Adversarial Eigen Attack on Black-Box Models

Linjun Zhou[1] Peng Cui[1] Xingxuan Zhang[1] Yinan Jiang[2] Shiqiang Yang[1]
[1]Tsinghua University [2]China Academy of Electronics and Information Technology
zhoulj16@mails.tsinghua.edu.cn, cuip@tsinghua.edu.cn
xingxuanzhang@hotmail.com, jiang_yinan@126.com, yangshq@mail.tsinghua.edu.cn

Abstract

Black-box adversarial attack has aroused much research attention for its difficulty on nearly no available information of the attacked model and the additional constraint on the query budget. A common way to improve attack efficiency is to transfer the gradient information of a white-box substitute model trained on an extra dataset. In this paper, we deal with a more practical setting where a pre-trained white-box model with network parameters is provided without extra training data. To solve the model mismatch problem between the white-box and black-box models, we propose a novel algorithm EigenBA by systematically integrating gradient-based white-box method and zeroth-order optimization in black-box methods. We theoretically show the optimal directions of perturbations for each step are closely related to the right singular vectors of the Jacobian matrix of the pretrained white-box model. Extensive experiments on ImageNet, CIFAR-10 and WebVision show that EigenBA can consistently and significantly outperform state-of-the-art baselines in terms of success rate and attack efficiency.

1. Introduction

Despite the fast development of deep learning, its security problem has aroused much attention. It has been demonstrated that a deep learning model can be successfully attacked at a small query cost without knowing the specific implementation of the model. Such techniques are called black-box attack [6, 11, 19], which is intensively studied in recent years with the aim of promoting the development of machine learning towards robustness.

In previous studies, there are two kinds of settings related to black-box attack. One is pure black-box attack, where nothing is available but the input and output of the black-box model. A common technique used in this setting is the zeroth-order optimization [11], where the gradient information is estimated by sampling different directions of perturbation and aggregating the relative changes of a certain loss function related to the output. The other setting is transfer-based attack [10], where a substitute white-box model is trained on an extra training dataset, and the gradient information of the white-box model is exploited to help improve the efficiency of attacking the black-box model. Usually, by leveraging extra information, transfer-based attack is more efficient and effective than pure black-box attack. But completely re-training a complex model is time-consuming and even infeasible if sufficient training data is unavailable.

In this paper, we aim for a new setting of transfer-based attack. Considering the easy availability of pre-trained models, we assume a pre-trained white-box model (*i.e.* its network structure and parameters) is given, but there is no additional training dataset available. In other words, the pre-trained model cannot be modified or fine-tuned before being used for black-box attack. Then in this setting, the critical challenge that we need to tackle is the model mismatch between the pre-trained white-box model and the black-box model, which is presented in two cases. One is that the conditional probability of $P(y|x)$ for two models is different. This will lead to disagreement on gradient direction of the two models. The other, a even more challenging case, is that the category label set is different in white-box and black-box models. In literature, the first case is partially tackled by [2, 7, 25]. However, they ask the label set of the two models to be the same and utilize the information of the output class probability given by the pre-trained model when attacking. This limits the practical use, as in real applications, cases of totally same label set for two models are rare and even in more extreme scenarios the pre-trained model is trained in an unsupervised manner [8], where no label information is available from the pre-trained model.

To solve this model mismatch problem in broader scenarios, we combine the ideas of white-box attack and black-box attack and utilize the representation layer of the pre-trained model. We regard the mapping function from the intermediate representation of the white-box model to the output of the black-box model as a black-box function, and exploit common practices of black-box attack on this black-box function. Meanwhile, the mapping from the original input to the intermediate representation layer is a part of the pre-trained model, which could be processed as in a white-box setting. It is noteworthy that the rationality of the idea depends on the generalization ability of the intermediate representation layer in the pre-trained white-box model. This can be underpinned by the findings in previous works that the lower layers of deep neural networks, *i.e.* the representation learning layers, are transferrable across different datasets or data distributions [26].

More specifically, we propose a novel Eigen Black-box Attack (EigenBA) method by systematically integrating the gradient-based white-box method and zeroth order optimization in black-box methods. We theoretically prove that the most efficient attack is to conduct singular value decomposition to the Jacobian matrix of the intermediate representation layer to the original inputs in the white-box model, and perturb the input sample with the right singular vectors corresponding to the k largest singular values iteratively.

We conduct extensive experiments to evaluate the effectiveness of EigenBA in multiple settings. The results demonstrate that EigenBA can consistently and significantly outperform state-of-the-art baselines in terms of success rate and attack efficiency. Also, the ablation studies show that EigenBA's advantage can be exerted as long as the representation layer of the white-box model has moderate generalization ability, implying its wide applicability in practice.

2. Related Works

White-Box Attack White-box attack requires knowing all the information of the attacked model. As the earliest research field among adversarial attacks, there has been a vast literature on the white-box attack, and we will only cover methods with first-order gradient attack in this part, which is closely related to our topic. The adversarial examples are first proposed by [23]. They found that adding some specific small perturbations to the original samples may lead to classification errors of the neural network and [4] further explains this phenomenon as the linear behavior in high-dimensional space of neural networks. Later on, several algorithms are proposed to find adversarial examples with a high success rate and efficiency. Classical first-order attack algorithms include FGSM [4], JSMA [21], C&W attacks [1], PGD [16]. The common principle for these methods is to iteratively utilize the first-order gradient information of a particular loss function with respect to the input of the neural networks. Specifically, the direction of the perturbation for each iteration is determined by a certain transformation of the gradient.

Black-Box Attack Black-box attack deals with the case when the attacked model is unknown, and the only way to obtain the information of the black-box model is to iteratively query the output of the model with an input. Hence, the efficiency evaluation of the black-box model includes three aspects: success rate, query numbers and the l_2 or l_∞ norm of the perturbation to original sample. Black-box attack could be divided to two categories: **black-box attacks with gradient estimation** and **black-box attacks with substitute networks** [19]. The former uses a technique called zeroth-order optimization. Typical work includes NES [11], Bandits-TD [12], LF-BA [5], SimBA [6]. The idea of these papers is to estimate gradient with sampling. More recently, some works view the problem as black-box optimization and propose several algorithms to find the optimal perturbation, for example, [19] uses a submodular optimization method, [22] uses a bayesian optimization method and [18] uses an evolutional algorithm. The latter utilizes a white-box substitute networks to help attack the black models. The substitute network either could be trained on additional samples or from a pre-trained model, the former includes Substitute Training [20], AutoZOOM [24], TREMBA [10], NAttack [15], and the latter includes P-RGF [2], Subspace Attack [7], TIMI [3] and LeBA [25]. The efficiency of these transfer-based methods is largely depended on the quality of the substitute networks. If the model mismatch is severe between two networks, the transfer-based method may underperform the methods with gradient estimation. Our work follows the latter setting, but with broader application scenarios, we could even deal with cases when only representation layer information is available from the white-box pre-trained model.

3. Models

3.1. Problem Formulation

Assume we have an input sample $x \in \mathbb{R}^n$ and a black model $F : \mathbb{R}^n \to [0,1]^{c_b}$, classifies c_b classes with output probability $p_F(y|x)$ with unknown parameters. The general goal for black-box attack is to find a small perturbation δ such that the prediction $\arg\max F(x+\delta) \neq y_{true}$, where y_{true} is the true label of corresponding x. A common practice for score-based black-box attack is to iteratively query the output probability vector given an input adding an evolutional perturbation. Three indicators are used to reflect the efficiency of the attack algorithm: the average query number for attacking one sample, the success rate and average l_2-norm or l_∞-norm of the perturbation (*i.e.* $||\delta||_2$ or $||\delta||_\infty$).

We propose a novel setting of transfer-based black-box attack. We further assume there is a white-box model $G(x) = g \circ h(x)$, where $h : \mathbb{R}^n \to \mathbb{R}^m$ maps the original input to a low-dimensional representation space, and $g : \mathbb{R}^m \to [0,1]^{c_w}$ maps the representation space to output classification probabilities, c_w is the number of classes with respect to G. The original classes for classifier F and G may or may not be the same. The parameters of g and h are known, but are not permitted to be further tuned by additional training samples. Our goal is to utilize G to enhance the efficiency of attacking the black-model F given an input x. *i.e.* to decrease the query number for black-box model under the same level of perturbation norm.

3.2. The EigenBA Algorithm

3.2.1 General Framework

One of the main challenges is that the white-box pre-trained model G may show model mismatch to the actual attacked model F. Even with the same output classes, the probability $p_G(y|x)$ may be different from $p_F(y|x)$. Hence, directly attacking $p_G(y|x)$ based on white-box methods may not work well on F, not to mention a different output classes case. However, benefited from the generalization ability of deep neural networks, if the classification tasks of the two models are related, the representation layer of G has a certain predictive power to the output classes of F. Formally, following notations $G = g \circ h$ in Section 3.1, the black-box model F could be approximated as $\tilde{g} \circ h$, where h is the encoder of the white-box model G, and $\tilde{g} : \mathbb{R}^m \rightarrow [0, 1]^{c_b}$ is a new mapping function from the representation space of G to the output of the attacked model F. As there exists an optimal $\tilde{g}$ but we do not know its realization, the function $\tilde{g}$ could be seen as a new black-box target. For convenience of expression, we keep $F = \tilde{g} \circ h$ as a hypothesis in the following analysis.

Hence, the black-box attack could be reformulated as:

$$\min_{\delta} p_F(y|x+\delta) \Rightarrow \min_{\delta} p_{\tilde{g}\circ h}(y|x+\delta) \quad s.t. \quad ||\delta||_2 \leq \rho \tag{1}$$

Here in this paper, we only consider the l_2-attack. Using a gradient-descent method to iteratively find an optimal perturbation is given by $x_{t+1} = x_t - \epsilon \cdot \nabla_x[F(x;\theta)_y]$. As $\nabla_x[F(x;\theta)_y]$ is unknown in black-box model, we need to estimate it by sampling some perturbations and aggregating the relative change of the output. Noticing that the query number is also important in black-box attack, we measure the attack efficiency as the number of samples used under the same $dp/||\delta||_2$ for each iteration, where $dp = |p_F(y|x+\delta) - p_F(y|x)|$.

Specifically, define $z = h(x)$, the gradient could be decomposed as:

$$\nabla_x[F(x;\theta)_y] = J_h(x)^T \nabla_z[\tilde{g}(z;\tilde{\theta})_y] \tag{2}$$

where $J_h(x)$ is the $m \times n$ Jacobian matrix $\frac{\partial(z_1,z_2,\cdots,z_m)}{\partial(x_1,x_2,\cdots,x_n)}$ with respect to h, and the subscript y represents the y-th component of the output of $\tilde{g}$. As h is a white-box function, we could obtain the exact value of $J_h(x)$. In contrast, $\tilde{g}$ is a black-box function, we need to estimate the gradient $\nabla_z[\tilde{g}(z;\tilde{\theta})]_y$ by sampling. As the equation below holds given by the definition of directional derivatives:

$$\nabla_z[\tilde{g}(z;\tilde{\theta})_y] = \sum_{i=1}^{m} \left(\left.\frac{\partial \tilde{g}(z;\tilde{\theta})_y}{\partial \vec{l_i}}\right|_z \cdot \vec{l_i} \right), \tag{3}$$
$$\vec{l_1}, \vec{l_2}, \cdots, \vec{l_m} \; are \; orthogonal.$$

To completely recover the gradient of $\tilde{g}$, we could iteratively set the direction of the perturbations of z to any group of orthogonal basis $\vec{l_1}, \vec{l_2}, \cdots, \vec{l_m}$, which totally uses m samples for each iteration. However, there is an optimal group of basis with respect to the black-box attack efficiency, which will be introduce in next section.

3.2.2 Globally Optimal Perturbation Basis for Transferred Black-box Attack

In this section we will introduce our EigenBA algorithm to maximize the attack efficiency. The core problem is to maximize change of the output probability dp under the same perturbation norm $||\delta||_2$ and decrease the query numbers per iteration.

We first consider finding the orthogonal basis on the representation space by greedily exploring directions of perturbation on the original input space to maximize relative change of representation. Specifically, considering the first-order approximation of the change in representation space given by:

$$\vec{l_i} = J_h(x)\delta_i \tag{4}$$

where δ_i is the perturbation on original input space resulting the change of the representation space to be $\vec{l_i}$, the optimal perturbation could be seen as solving the following iterative problem:

$$\begin{aligned} &(P1) \quad \max_{\delta_1} ||J\delta_1||_2 \quad s.t. \quad ||\delta_1||_2 \leq \epsilon \\ &(P2) \quad \max_{\delta_i} ||J\delta_i||_2 \quad s.t. \quad ||\delta_i||_2 \leq \epsilon, \\ &\qquad\qquad \delta_j^T J^T J \delta_i = 0 \; for \; all \; j < i, \; i > 1 \end{aligned} \tag{5}$$

where $J_h(x)$ is simplified as J. We iteratively solve $\delta_1, \delta_2, \cdots, \delta_m$ of problem given by 5. In this way, the first constraint assures that the relative l_2-norm change from the original space to the representation space, *i.e.* $||\vec{l_i}||_2/||\delta_i||_2$ reaches a maximum and the second constraint assures the changes on the representation space are orthogonal.

Theorem 1 *The optimal solutions for problem given by (P1) and (P2) are that $\delta_1, \delta_2, \cdots, \delta_m$ are just the eigenvectors corresponding to the top-m eigenvalues of $J^T J$.*

The proof is shown in Appendix 1.1. Hence, if we iteratively sample the perturbation to $\delta_1, \delta_2, \cdots, \delta_m$ in order, the one-step actual perturbation $\nabla_x[F(x;\theta)_y]$ could be approximated by Equation 2 and Equation 3.

As the tail part of the eigenvalues may be small, *i.e.* the norm of perturbation for representation space may not be sensitive to the perturbation on the original input space with the corresponding eigenvector direction. To decrease the query number without sacrificing much attack efficiency, we only keep the top-K perturbations for exploration, $\delta_1, \delta_2, \cdots, \delta_K$. The eigenvectors of $J^T J$ could be fast calculated by processing a truncated singular value decomposition (SVD) to Jacobian matrix J, only keeping top K components.

From the discussion above, we demonstrate that the group of basis we found maximizes the change on representation space under the same perturbation norm of input. Next, we generalize our conclusion to the change on output space. The following theorem guarantees that by greedily exploring the optimal perturbations given by (P1) and (P2), the attack efficiency defined in Section 3.1 will be globally optimal for any composition of K orthogonal perturbation vectors on representation space, which forms the foundation of our EigenBA algorithm. The proof is shown in Appendix 1.2.

Theorem 2 *(Property of Eigen Perturbations) Assume there is no prior information about the gradient of $\tilde{g}$ (the direction of the actual gradient is uniformly distributed on the surface of an m-dimensional ball with unit radius). Given a query budget K for each iteration, the perturbations $\vec{l_1}, \vec{l_2}, \cdots, \vec{l_K}$ on representation space and the corresponding perturbations $\delta_1, \delta_2, \cdots, \delta_K$ on input space solved by Problem (P1) and (P2) is most efficient among any choice of exploring K orthogonal perturbation vectors on the representation space. Specifically, the final one-step gradient for $\nabla_z[\tilde{g}(z;\tilde{\theta})_y]$ is estimated by:*

$$\nabla_z[\tilde{g}(z;\tilde{\theta})_y] \simeq \sum_{i=1}^{K} \left(\frac{\partial \tilde{g}(z;\tilde{\theta})_y}{\partial \vec{l_i}} \bigg|_z \cdot \vec{l_i} \right)$$

and the expected change of the output probability $dp_F(y|x)$ reaches the largest with the same l_2-norm of perturbation on input space for all cases.

3.2.3 Further Improvements on Query Numbers

Another important improvement is inspired by SimBA [6] (See Algorithm 2 in Appendix 5). Instead of estimating the gradient by exploring a series of directional derivatives before processing one-step gradient descent, SimBA iteratively updates the perturbation by picking random orthogonal directions and either adding or subtracting to the current perturbation, depending on which operation could decrease the output probability. The main difference is that, SimBA pursues fewer queries by using a relatively fuzzy gradient estimation. SimBA does not concern about the absolute value of the directional derivatives, but only positive or negative. In such a way, the perturbations of the orthogonal basis used to explore the real gradient could also contribute to the decrease of the output probability. Inspired by SimBA, we substitute their randomly picked basis or DCT basis to our orthogonal basis $\delta_1, \delta_2, \cdots, \delta_K$ given by solving Problem 5. The whole process for our EigenBA algorithm is shown in Algorithm 1. Considering time efficiency, for each loop, we calculate SVD once with respect to the initial state of input of this loop and process K steps directional derivatives exploration with the corresponding K eigenvectors as perturbations. The idea of SimBA significantly reduces the number of queries, as shown in [6].

Algorithm 1 The EigenBA Algorithm for untargeted attack

Input: Target black-box model F, the substitute model $G = g \circ h$, the input x and its label y, stepsize α, number of singular values K.
Output: Perturbation on the input δ.

```
Let $\delta = 0$, $\mathbf{p} = p_F(y_1, y_2, \cdots, y_{c_b}|x)$, $succ = 0$.
while $succ = 0$ do
    Calculate Jacobian matrix w.r.t. $h$: $J = J_h(x + \delta)$.
    Process truncated-SVD as trunc-svd($J$,$K$) = $U, \Sigma, V^T$.
    Normalize each column of $V$: $q_i$ = normalize($V[:, i]$).

    for $i = 1 \cdots K$ do
        $\mathbf{p}_{neg} = p_F(y_1, \cdots, y_{c_b}|clip(x + \delta - \alpha \cdot q_i))$ //$clip(\cdot)$ for validity of the input.
        if $\mathbf{p}_{neg,y} < \mathbf{p}_y$ then
            $\delta = clip(x + \delta - \alpha \cdot q_i) - x$
            $\mathbf{p} = \mathbf{p}_{neg}$
            //negative direction decreases the probability.
        else
            $\mathbf{p}_{pos} = p_F(y_1, \cdots, y_{c_b}|clip(x + \delta + \alpha \cdot q_i))$
            if $\mathbf{p}_{pos,y} < \mathbf{p}_y$ then
                $\delta = clip(x + \delta + \alpha \cdot q_i) - x$
                $\mathbf{p} = \mathbf{p}_{pos}$
                //positive direction decreases the probability.
            end if
        end if
        if $\mathbf{p}_y \neq max_{y'} \mathbf{p}_{y'}$ then
            $succ = 1$
            break;
        end if
    end for
end while
Return $\delta$
```

Moreover, for complexity analysis of our EigenBA algorithm and some tricks to decrease the time complexity, we refer the readers to Appendix 2.

4. Experiments

4.1. Setup

In practical scenarios of the transfer-based black-box attack, there are two main sources of model mismatches: the attacked model is different from the pre-trained model in the model architecture, or the output classes (or both). Hence, we will evaluate our EigenBA algorithm from two aspects in the experiment part.

For the first group of experiments, we use a ResNet-18 [9] trained on ImageNet as the fixed white-box pre-trained model, and the attacked model is a ResNet-50 or Inception-v3 trained on the same training dataset of ImageNet. The attacked images are randomly sampled from the ImageNet validation set that are initially classified correctly to avoid artificial inflation of the success rate. For all baselines, we use the same group of attacked images. For the second group of experiments we show two different cases. A rather simple case is to use a ResNet-18 trained on CIFAR-100 [13] as white-box model, and the attacked model is a ResNet-18 trained on CIFAR-10 [13]. The more complex one is to use a ResNet-18 trained on ImageNet to attack a ResNet-50 trained on WebVision2.0 [14]. WebVision2.0 contains 16 million training images from 5,000 different visual concepts. Among them 1,000 concepts are overlapped with ImageNet, but the images are selected from a different source from ImageNet, and the other 4,000 concepts are newly added. To show difference on output classes, we randomly pick a subset containing 1,000 classes from the non-overlapped 4,000 classes for simplicity. The attacked model is limited on classifying the picked 1,000 classes. The reason we choose to attack model trained on WebVision dataset is that the categories of the two datasets are sufficiently different to show the superiority of our algorithm and the attacked model is more like a real scene model. Similarly, the attacked images are also randomly sampled from the correctly classified images from the validation set of CIFAR-10 or WebVision2.0. We summarize the settings of all experiments in Table 1. The top two settings and the bottom two settings illustrate the two types of model mismatch described above separately, with a more detailed description of the differences on models.

We also process the untargeted attack case and the targeted attack case in some settings, same as the previous literature of black-box attack. The main difference is that the targeted attack requires the model misclassifies the adversarial sample to the assigned class, while the untargeted attack just makes the model misclassified. Compared with untargeted attack, the goal for targeted attack is to increase $p_F(c|x)$ instead of decreasing $p_F(y|x)$, where c is the assigned class. Hence, we only need to make a small change to Algorithm 1 by substituting $p_F(y|x)$ by $-p_F(c|x)$.

For all experiments, we limit the attack algorithm to 10,000 queries for ImageNet, 2,000 for CIFAR-10 and 5,000 for WebVision. Exceeding the query limit is considered as an unsuccessful attack. There are 1,000 images to be attacked for each setting. We evaluate our algorithm and all baselines from 4 indicators: The average query number for success samples only, the average query number for all attacked images, the success rate and the average l_2-norm of the perturbation for success samples.

We compare EigenBA to several baselines. Despite our l_2 attack setting, we also test some state-of-the-art baselines for l_∞ attack, as the l_2 norm $||\delta||_2$ is bounded by $\sqrt{dim(\delta)} \cdot ||\delta||_\infty$ and algorithms for l_∞ attack could also be adapted to l_2 attack. Baseline algorithms could be divided into two branches. One of the branches is the common black-box attack with no additional information, we compare several state-of-the-art algorithms including SimBA [6], SimBA-DCT [6] and Parsimonious Black-box Attack (ParsiBA) [19]. The main concern to be explained by comparing with these methods is to show exploring the representation space provided by a pre-trained model with a slight distribution shift is more efficient than the primitive input space or low-level image space (*e.g.* DCT space). The other branch is some extensible first-order white-box attack methods that could be adapted to this setting. We design two baselines: Trans-FGSM and Trans-FGM. The two baselines are based on the Fast Gradient Sign Method and the Fast Gradient Method [4]. While conducting them, we use the same pre-trained white-box model as our algorithm. The two baselines iteratively run SimBA algorithm by randomly selecting from the Cartesian basis on the representation space. And the updating rule for the perturbation on input space is given by:

$$\text{Trans-FGSM:} \quad \delta_{t+1} = \delta_t \pm \alpha \cdot sign(\nabla_x h(x_t; e_i))$$

$$\text{Trans-FGM:} \quad \delta_{t+1} = \delta_t \pm \alpha \cdot \frac{\nabla_x h(x_t; e_i)}{||\nabla_x h(x_t; e_i)||_2}$$

where e_i is the selected i_{th} basis and $\nabla_x J_h(x_t; e_i)$ is the gradient of the i_{th} output representation value z_i with respect to the input x_t. By comparing these two methods, we will show afterward that exploring the eigenvector orthogonal subspace on representation space is more efficient than other subspace, which is consistent with Theorem 2. It is noteworthy that ParsiBA and Trans-FGSM are originally for l_∞ attack. More details of the experimental setting is shown in Appendix 2.3.

Moreover, it is noteworthy that P-RGF [2], Subspace Attack [7] and LeBA [25] could also deal with the first setting, *i.e.* the change of the model architecture. However, they utilize more information from the output classification probability of the pre-trained model than ours, leading to more efficient attack but narrower usage scenarios. Their methods could not deal with pre-trained models without classification layer (*e.g.* training in an unsupervised manner) or different label set between pre-trained model and attacked model (*i.e.* the second setting in our experiment). Hence

Table 1. Summary of our experiments: the differences of the pre-trained model and the black-box model on 4 aspects. The check mark indicates the two models are different on the corresponding aspect. Content in brackets shows the training dataset of the model.

Pre-trained Model	Attacked Black-box Model	Model Variant	Model Type	Training Data	Labels
ResNet-18 (ImageNet)	ResNet-50 (ImageNet)	✓			
ResNet-18 (ImageNet)	Inception-v3 (ImageNet)	✓	✓		
ResNet-18 (CIFAR-100)	ResNet-18 (CIFAR-10)			✓	✓
ResNet-18 (ImageNet)	ResNet-50 (WebVision)	✓		✓	✓

Table 2. Results for untargeted and targeted attack on attacking ResNet-50 (trained on ImageNet). Max queries = 10000

Method	Transfer	Untargeted				Targeted			
		Avg. queries (success)	Avg. queries (all)	Success Rate	Avg. l_2	Avg. queries (success)	Avg. queries (all)	Success Rate	Avg. l_2
SimBA	No	1322	1417	0.989	3.989	5762	6719	0.774	8.424
SimBA-DCT	No	804	933	0.986	3.096	4387	5437	0.813	6.612
ParsiBA	No	997	1312	0.965	3.957	5075	6878	0.634	8.422
Trans-FGSM	Yes	510	614	0.989	4.634	3573	4807	0.808	9.484
Trans-FGM	Yes	675	843	0.982	3.650	3562	5867	0.642	8.200
EigenBA (Ours)	Yes	383	518	0.986	3.622	2730	4140	0.806	7.926

in this paper, we only adopt the baselines with the same applicability as our method for fair comparison.

4.2. Results on Change of Architectures

In this section we show the main results of attacking ImageNet in Table 2 and Table 3, *i.e.* the top two settings shown in Table 1. We adjust the hyper-parameter stepsize α for our method and all baselines to make sure the average l_2-norm of perturbation is close and compare average queries and success rate for easier comparison.

Table 2 shows the results of untargeted attack and targeted attack under the pre-trained model ResNet-18 and the attacked model ResNet-50. Comparing our EigenBA to those algorithms without transferred pre-trained model, our method uses at most 56% query numbers for untargeted attack and about 76% for targeted attack and reaches a comparable success rate, which demonstrates that utilizing the representation space of a smaller model could attack more efficiently than the original pixel space or manually designed low-level DCT space. Moreover, some state-of-the-art methods, *e.g.* SimBA-DCT, take advantage of the general properties of images and could not be generalized to other fields. In contrast, our method is applicable to any black-box attack scenario with a pre-trained model.

Comparing EigenBA to Trans-FGM, which is more suitable for l_2-attack than Trans-FGSM, our method use about 61% query numbers for untargeted attack and 71% for targeted attack. The results demonstrate that exploring the eigenvector subspace generated by solving problem given by 5 on the representation space is more efficient than the subspace generated by randomly chosen orthogonal basis, which is consistent to our theoretic reflection in Section 3. It is noteworthy that Trans-FGM performs similar or even worse to SimBA-DCT, which shows transfer-based method is not necessarily better than pure black-box attack methods, depending on whether the representation space provided by the transferred model is strong enough and the efficiency of the algorithm itself.

Table 3. Results for untargeted attack on attacking Inception-v3 (trained on Imagenet). Max queries = 10000

Methods	Avg. queries (success)	Avg. queries (all)	Success Rate	Avg. l_2
SimBA	2541	3533	0.867	5.906
SimBA-DCT	1625	2169	0.935	4.245
ParsiBA	1710	2829	0.865	6.916
Trans-FGSM	967	1482	0.943	5.571
Trans-FGM	955	1733	0.914	4.759
EigenBA (Ours)	968	1356	0.957	4.629

Figure 1 further shows the change of success rate with the change of query number limit for EigenBA, SimBA-DCT and Trans-FGM. We can conclude the distribution of the query number for 1000 attacked images for each attack method. Our EigenBA algorithm performs especially better when the limit of query number is relatively small, which

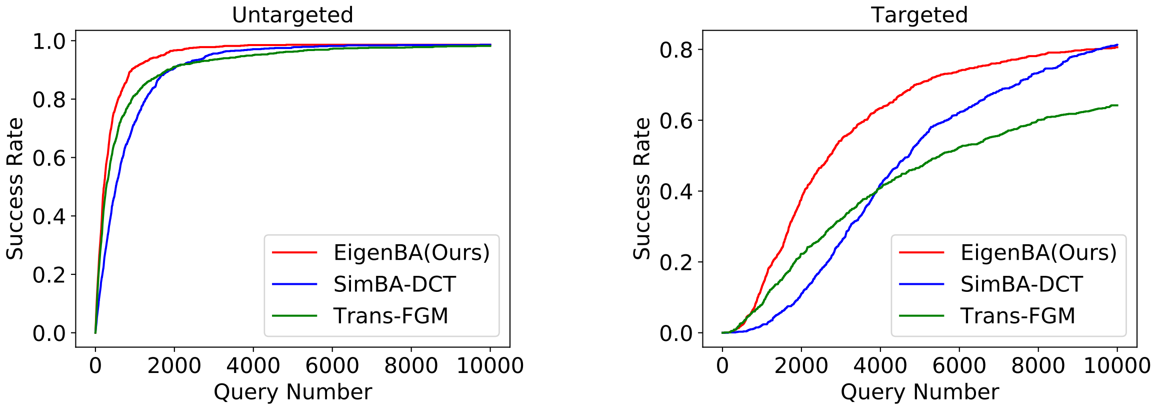

Figure 1. The change of success rate with fixed query limit on attacking ResNet-50 (trained on ImageNet).

will significantly reduce the query cost.

Table 3 shows the result of untargeted attack under the pre-trained model ResNet-18 and the attacked model Inception-v3. Our EigenBA algorithm still performs the best among all baselines, with the smallest average query number for all attacked images, highest success rate and nearly smallest perturbation, which shows that even the pre-trained model is totally different from the attacked model, our EigenBA algorithm still works well.

4.3. Results on Change of Output Classes

A more difficult setting is that the training dataset and the output classes of the attacked model are totally different from the pre-trained model, referring to the bottom two settings in Table 1. However, similar to the experiments on ImageNet, our EigenBA method still performs the best of all on attacking CIFAR-10, with a pretrained model trained on CIFAR-100, as shown in Table 4, and attacking WebVision with a pretrained model on ImageNet shown in Table 6. For attacking CIFAR-10, compared with SimBA-DCT, our algorithm uses 23% and 29% query numbers on untargeted attack and targeted attack while reaching a higher success rate. Compared with Trans-FGM, the proportion is 73% and 58%. Moreover, on the more difficult setting of WebVision dataset, even the training dataset, output classes and the model architecture are all changed, our EigenBA algorithm still saves about 19% query numbers compared with SimBA-DCT, while reaching a higher success rate. In contrastive, the other two transferred algorithms perform worse than pure black-box attack. It further shows that our algorithm can more effectively use the information of the pre-trained model. In conclusion, even the classes of the transferred model are different from the attacked model, depending on the strong generalization ability of neural network, the representation space of the transferred network can still improve the efficiency of black-box attack.

It is also noteworthy that the performance of our EigenBA algorithm highly depends on the generalization ability of the pre-trained model to the categories related to the attacked model, which is largely attributed to the similarity of the two training datasets for the pre-trained model and the attacked model. As CIFAR-100 and CIFAR-10 have a closer relationship than ImageNet and WebVision, our algorithm performs much better on attacking CIFAR-10. In next section, we will show more experimental evidences for the relationship between generalization ability and the efficiency of the attack.

4.4. Ablation Study: How the generalization ability affects the efficiency of attack?

From the results of Section 4.2 and 4.3, one interesting problem is how strong the generalization ability of the pre-trained white-box model can help improve the efficiency of black-box attack. In this section, we conduct an ablation study on this problem. In this experiment, we set the pre-trained model and the attacked model to be the same ResNet-18 trained on CIFAR-10, but randomly setting a certain proportion of parameters to be zero for the pre-trained model. If the reserve rate of parameters is 1.0, the pre-trained model will be totally the same with the attacked model, and with the decrease of the reserve rate, the generalization ability of the pre-trained model will become weaker. Setting a random part of parameters to zero could also be seen as a change to the structure of the pre-trained network. We test the attack efficiency of EigenBA under different reserve rate ratios and compare the result with the pure black-box method SimBA-DCT in Table 5. We also report the pre-trained model accuracy in different settings by fixing network parameters below the final representation layer and only re-training the top classifier with the training dataset of CIFAR-10, which reflects the generalization ability of the pre-trained

Table 4. Results for untargeted and targeted attack on attacking ResNet-18 (trained on CIFAR-10). Max queries = 2000

Methods	**Transfer**	**Untargeted**				**Targeted**			
		Avg. queries (success)	**Avg. queries (all)**	**Success Rate**	**Avg. l_2**	**Avg. queries (success)**	**Avg. queries (all)**	**Success Rate**	**Avg. l_2**
SimBA	No	460	467	0.995	0.574	817	883	0.944	0.782
SimBA-DCT	No	426	436	0.994	0.573	772	830	0.953	0.777
Trans-FGSM	Yes	111	115	0.998	0.638	305	310	0.997	0.918
Trans-FGM	Yes	129	135	0.997	0.524	369	419	0.969	0.747
EigenBA (Ours)	Yes	95	99	0.998	0.472	241	244	0.998	0.692

Table 5. Set a certain proportion of the parameters of the pre-trained model in EigenBA to zero, for attack on CIFAR-10.

Methods	**Parameters Reserved Rate**	**Avg. queries (all)**	**Success Rate**	**Avg. l_2**	**Pre-trained Model Accuracy**
EigenBA	1.0	88	1.000	0.453	89.19%
	0.9	85	1.000	0.446	86.17%
	0.8	130	0.997	0.459	77.78%
	0.7	195	0.999	0.560	69.36%
	0.6	382	0.991	0.760	35.36%
	0.5	700	0.921	0.951	27.57%
SimBA-DCT	-	440	0.998	0.575	-

Table 6. Results for untargeted attack on attacking ResNet-50 (trained on WebVision). Max queries = 5000

Methods	**Avg. queries (success)**	**Avg. queries (all)**	**Success Rate**	**Avg. l_2**
SimBA	1429	1672	0.932	4.306
SimBA-DCT	891	1068	0.957	4.354
Trans-FGSM	973	1713	0.816	5.125
Trans-FGM	853	1375	0.874	4.402
EigenBA (Ours)	679	861	0.958	4.406

model.

The results show that when the reserve rate is larger than 0.7, the pre-trained model is helpful to the efficiency of the black-box attack (both query number and average l_2 are lower.). And when the reserve rate is smaller than 0.5, the model will degrade the attack efficiency. The breakeven point may appear around 0.6. It shows that even the pre-trained model cannot achieve the classification accuracy of the attacked model, it can still improve the efficiency of the black-box attack, *e.g.* in this experiment, a pre-trained model with reserve rate of 0.7 just reaches 69.36% of classification on CIFAR-10, roughly equivalent to a shallow convolutional network [17], which is largely below the attacked model with 89.19%. Hence, as the representation layer of the modern neural networks generally has a strong transferability [26], our EigenBA algorithm has strong applicability in practice.

5. Conclusions

In this paper, we dealt with a novel setting for transfer-based black-box attack. Attackers may take advantage of a fixed white-box pre-trained model without additional training data, to improve the efficiency of the black-box attack. To solve this problem, we proposed EigenBA, which iteratively adds or subtracts perturbation to the input sample such that the expected change on the representation space of the transferred model to be the direction of right singular vectors corresponding to the first K singular values of the Jacobian matrix of the pre-trained model. Our experiments showed that EigenBA is more query efficient in both untargeted and targeted attack compared with state-of-the-art transfer-based and gradient estimation-based attack methods.We believe that the applicability in the real world of our algorithm will promote more research on robust deep learning and the generalization ability between deep learning models.

6. Acknowledgement

This work was supported in part by National Key R&D Program of China (No. 2018AAA0102004), National Natural Science Foundation of China (No. U1936219, 62141607), and Beijing Academy of Artificial Intelligence (BAAI).

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 2

[2] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. In *Advances in Neural Information Processing Systems*, pages 10932–10942, 2019. 1, 2, 5

[3] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 2

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2, 5

[5] Chuan Guo, Jared S Frank, and Kilian Q Weinberger. Low frequency adversarial perturbation. *arXiv preprint arXiv:1809.08758*, 2018. 2

[6] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pages 2484–2493, 2019. 1, 2, 4, 5

[7] Yiwen Guo, Ziang Yan, and Changshui Zhang. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. In *Advances in Neural Information Processing Systems*, pages 3825–3834, 2019. 1, 2, 5

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[10] Zhichao Huang and Tong Zhang. Black-box adversarial attack with transferable model-based embedding. *arXiv preprint arXiv:1911.07140*, 2019. 1, 2

[11] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146, 2018. 1, 2

[12] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018. 2

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[14] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017. 5

[15] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *International Conference on Machine Learning*, pages 3866–3876, 2019. 2

[16] 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. 2

[17] Mark D McDonnell and Tony Vladusich. Enhanced image classification with a fast-learning shallow convolutional neural network. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2015. 8

[18] Laurent Meunier, Jamal Atif, and Olivier Teytaud. Yet another but more efficient black-box adversarial attack: tiling and evolution strategies. *arXiv preprint arXiv:1910.02244*, 2019. 2

[19] Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *International Conference on Machine Learning*, pages 4636–4645, 2019. 1, 2, 5

[20] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017. 2

[21] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016. 2

[22] Binxin Ru, Adam Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. In *International Conference on Learning Representations*, 2020. 2

[23] 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. 2

[24] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019. 2

[25] Jiancheng Yang, Yangzhou Jiang, Xiaoyang Huang, Bingbing Ni, and Chenglong Zhao. Learning black-box attackers with transferable priors and query feedback. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 5

[26] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 2, 8