	82.8 / 95.7	45.5 / 72.5	40.6 / 63.4	35.7 / 64.7
resnet34	55.1 / 61.3	45.8 / 72.2	36.6 / 68.3	94.5 / 98.5	50.7 / 70.6	45.2 / 70.6
vgg13_bn	45.4 / 55.1	25.7 / 53.8	14.9 / 38.1	28.6 / 59.3	97.0 / 98.3	53.7 / 86.6
vgg19_bn	42.9 / 53.3	27.4 / 64.0	18.1 / 48.4	32.0 / 63.2	57.1 / 85.9	91.8 / 97.7

Table 6: Top-1 success rates of ensemble-based non-targeted FGSM and nr-FGSM attacks. Each cell (S,T) indicate the percentages of targeted adversarial examples are predicted as the target label by model (T). The left is the result of FGSM, while the right is the ones of nr-FGSM. Distortion $\varepsilon = 15$

Ensemble	densenet121	resnet152	resnet50	vgg13_bn
resnet18+resnet34+resnet101	62.0 / **86.0**	59.0 / **86.2**	67.3 / **89.6**	63.7 / 83.2
vgg11_bn+densenet161	52.3 / 81.9	35.8 / 70.5	48.0 / 78.8	69.3 / 89.6
resnet34+vgg16_bn+alexnet	50.8 / 80.2	38.3 / 73.3	52.8 / 82.3	74.8 / **92.3**

Table 7: Top-1 success rates of ensemble-based non-targeted IGSM and nr-IGSM attacks. Each cell (S,T) indicate the percentages of targeted adversarial examples are predicted as the target label by model (T). The left is the result of IGSM ($k = 100, \alpha = 3$), while the right is the ones of nr-IGSM ($k = 50, \alpha = 3, m = 20, \sigma = 15$). 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 8: Top-1 success rates of ensemble-based targeted IGSM and nr-IGSM attacks. The cell (S,T) indicates the percentages of targeted adversarial examples generated from model S(row) are predicted as the target label by model T(column). For each cell: The left is the results of IGSM ($k = 20, \alpha = 15$), while the right is the ones of nr-IGSM ($k = 20, \alpha = 15, m = 20, \sigma = 15$). 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**

Figure 8: The image used to perform decision boundary analysis. Its ID in ILSVRC2012 validation set is 26, with ground truth label being *table lamp*.

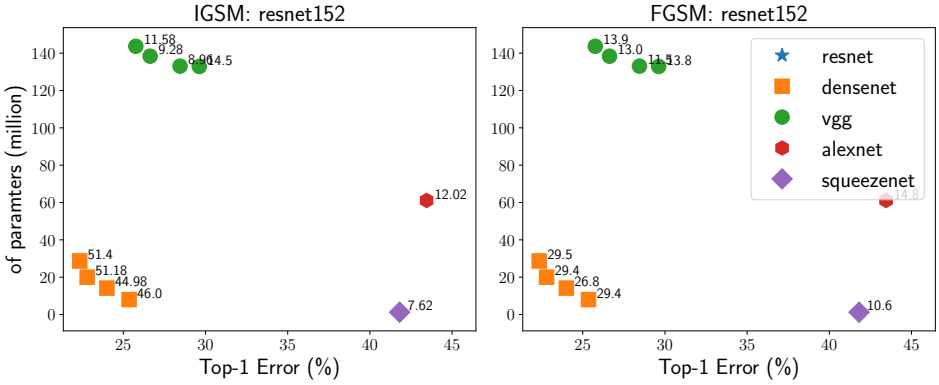


Figure 9: Top-1 success rates of FGSM and IGSM ($k = 20, \alpha = 5$) attacks against *resnet152* for various models. The annotated value is the percentage of adversarial examples that can transfer to the *resnet152*. Here, the models of resnet-style have been removed, since the contribution from architecture similarities is not in our consideration. The distortion is chosen as $\varepsilon = 15$.