Dual-Path Distillation: A Unified Framework to Improve Black-Box Attacks

Yonggang Zhang [1] **Ya Li** [2] **Tongliang Liu** [3] **Xinmei Tian** [1]

Abstract

We study the problem of constructing black-box adversarial attacks, where no model information is revealed except for the feedback knowledge of the given inputs. To obtain sufficient knowledge for crafting adversarial examples, previous methods query the target model with inputs that are perturbed with different searching directions. However, these methods suffer from poor query efficiency since the employed searching directions are sampled randomly. To mitigate this issue, we formulate the goal of mounting efficient attacks as an optimization problem in which the adversary tries to fool the target model with a limited number of queries. Under such settings, the adversary has to select appropriate searching directions to reduce the number of model queries. By solving the efficient-attack problem, we find that we need to distill the knowledge in both the path of the adversarial examples and the path of the searching directions. Therefore, we propose a novel framework, dual-path distillation, that utilizes the feedback knowledge not only to craft adversarial examples but also to alter the searching directions to achieve efficient attacks. Experimental results suggest that our framework can significantly increase the query efficiency.

1. Introduction

Recent studies have shown that neural networks exhibit vulnerability to adversarial examples (Szegedy et al., 2014), which are constructed to fool models by adding imperceptible perturbations to normal examples. A host of methods have been developed to craft adversarial examples, and they can be utilized to analyze the weakness of machines (Athalye et al., 2018), evaluate model robustness (Carlini & Wagner, 2017; Moosavi-Dezfooli et al., 2016) and devise robust neural networks (Goodfellow et al., 2015; Madry et al., 2018). However, most of them are white-box attacks where the target models are transparent to the adversary. Black-box attacks can be more practical and valuable in real-world applications as long as the intrinsic details of the black box can be omitted.

From the view of an attack, accessing all the information of the target model may not be realistic in many real-world situations (Ilyas et al., 2018). In restrictive black-box settings, some works (Moosavi-Dezfooli et al., 2016; Dong et al., 2018; Poursaeed et al., 2018; Papernot et al., 2017; Zhang et al., 2020) build a transparent surrogate model to generate adversarial examples to attack the target, which are termed transfer-based attacks. Recent works (Bhagoji et al., 2018; Chen et al., 2017; Tu et al., 2019; Ilyas et al., 2018; 2019; Cheng et al., 2019) propose accessing the target model with inputs perturbed by different searching directions and utilizing the feedback knowledge to craft adversarial examples, which are termed query-based attacks.

However, current black-box attack methods suffer from either a low attack success rate or a high query complexity. It is shown that transfer-based attacks suffer from low attack success rates (Chen et al., 2017), because they construct adversarial examples either for a local surrogate model (Moosavi-Dezfooli et al., 2016; Dong et al., 2018; Poursaeed et al., 2018; Papernot et al., 2017) or in an unsupervised approach (Zhang et al., 2020) without sufficient knowledge of the target models. In query-based attacks, the adversary requires extensive model queries to search the data space. The demand for extensive queries stems from the fact that searching directions employed by existing methods lack the ability to adjust the target model. Hence, the problem of improving query efficiency still falls short of full mitigation.

Existing works improve the query efficiency mainly from two directions, either finding a more efficient gradient estimation algorithm (Chen et al., 2017; Bhagoji et al., 2018; Tu et al., 2019; Ilyas et al., 2018) or utilizing more prior information (Ilyas et al., 2019; Cheng et al., 2019). All these methods estimate the gradient with respect to the input images by randomly sampling the searching directions. Although these randomly selected searching directions can

[1]Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China [2]iFLYTEK, Hefei, China [3]UBTECH Sydney AI Centre, University of Sydney, New South Wales, Australia. Correspondence to: Xinmei Tian <xinmei@ustc.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

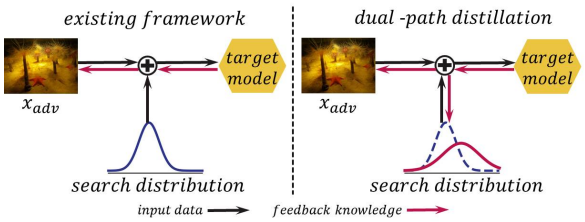

Figure 1. Difference between dual-path distillation and the existing black-box attack framework.

approximate the gradients well, a better searching directions selection method is required for efficiency considerations. In this paper, we formulate the problem of efficient attacks as an optimization problem with respect to searching directions. The search distribution is optimized to adapt to the target model for generating appropriate searching directions. The feedback knowledge is distilled through not only the path of adversarial examples but also the path of search distributions. The intuition behind dual-path distillation is to guarantee the better use of feedback knowledge from the target model. The difference between the derived dual-path framework and the existing framework is depicted in Fig. 1.

Our contributions are summarized as follows:

- We are the first to formulate the problem of efficient black-box attacks with respect to the searching directions. This challenging optimization problem is reduced to a dual-path distillation method that can be applied to existing attacks effortlessly.
- We propose a residual search distribution that can further improve the query efficiency.
- Extensive experiments demonstrate that our method can considerably improve the efficiency of various black-box attack methods and achieve state-of-the-art performance.

2. Related Work

In this section, we describe the generic formulation for constructing adversarial examples and review the most related works on the existing black-box attack framework.

2.1. Problem Setting

Given an input-label pair $(\boldsymbol{x}, y)$, an adversarial example $\boldsymbol{x}_{adv}$ tries to fool the target model C with imperceptible perturbations. For targeted attacks, the adversary tries to mislead the classifier to a specific target label y_{tar}, while in untargeted attacks, y_{tar} can be any incorrect label. The adversarial attack task can be generically formulated as

$$C\left(\boldsymbol{x}_{adv}\right) = y_{tar}, \text{ s.t. } \left\|\boldsymbol{x}_{adv} - \boldsymbol{x}\right\|_p \leq \epsilon, \tag{1}$$

where ϵ bounds the distance between $\boldsymbol{x}$ and $\boldsymbol{x}_{adv}$ measured by the ℓ_p norm.

We can thus generate adversarial examples by solving a constrained optimization problem

$$\boldsymbol{x}_{adv} = \underset{\left\|\boldsymbol{x}'-\boldsymbol{x}\right\|_p \leq \epsilon}{\arg\min}\ f\left(\boldsymbol{x}'\right). \tag{2}$$

Here, $f(x)$ is the classification loss of input x with respect to the target label y_{tar}, e.g., the cross-entropy loss. According to Eq. (2), we can use the gradient information of the inputs to construct adversarial examples. Some works that use white-box attacks (Goodfellow et al., 2015) assume that the adversary has full knowledge of the target model. It is shown that, in white-box attacks, the first-order approximation method can efficiently generate adversarial examples (Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017; Madry et al., 2018; Dong et al., 2018). Projected gradient decent (Madry et al., 2018) is a state-of-the-art method, which utilizes the iterative method (Kurakin et al., 2017) to craft adversarial examples

$$\boldsymbol{x}_{adv}^{t+1} = \textstyle\prod_{\mathcal{B}_p(\boldsymbol{x},\epsilon)}\left(\boldsymbol{x}_{adv}^{t} - \eta \boldsymbol{g}_{\boldsymbol{x}}^{t}\right), \tag{3}$$

where $\prod$ is the projection operator, $\mathcal{B}_p\left(\boldsymbol{x}, \epsilon\right)$ stands for the ℓ_p ball centered at $\boldsymbol{x}$ with a radius ϵ, η is the step size and $\boldsymbol{g}_{\boldsymbol{x}}^{t}$ denotes the gradient of the loss w.r.t. $\boldsymbol{x}^t$, and t is the step number of the iteration.

2.2. Black-Box Attacks

The assumption of black-box attacks is restrictive. In black-box attacks, the parameters of the target model are agnostic. Thus, the focus of black-box attacks is to acquire sufficient information to approximate the gradients[1].

Some methods utilize the transferability (Szegedy et al., 2014) of adversarial examples for gradient approximation, which are termed transfer-based attacks (Dong et al., 2018; Papernot et al., 2017; Poursaeed et al., 2018). These attacks train a local surrogate model to calculate gradients and suppose that the constructed adversarial examples can fool the target model because of transferability. Obviously, the difference between the surrogate model and the target model will lead to a bad approximation of the gradients. Besides the supervised approach, a recent work shows that the adversarial examples can be generated without requiring surrogate models (Zhang et al., 2020). Specifically, the unsupervised black-box attack constructs the adversarial examples through modeling the data manifold.

Different from transfer-based attacks, query-based attacks approximate the gradients by searching the data space. Specifically, they first perturb the inputs with different searching directions, then feed them into the target model,

[1]Combinatorial optimization is an alternative without gradient estimation (Moon et al., 2019), but it is applicable only to the ℓ_∞-norm.

and finally combine the inputs and outputs to approximate the gradients. According to the obtained outputs, query-based attacks can be roughly divided into two categories: score-based and decision-based attacks.

In score-based attacks (Chen et al., 2017; Tu et al., 2019; Ilyas et al., 2018), the probabilities (or logits) are available for the adversary, thus, zeroth-order optimization can be employed to estimate the gradients. The symmetric difference quotient (Lax & Terrell, 2014) is employed to estimate gradients pixel by pixel (Chen et al., 2017). The random gradient-free method (Nesterov & Spokoiny, 2017) and natural evolution strategy (Wierstra et al., 2011) are then utilized to improve the query efficiency (Tu et al., 2019; Ilyas et al., 2018). These methods explore the data space based on a search distribution $\mathcal{U}$. At each step t, the gradient $\boldsymbol{g}_{\boldsymbol{x}}^t$ of the classification loss w.r.t. the input $\boldsymbol{x}_{adv}^t$ is estimated over q searching directions. The estimated gradient $\widehat{\boldsymbol{g}}_{\boldsymbol{x}}$ takes the generic form of

$$\widehat{\boldsymbol{g}}_{\boldsymbol{x}}^t = \frac{1}{q}\sum_{i=1}^{q}\frac{f\left(\boldsymbol{x}_{adv}^t + \alpha\boldsymbol{u}_i\right) - f\left(\boldsymbol{x}_{adv}^t\right)}{\alpha}\boldsymbol{u}_i, \tag{4}$$

where $\boldsymbol{u}_i$ is a searching direction randomly sampled from $\mathcal{U}$ and α governs the estimation quality. When the gradient is obtained, it can be used to update the adversarial example.

A recent work (Ilyas et al., 2019) shows that incorporating time- and data-dependent priors can increase query efficiency. It points a new direction to improve black-box attacks. Then the transfer-based prior shows superiority in decreasing the query complexity (Cheng et al., 2019). In fact, these methods assume that real gradients have certain properties that can be used as priors before attacks. If the employed priors are consistent with the properties of real gradients, then model queries for this consistent knowledge can be saved. Because the estimated gradients are constructed by searching directions, these methods sample directions that have the same properties as real gradients. Nevertheless, these methods crucially depend on the utilized priors and extensive experiments to verify the effectiveness of potential priors. Different from these methods, our method can adapt the searching directions to the properties of the target model without any prior information.

In decision-based attacks, only the final decision is accessible. Random walk can work well for decision-based attacks (Brendel et al., 2018). This method starts from an adversarial example with large distortions and performs rejection sampling based on a distribution to progressively decrease the distortion (Brendel et al., 2018). A recent work shows that incorporating low-frequency priors can increase the query efficiency (Guo et al., 2019). However, this method suffers from the same shortcomings as methods based on priors.

3. Method

In this section, we first reformulate the problem of efficient black-box attacks with respect to the searching directions. Then, we propose a new method, dual-path distillation, to improve black-box attacks by better utilization of feedback knowledge from the target model. An efficient residual search distribution is proposed to make full use of the feedback knowledge. Finally, we show that dual-path distillation can be effortlessly applied to existing black-box attack methods.

3.1. The Problem of Efficient Black-Box Attacks

In black-box attacks, the adversary has a limited budget (e.g., the maximum number of queries) to access the target model. Previous methods randomly draw q searching directions and perturb the original input with these directions to access the target model. To alleviate the negative impact of this random selection process, the number of random sample searching directions must be large enough to approximate the gradients well, which causes an increase in the number of queries. However, a large sample size will definitely reduce the attack efficiency. On the other hand, if the searching directions are carefully chosen to significantly reduce the classification loss in Eq. (2), the convergence rate of the gradient approximation will be improved compared to randomly selected searching directions. Thus, we propose to endow the adversary with the ability to identify the searching directions that can significantly reduce the classification loss. That is, we prefer the estimated gradient, which can significantly reduce the classification loss in Eq. (2). To this end, we define the efficient attack loss as

$$\min_{\widehat{\boldsymbol{g}}_{\boldsymbol{x}}} J \coloneqq f\left(\boldsymbol{x}' - \eta\widehat{\boldsymbol{g}}_{\boldsymbol{x}}\right) - f\left(\boldsymbol{x}'\right). \tag{5}$$

Here, we denote $\boldsymbol{x}_{adv}^t$ by $\boldsymbol{x}'$ for simplicity and use it in the rest of the paper. The problem of efficient black-box attacks is difficult to solve due to its nonconvex nature with respect to $\widehat{\boldsymbol{g}}_{\boldsymbol{x}}$. In this paper, we propose to solve the problem using approximation techniques. To this end, we approximate the loss function by its first-order Taylor expansion at point $\boldsymbol{x}'$. The efficient attack loss then becomes:

$$\min_{\widehat{\boldsymbol{g}}_{\boldsymbol{x}}} J \approx f\left(\boldsymbol{x}'\right) - \eta\boldsymbol{g}_{\boldsymbol{x}}^\top\widehat{\boldsymbol{g}}_{\boldsymbol{x}} - f\left(\boldsymbol{x}'\right) = -\eta\boldsymbol{g}_{\boldsymbol{x}}^\top\widehat{\boldsymbol{g}}_{\boldsymbol{x}}. \tag{6}$$

Suppose the adversary samples q searching directions $\{\boldsymbol{u}_i\}_{i=1}^q$ and uses the random gradient-free method (Nesterov & Spokoiny, 2017) to estimate the gradients. Therefore, we can substitute (4) into (6):

$$\min_{\{\boldsymbol{u}_i\}_{i=1}^q} J \approx -\frac{\eta}{q}\sum_{i=1}^{q} h\left(\boldsymbol{x}', \boldsymbol{u}_i, \alpha\right)\boldsymbol{g}_{\boldsymbol{x}}^\top\boldsymbol{u}_i, \tag{7}$$

where $h\left(\boldsymbol{x}', \boldsymbol{u}_i, \alpha\right) = \frac{f\left(\boldsymbol{x}'+\alpha\boldsymbol{u}_i\right) - f\left(\boldsymbol{x}'\right)}{\alpha}$. Eq. (7) provides a

reformulation of the black-box optimization problem with respect to the searching directions.

Unfortunately, problem (7) does not have a closed-form solution because of the nonconvex nature of $h(\boldsymbol{x}', \boldsymbol{u}_i, \alpha)$. To solve the problem of efficient attacks, we need to calculate the gradient $\boldsymbol{g}_{\boldsymbol{u}_i}$ of the loss J w.r.t. $\boldsymbol{u}_i$. However, the parameters of $h(\boldsymbol{x}', \boldsymbol{u}_i, \alpha)$ are unavailable since h is a function of f and the parameters of f are hidden. This major obstacle prevents us from calculating $\boldsymbol{g}_{\boldsymbol{u}_i}$.

To address this problem, we propose to employ zeroth-order optimization as a black-box gradient estimation technique to approximate $\boldsymbol{g}_{\boldsymbol{u}_i}$. This yields (see Supplementary for details):

$$\begin{aligned}\widehat{\boldsymbol{g}}_{\boldsymbol{u}_i} &= -\frac{\eta}{qq_v}\sum_{j=1}^{q_v}\frac{\phi(\boldsymbol{u}_i+\beta\boldsymbol{v}_{ij})-\phi(\boldsymbol{u}_i)}{\beta}\boldsymbol{v}_{ij}\\ &\approx -\frac{\eta}{qq_v}\sum_{j=1}^{q_v}h(\boldsymbol{x}', \boldsymbol{u}_i, \alpha)\, h(\boldsymbol{x}'+\alpha\boldsymbol{u}_i, \boldsymbol{v}_{ij}, \alpha\beta)\,\boldsymbol{v}_{ij}\end{aligned} \tag{8}$$

where $\phi(\boldsymbol{u}_i) = h(\boldsymbol{x}', \boldsymbol{u}_i, \alpha)\,\boldsymbol{g}_{\boldsymbol{x}}^{\top}\boldsymbol{u}_i$, $\boldsymbol{v}_{ij}$ is a searching direction randomly sampled from a search distribution and β controls the sampling variance. Because of the limited query budget, we propose using a rough gradient estimate to alter the sampled direction. A recent work (Tu et al., 2019) shows that a rough gradient estimate can be obtained via one model query, so we set $q_v = 1$ in this paper and denote $\boldsymbol{v}_{ij}$ by $\boldsymbol{v}_i$ in the rest of the paper.

The derived formulation (8) shows that the adversary can learn to select the searching directions instead of randomly selecting them by accessing the target model. However, searching directions used in step t will be dropped and novel directions will be sampled for gradient estimation in step $t+1$. Thus, the ability to alter the searching directions utilized in step t has no benefit in estimating the gradient at step $t+1$. To solve this problem, we propose dual-path distillation (DPD).

3.2. Dual-Path Distillation

The proposed dual-path distillation method aims to learn the ability of selecting searching directions through an additional function. Without loss of generality, we can employ a function φ to transform a sampled direction into a desired direction. Thus, we can endow the adversary with the ability to identify appropriate directions by solving the following problem:

$$\min_{\boldsymbol{\theta}} J = -\frac{\eta}{q}\sum_{i=1}^{q}h(\boldsymbol{x}', \varphi(\boldsymbol{u}_i;\boldsymbol{\theta}), \alpha)\,\boldsymbol{g}_{\boldsymbol{x}}^{\top}\varphi(\boldsymbol{u}_i;\boldsymbol{\theta}), \tag{9}$$

where $\boldsymbol{\theta}$ denotes the parameters of φ. In light of (8), we can first utilize the feedback knowledge to calculate the gradients of the generated directions. According to the chain rule, we can then calculate the gradients $\boldsymbol{g}_{\boldsymbol{\theta}}$ of the loss J w.r.t. $\boldsymbol{\theta}$:

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{q}\frac{\partial J}{\partial \varphi(\boldsymbol{u}_i;\boldsymbol{\theta})}\frac{\partial \varphi(\boldsymbol{u}_i;\boldsymbol{\theta})^{\top}}{\partial \boldsymbol{\theta}} \approx \sum_{i=1}^{q}\widehat{\boldsymbol{g}}_{\varphi(\boldsymbol{u}_i;\boldsymbol{\theta})}\frac{\partial \varphi(\boldsymbol{u}_i;\boldsymbol{\theta})^{\top}}{\partial \boldsymbol{\theta}}. \tag{10}$$

With the gradient $\boldsymbol{g}_{\boldsymbol{\theta}}$, the parameters of function φ can be updated when the feedback information of the target model is provided.

The gradient $\boldsymbol{g}_{\boldsymbol{x}}$ for altering adversarial examples can be estimated as:

$$\widehat{\boldsymbol{g}}_{\boldsymbol{x}} = \frac{1}{q}\sum_{i=1}^{q}h(\boldsymbol{x}', \varphi(\boldsymbol{u}_i;\boldsymbol{\theta}), \alpha)\,\varphi(\boldsymbol{u}_i;\boldsymbol{\theta}). \tag{11}$$

If φ is a linear function and the search distribution is a normal distribution, we then obtain:

$$\min_{A} J = -\frac{\eta}{q}\sum_{i=1}^{q}h(\boldsymbol{x}', A\boldsymbol{u}_i, \alpha)\,\boldsymbol{g}_{\boldsymbol{x}}^{\top}A\boldsymbol{u}_i. \tag{12}$$

The gradients to update adversarial examples can be estimated as:

$$\widehat{\boldsymbol{g}}_{\boldsymbol{x}} = \frac{1}{q}\sum_{i=1}^{q}h(\boldsymbol{x}', A\boldsymbol{u}_i, \alpha)\,A\boldsymbol{u}_i. \tag{13}$$

We can see that the Hessian-aware method (Ye et al., 2018) is a special case of DPD under these assumptions.

We name the derived general framework as dual-path distillation since it distills knowledge not only in the adversarial examples path but also in the search distribution path. The basic idea behind the dual-path distillation method is to make full use of the feedback knowledge, which is consistent with distillation in (Hinton et al., 2015; Kim et al., 2018). In detail, recent works (Kim et al., 2018; Heo et al., 2019) show that mitigating information leakage is necessary for knowledge distillation. Thus, DPD provides a reasonable direction to improve the efficiency of black-box attacks.

However, Eq. (8) implies that DPD requires extra model queries to calculate the gradients for updating the function φ. This property seems to be harmful to the efficiency of the proposed DPD method. Therefore, DPD demands further exploitation to make full use of the feedback knowledge.

According to the definition of h, three feedback values $f(\boldsymbol{x}')$, $f(\boldsymbol{x}'+\alpha\varphi(\boldsymbol{u}_i;\boldsymbol{\theta}))$ and $f(\boldsymbol{x}'+\alpha\varphi(\boldsymbol{u}_i;\boldsymbol{\theta})+\alpha\beta\boldsymbol{v}_i)$ are required to calculate $\widehat{\boldsymbol{g}}_{\varphi(\boldsymbol{u}_i;\boldsymbol{\theta})}$, while only the first two of them are used in Eq. (11) for calculating the gradients $\widehat{\boldsymbol{g}}_{\boldsymbol{x}}$. In effect, we can also use $f(\boldsymbol{x}'+\alpha\varphi(\boldsymbol{u}_i;\boldsymbol{\theta})+\alpha\beta\boldsymbol{v}_i)$ and $f(\boldsymbol{x}')$ to estimate the

gradient at point $\boldsymbol{x}'$ as

$$\begin{aligned}\widehat{\boldsymbol{g}}_{\boldsymbol{x}} &= \frac{1}{q}\sum_{i=1}^{q}\frac{f\left(\boldsymbol{x}'+\alpha\boldsymbol{z}_i\right)-f\left(\boldsymbol{x}'\right)}{\alpha}\boldsymbol{z}_i \\ &= \frac{1}{q}\sum_{i=1}^{q}h\left(\boldsymbol{x}',\boldsymbol{z}_i,\alpha\right)\boldsymbol{z}_i,\end{aligned} \tag{14}$$

where $\boldsymbol{z}_i = \varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)+\beta\boldsymbol{v}_i$.

Combining Eq. (11) and Eq. (14), we can finally approximate the gradient with respect to $\boldsymbol{x}'$ as follows:

$$\begin{aligned}\widehat{\boldsymbol{g}}_{\boldsymbol{x}} = &\frac{1}{q}\sum_{i=1}^{q}h\left(\boldsymbol{x}',\varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right),\alpha\right)\varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right) \\ &+\frac{1}{q}\sum_{i=1}^{q}h\left(\boldsymbol{x}',\varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)+\beta\boldsymbol{v}_i,\alpha\right)\left(\varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)+\beta\boldsymbol{v}_i\right),\end{aligned} \tag{15}$$

Therefore, all query feedback can be utilized to update the adversarial examples and optimize the search distributions. This means that DPD reuses all query feedbacks for altering the searching directions and requires no extra model queries. Nevertheless, we find that the second term in Eq. (15) contains a randomly sampled direction $\boldsymbol{v}_i$. We introduce a similar way to transform the sampled direction $\boldsymbol{v}_i$ as that of updating $\boldsymbol{u}_i$. Another function $\psi\left(\boldsymbol{v_i};\boldsymbol{w}\right)$ is employed to transform the searching direction $\boldsymbol{v_i}$ used in Eq. (15). By introducing ψ, there are two transform functions ($\varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)$ and $\psi\left(\boldsymbol{v}_i;\boldsymbol{w}\right)$) for calculating the gradient $\widehat{\boldsymbol{g}}_{\boldsymbol{x}}$ in Eq. (15) and we can estimate the gradient as

$$\begin{aligned}\widehat{\boldsymbol{g}}_{\varphi_i} &\approx -\frac{\eta}{q}h\left(\boldsymbol{x}',\varphi_i,\alpha\right)h\left(\boldsymbol{x}'+\alpha\varphi_i,\psi_i,\alpha\beta\right)\psi_i, \\ \widehat{\boldsymbol{g}}_{\psi_i} &\approx -\frac{\eta}{q}h\left(\boldsymbol{x}',\psi_i,\alpha\right)h\left(\boldsymbol{x}'+\alpha\psi_i,\varphi_i,\alpha\beta\right)\varphi_i,\end{aligned} \tag{16}$$

where we denote $\varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)$ and $\psi\left(\boldsymbol{v}_i;\boldsymbol{w}\right)$ by φ_i and ψ_i, respectively. However, we derive Eq. (8) based on the randomly sampled searching directions, while the searching directions in (16) are optimized to decrease the efficient attack loss, which may be harmful to the convergence of these two transform functions (φ and ψ).

To alleviate the negative impact on the convergence of the transform functions, we adopt a residual connection of the search distribution that draws inspiration from (He et al., 2016). We model the residual search distribution by

$$\begin{aligned}\varphi^{res}\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right) &= \varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)+\boldsymbol{u}_i, \\ \psi^{res}\left(\boldsymbol{v}_i;\boldsymbol{w}\right) &= \psi\left(\boldsymbol{v}_i;\boldsymbol{w}\right)+\boldsymbol{v}_i.\end{aligned} \tag{17}$$

This formulation of the proposed residual search distribution implies that it has the properties of random directions and optimized directions.

3.3. Implementation

As DPD offers an efficient method of finding searching directions, it can be applied effortlessly to improve the efficiency of existing methods that require random searching directions. Algorithm 1 summarizes the dual-path distillation algorithm. Before we show how to apply our method, we first classify existing black-box attack methods into 3 categories as described in Table 1: methods without priors or a predefined distribution, methods with priors but no predefined distribution and methods with both priors and a predefined distribution. To demonstrate that dual-path distillation is applicable to all three categories, we select one of the most efficient attacks in each category to show that DPD can be easily applied.

For methods without priors or predefined distribution, we combine NES (Ilyas et al., 2018) with our framework, termed NES + DPD, we need to replace the original Gaussian distribution in NES with the proposed residual search distribution described in Eq.(17). Note that, although ZO-ADMM (Zhao et al., 2019) involves fewer model queries than that of NES (Ilyas et al., 2018), NES can achieve a higher attack success rate than ZO-ADMM. Hence, we regard NES as the baseline attack in this category.

For methods with priors but no predefined distribution, we apply our framework to BTD (Ilyas et al., 2019), termed BTD + DPD. BTD utilizes data- and time-dependent priors to reduce the query complexity. In effect, the data-dependent prior can be incorporated by reducing the dimensionality of the sampling directions. To the leverage time-dependent prior, BTD uses the last estimated gradient $\widehat{\boldsymbol{g}}_{\boldsymbol{x}}^{t-1}$ to update the estimated gradient $\widehat{\boldsymbol{g}}_{\boldsymbol{x}}^{t}$ with a correction direction (Δ). That is, $\widehat{\boldsymbol{g}}_{\boldsymbol{x}} = \widehat{\boldsymbol{g}}_{\boldsymbol{x}}^{t-1}+\eta\Delta$. Formally, $\widehat{\boldsymbol{g}}_{\boldsymbol{x}}^{t-1}$ is denoted as $\boldsymbol{d}$ and the correction factor (Ilyas et al., 2019) is estimated as follows:

$$\begin{aligned}\Delta &= \frac{f\left(\boldsymbol{x}'+\alpha\left(\boldsymbol{d}+\beta\boldsymbol{u}\right)\right)-f\left(\boldsymbol{x}'+\alpha\boldsymbol{d}\right)}{\alpha\beta}\boldsymbol{u} \\ &= h\left(\boldsymbol{x}'+\alpha\boldsymbol{d},\boldsymbol{u},\alpha\beta\right)\boldsymbol{u}.\end{aligned} \tag{18}$$

Algorithm 1 Pseudocode of dual-path distillation

1: **repeat**
2: Transform the sampled directions using (17)
3: Calculate the gradients used for updating adversarial examples according to (15)
4: Compute the gradients used for updating the transform function using (16)
5: Update the adversarial examples through (3)
6: Optimize the distribution transform functions using (10)
7: **until** meet the break criterion

Table 1. Differences between various kinds of black-box attacks.

methods	priors	predefined distribution
NES (Ilyas et al., 2018), BoundaryAttack (Brendel et al., 2018), ZO-ADMM (Zhao et al., 2019), ZOO (Chen et al., 2017)	✗	✗
Bandits$_{TD}$ (Ilyas et al., 2019), $\mathcal{N}$attack (Li et al., 2019), FD (Bhagoji et al., 2018)	✓	✗
P-RGF$_D$ (Cheng et al., 2019), LowFrequency (Guo et al., 2019), AutoZoom (Tu et al., 2019)	✓	✓

Substituting Eq. (18) into (6), we obtain

$$\widehat{\boldsymbol{g}}_{\boldsymbol{u}} \approx -\frac{\eta}{q} h\left(\boldsymbol{x}' + \alpha \boldsymbol{d}, \boldsymbol{u}, \alpha\beta\right) h\left(\boldsymbol{x}' + \alpha \boldsymbol{d} + \alpha\beta \boldsymbol{u}, \boldsymbol{v}, \alpha\beta^2\right) \boldsymbol{v}, \tag{19}$$

which is consistent with Eq. (8). In fact, Eq. (18) implies that if we perturb $\boldsymbol{x}'$ with $\alpha\boldsymbol{d}$ and scale the variance α to $\alpha\beta$ in Eq. (8), then we can obtain Eq. (19). The gradient $\widehat{\boldsymbol{g}}_{\boldsymbol{v}}$ can be estimated in a similar way. Then, we can employ $\widehat{\boldsymbol{g}}_{\boldsymbol{u}}$ and $\widehat{\boldsymbol{g}}_{\boldsymbol{v}}$ to update the utilized search distribution transform functions.

To the best of our knowledge, P-RGF$_D$, which has both priors and a predefined distribution, is the state-of-the-art method among all the categories. To demonstrate the effectiveness of our method, we evaluate dual-path distillation on P-RGF$_D$, and the improved method is termed P-RGF$_D$ + DPD, which can achieve the new state-of-the-art performance.

P-RGF$_D$ proposes using transfer-based priors to devise a covariance matrix and select searching directions that obey the constructed covariance matrix, where transfer-based priors represent the gradient of a surrogate model. According to (Cheng et al., 2019), the loss function for estimating gradients is formulated as

$$\min \ell\left(\widehat{\boldsymbol{g}}_{\boldsymbol{x}}\right) = \|\boldsymbol{g}_{\boldsymbol{x}}\|^2 - \frac{\left(\boldsymbol{g}_{\boldsymbol{x}}^T \mathrm{C} \boldsymbol{g}_{\boldsymbol{x}}\right)^2}{\left(1 - \frac{1}{q}\right) \boldsymbol{g}_{\boldsymbol{x}}^T \mathrm{C}^2 \boldsymbol{g}_{\boldsymbol{x}} + \frac{1}{q} \boldsymbol{g}_{\boldsymbol{x}}^T \mathrm{C} \boldsymbol{g}_{\boldsymbol{x}}}, \tag{20}$$

where $\mathrm{C} = \mathbb{E}\left[\boldsymbol{u}\boldsymbol{u}^T\right]$ is the covariance matrix, q denotes the number of sampled directions and $\boldsymbol{g}_{\boldsymbol{x}}$ is the gradient of the target model. To minimize $\ell\left(\widehat{\boldsymbol{g}}_{\boldsymbol{x}}\right)$, C is constructed with the gradient of a surrogate model.

In light of dual-path distillation, we can alter C directly according to the feedback knowledge. To this end, we simplify the loss based on the orthonormal assumption used in (Cheng et al., 2019). Then the loss takes the form of (see Supplementary for details)

$$\begin{aligned} &\min \ell\left(\widehat{\boldsymbol{g}}_{\boldsymbol{x}}\right) = -\frac{q}{D+q-1} \sum_{i=1}^{D} \left(\boldsymbol{g}_{\boldsymbol{x}}^T \boldsymbol{p}_{\boldsymbol{i}}\right)^2 \\ &\text{s.t.} \boldsymbol{p}_i^T \boldsymbol{p}_j = 0 \text{ for } i \neq j, \|\boldsymbol{p}_i\|_2 = 1 \text{ for } \forall i \end{aligned} \tag{21}$$

Table 2. Impact of different searching directions.

searching directions	$\boldsymbol{u}_i$ $\boldsymbol{v}_i$	$\varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)$ $\boldsymbol{v}_i$	$\varphi\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)$ $\psi\left(\boldsymbol{v}_i;\boldsymbol{w}\right)$	$\varphi^{res}\left(\boldsymbol{u}_i;\boldsymbol{\theta}\right)$ $\psi^{res}\left(\boldsymbol{v}_i;\boldsymbol{w}\right)$
ASR	98.4%	98.6%	99.3%	**99.9%**
AVG. Q	969	540	712	**482**

where $\boldsymbol{p}_i$ is the i^{th} eigenvector of C; D represents the dimension of inputs. It is noted from Eq. (21) that we need to train D networks, one for each eigenvector, which is impractical. To solve this problem, we employ a mix-up approach (Christopher Beckham & Pal, 2019) to train a network to generate many eigenvectors. Specifically, we synthesize features with different masks to generate different eigenvectors and use these eigenvectors to search the data space. From the viewpoint of DPD, the generated $\boldsymbol{p}_i$ is actually a searching direction. Then, we can use Eq.(8) to alter these directions. For a fair comparison, we also use the transfer-based priors in P-RGF$_D$ + DPD. In detail, we pretrain the network to attack a surrogate model and use the pretrained network for initialization.

4. Experiments

In this section, we first provide the evaluation methodology. Then, we show the effectiveness of the proposed residual search distribution through ablation experiments. After that, we present experimental results to demonstrate that our method can accelerate the efficiency of different black-box attack methods. Furthermore, we investigate the search distribution to present some interesting findings.

4.1. Methodology

Following previous methods (Cheng et al., 2019; Ilyas et al., 2019), we evaluate all the methods on 1000 images randomly sampled from the validation set of ImageNet (Deng et al., 2009). We perform targeted and untargeted attacks with both ℓ_2 and ℓ_∞ norms to highlight the fact that the proposed framework is effective in different settings. All the attacks are assessed on three normally trained models: Inception-v3 (Szegedy et al., 2016), VGG-16 (Simonyan & Zisserman, 2015), and ResNet-50 (He et al., 2016). All

Table 3. Results of **untargeted** black-box attacks against Inception-v3, VGG-16 and ResNet-50 **without defense**. We report the attack success rate (ASR) and the average number of queries (AVG. Q).

Methods	l_2-norm						l_∞-norm					
	Inception-v3		VGG-16		ResNet-50		Inception-v3		VGG-16		ResNet-50	
	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓
NES (Ilyas et al., 2018)	95.5%	1718	98.7%	1081	98.4%	969	87.5%	1850	95.6%	1477	94.5%	1405
NES + DPD	**97.1%**	**1035**	**99.2%**	**402**	**99.9%**	**482**	**94.8%**	**1063**	**98.7%**	**635**	**99.8%**	**814**
Bandits$_{TD}$ (Ilyas et al., 2019)	97.2%	874	94.9%	278	96.8%	512	94.7%	**1099**	95.1%	**288**	96.5%	651
Bandits$_{TD}$ + DPD	**98.6%**	**846**	**100.0%**	**216**	**100.0%**	**243**	**97.0%**	1204	**99.8%**	445	**99.9%**	**522**
P-RGF$_D$ (Cheng et al., 2019)	99.1%	649	99.7%	370	99.6%	352	97.3%	812	99.6%	433	99.6%	452
P-RGF$_D$ + DPD	**99.5%**	**263**	**100.0%**	**36**	**100.0%**	**37**	**98.7%**	**641**	**99.8%**	**162**	**99.8%**	**162**

Table 4. Results of **targeted** black-box attacks against Inception-v3, VGG-16 and ResNet-50 **without defense**. We report the attack success rate (ASR) and the average number of queries (AVG. Q).

Methods	l_2-norm						l_∞-norm					
	Inception-v3		VGG-16		ResNet-50		Inception-v3		VGG-16		ResNet-50	
	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓
NES (Ilyas et al., 2018)	22.1%	7350	83.4%	6224	60.5%	6540	7.9%	4866	24.5%	5083	16.8%	5017
NES + DPD	**30.5%**	**4173**	**93.0%**	**2552**	**90.0%**	**3241**	**27.5%**	**4343**	**67.0%**	**4038**	**70.5%**	**4531**
Bandits$_{TD}$ (Ilyas et al., 2019)	68.9%	4637	93.5%	2383	92.1%	3185	17.4%	5792	46.1%	5299	39.1%	5003
Bandits$_{TD}$ + DPD	**70.8%**	**3970**	**95.0%**	**2221**	**93.5%**	**3084**	**28.0%**	**3882**	**68.5%**	**3825**	**71.1%**	**4216**
P-RGF$_D$ (Cheng et al., 2019)	73.0%	5011	74.1%	3207	71.2%	2750	29.0%	4228	75.6%	3257	74.0%	3025
P-RGF$_D$ + DPD	**80.6%**	**2768**	**99.1%**	**1002**	**99.9%**	**580**	**54.8%**	**3970**	**88.3%**	**3010**	**91.7%**	**2813**

these networks are provided by torchvision (Marcel & Rodriguez, 2010). In addition, we perform attacks against a defensive approach, JPEG compression (Guo et al., 2018), because JPEG compression is the most robust approach used in (Cheng et al., 2019). Following the experimental protocol in (Cheng et al., 2019), we set the maximum distortion $\epsilon = \sqrt{0.001 \cdot D}$ ($\epsilon = 0.05$) under ℓ_2 (ℓ_∞), with images scaled to $[0, 1]$, where D is the dimension of the inputs.

Following previous works (Cheng et al., 2019; Ilyas et al., 2019; Zhao et al., 2019), we limit the maximum number of queries for each image to be $10,000$ and report both the attack success rate (ASR) and the average number of queries (AVG. Q). A successful attack means that the constructed adversarial example can fool the target model before the maximum number of queries is reached. Note that we calculate the average number of queries over successful attacks, which is a method widely used in the literature (Ilyas et al., 2018; Cheng et al., 2019; Ilyas et al., 2019; Zhao et al., 2019).

4.2. Ablation Experiments

We compare the proposed residual search distribution with the different search distributions mentioned above. All these approaches are evaluated on untargeted attacks with the ℓ_2-norm, and the target model is set to ResNet-50.

The results are presented in Table 2. $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ means using searching directions that are randomly sampled from a Gaussian distribution. $\varphi(\boldsymbol{u}_i;\boldsymbol{\theta})$ and $\psi(\boldsymbol{v}_i;\boldsymbol{w})$ denote applying the transform functions to the randomly selected searching directions $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, respectively. $\varphi^{res}(\boldsymbol{u}_i;\boldsymbol{\theta})$ and $\psi^{res}(\boldsymbol{v}_i;\boldsymbol{w})$ are the proposed residual search distributions defined in Eq.(17).

We find that the searching directions generated by the residual search distribution outperform all the other searching directions. On the other hand, the random directions are the worst solution. These results are reasonable, because the transformed searching direction contains more knowledge distilled from the target model. Moreover, the performance in the third column is worse than that in the last column, which indicates the effectiveness of our residual approach. Therefore, we employ residual search distributions in the following experiments.

4.3. Results

The results of the **untargeted attacks** are summarized in Table 3. We find that our method improves these methods significantly. NES + DPD achieves a 99.9% attack success rate when ResNet-50 is attacked, while the success rate of P-RGF$_D$ is 99.6%. That is, NES + DPD can achieve a comparable performance with or even better performance than prior-based methods which are the current state-of-the-art methods. This finding is promising since NES + DPD uses no priors, while P-RGF$_D$ employs transfer-gradient and data-dependent priors. In addition, only **36** model queries are required for P-RGF$_D$ + DPD to mount a successful attack to VGG-16 under the ℓ_2 norm.

We also evaluate the effectiveness of different methods on the more challenging **targeted attack** and display the re-

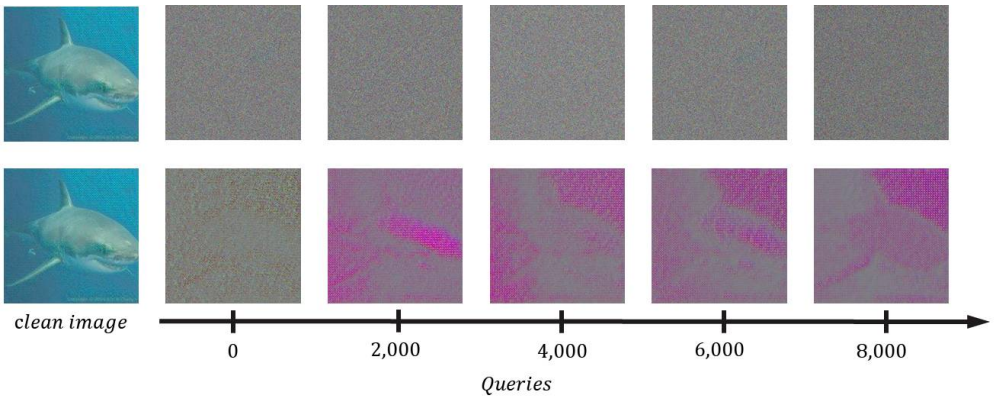

Figure 2. The generated searching directions w.r.t. the number of queries. The searching directions in the first line are sampled from the Gaussian distribution used in the NES, while directions in the second line are generated by our residual search distribution.

Table 5. Results of **untargeted** black-box attacks against Inception-v3, VGG-16 and ResNet-50 **with defense**. We report the attack success rate (ASR) and the average number of queries (AVG. Q).

Methods	l_2-norm						l_∞-norm					
	Inception-v3		VGG-16		ResNet-50		Inception-v3		VGG-16		ResNet-50	
	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓	ASR ↑	AVG. Q ↓
NES (Ilyas et al., 2018)	18.7%	2540	46.3%	640	23.8%	1870	17.0%	1251	42.1%	644	16.8%	1305
NES + DPD	**28.7%**	**1057**	**81.3%**	**330**	**56.8%**	**1066**	**27.4%**	**772**	**61.3%**	**597**	**36.4%**	**758**
Bandits$_{TD}$ (Ilyas et al., 2019)	23.1%	**774**	44.2%	**209**	34.8%	**679**	33.6%	161	46.3%	**472**	33.1%	**320**
Bandits$_{TD}$ + DPD	**33.5%**	982	**82.3%**	667	**56.8%**	803	**41.2%**	**155**	**64.5%**	558	**43.8%**	587
P-RGF$_D$ (Cheng et al., 2019)	50.2%	1620	73.6%	1681	72.4%	1857	32.1%	423	46.1%	381	42.8%	539
P-RGF$_D$ + DPD	**68.0%**	**194**	**90.6%**	**41**	**90.8%**	**41**	**64.3%**	**258**	**82.5%**	**195**	**82.7%**	**210**

sults in Table 4. The results demonstrate that dual-path distillation can drastically increase the attack success rate of previous methods. In particular, the proposed framework can increase the attack success rate of P-RGF$_D$ from 29.0% to 54.8% when attacking Inception-v3 under the ℓ_∞ norm. In addition to the improvement in the attack success rate, the average numbers of queries are also reduced significantly in all settings.

Table 5 shows the performance evaluated on the **defense approach** (Guo et al., 2018). Although the defensive method makes the target model more robust against adversarial examples, applying dual-path distillation can still increase the attack success rate and reduce the required number of queries considerably. We find that even in the most difficult setting in our experiments (attack Inception-v3 under the ℓ_∞-norm), our method can increase the success rate from 17.0% to 27.4% and reduce the number of queries from 1251 to 772.

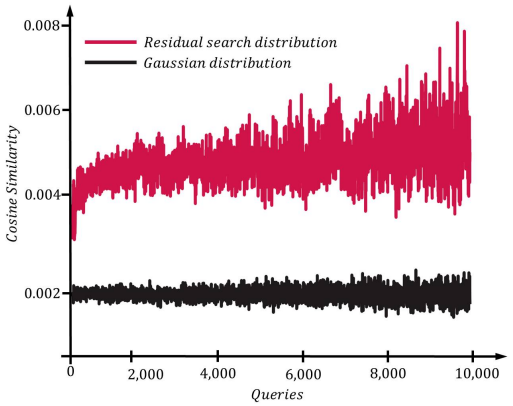

Figure 3. The averaged cosine similarity between the searching directions and real gradients. Here, we average the absolute value of the cosine similarity, because the mean of uniform directions is zero.

4.4. Discussion

To further demonstrate the effectiveness of the proposed dual-path distillation method, we show some details of the learned search distributions in this section.

First, we attempt to investigate the knowledge encoded in the residual search distribution. Intuitively, "appropriate" searching directions should be dependent on the inputs. A qualitative analysis is provided in Fig. 2. With the increase in the number of model queries, searching directions gradually omit some irrelevant pixels and contain more information that is similar to the object in the input image.

In addition to qualitative analyses, we provide one quantitative description for discovering the knowledge encoded in the search distribution. From the finite difference perspective, which is also used in (Ilyas et al., 2019), we need to find sufficient components to reconstruct the real gradients. Thus, the high cosine similarity between the searching directions and the real gradients means that we can sample fewer directions to approximate the real gradients. Fig. 3 shows the cosine similarity between the sampled direction and the real gradient. The result indicates that the directions generated by the residual search distribution are more similar to the real gradients than those of the randomly generated searching directions.

In summary, dual-path distillation transfers feedback knowledge to residual search distributions, which endows the sampled directions with two properties: correlation with the content of the inputs and high cosine similarity to the real gradients.

5. Conclusion

To improve black-box attacks, we formulate the problem of efficient black-box attacks and introduce dual-path distillation, which can be effortlessly applied to the existing works. Dual-path distillation increases query efficiency by making use of feedback knowledge. In addition, we study different knowledge utilization approaches and propose a residual search distribution to further mitigate knowledge leakage. Experimental results demonstrate that our method can significantly improve existing black-box attacks and achieve state-of-art performance. Dual-path distillation takes a further step to eliminate the overhead cost of black-box attacks, which can provide a new option to efficiently evaluate defense approaches.

Acknowledgements

This work was supported by National Key Research and Development Program of China under Grant 2017YFB1002203 and the National Natural Science Foundation of China under Grant 61872329.

References

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ICML*, 2018.

Bhagoji, A. N., He, W., Li, B., and Song, D. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *European Conference on Computer Vision*. Springer, 2018.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *ICLR*, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26. ACM, 2017.

Cheng, S., Dong, Y., Pang, T., Su, H., and Zhu, J. Improving black-box adversarial attacks with a transfer-based prior. *NeurIPS*, 2019.

Christopher Beckham, Sina Honari, A. L. V. V. F. G. R. D. H. Y. B. and Pal, C. On adversarial mixup resynthesis. In *NeurIPS*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *CVPR*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Guo, C., Rana, M., Cisse, M., and Van Der Maaten, L. Countering adversarial images using input transformations. *ICLR*, 2018.

Guo, C., Frank, J. S., and Weinberger, K. Q. Low frequency adversarial perturbation. *AAAI*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., and Choi, J. Y. A comprehensive overhaul of feature distillation. In *ICCV*, 2019.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018.

Ilyas, A., Engstrom, L., and Madry, A. Prior convictions: Black-box adversarial attacks with bandits and priors. *ICLR*, 2019.

Kim, J., Park, S., and Kwak, N. Paraphrasing complex network: Network compression via factor transfer. In *NeurIPS*, 2018.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2017.

Lax, P. D. and Terrell, M. S. *Calculus with applications*. Springer, 2014.

Li, Y., Li, L., Wang, L., Zhang, T., and Gong, B. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. *ICML*, 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Marcel, S. and Rodriguez, Y. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1485–1488, 2010.

Moon, S., An, G., and Song, H. O. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. *ICML*, 2019.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 2017.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *ACM on Asia conference on computer and communications security*. ACM, 2017.

Poursaeed, O., Katsman, I., Gao, B., and Belongie, S. Generative adversarial perturbations. In *CVPR*, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Tu, C.-C., Ting, P., Chen, P.-Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.-J., and Cheng, S.-M. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *AAAI*, 2019.

Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. Natural evolution strategies. In *Evolutionary Computation*, 2011.

Ye, H., Huang, Z., Fang, C., Li, C. J., and Zhang, T. Hessian-aware zeroth-order optimization for black-box adversarial attack. *arXiv preprint arXiv:1812.11377*, 2018.

Zhang, Y., Tian, X., Li, Y., Wang, X., and Tao, D. Principal component adversarial example. *IEEE Transactions on Image Processing*, 29:4804–4815, 2020.

Zhao, P., Liu, S., Chen, P.-Y., Hoang, N., Xu, K., Kailkhura, B., and Lin, X. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. In *ICCV*, pp. 121–130, 2019.