Query-Efficient Black-Box Adversarial Attack with Random Pattern Noises

Makoto Yuito, Kenta Suzuki, and Kazuki Yoneyama(✉)

Ibaraki University, Hitachi, Japan
kazuki.yoneyama.sec@vc.ibaraki.ac.jp

Abstract. Adversarial examples are one of the largest vulnerability of deep neural networks. An attacker can deceive the classifiers easily with the malicious inputs (called adversarial examples), which perturbations are slightly added to benign inputs. Various attack methods have been studied in both white-box and black-box settings, and some methods achieve high attack success rates even in the black-box settings; that is, the attacker is restricted to only query accesses to the target network. In this paper, we propose a simple hyperparameter-free score-based black-box ℓ_∞-adversarial attack using local uniform noises and a random search. Specifically, we construct adversarial perturbations by combining local uniform noises such as vertical-wise and horizontal-wise, and incorporate this idea into the random search method to update the perturbation sequentially. We evaluate our method in terms of attack success rates and query efficiency using models that classify common datasets CIFAR-10 and ImageNet. We show that our method achieves higher attack success rates and query efficiency than previous attack methods, especially in low-query budgets on both untargeted and targeted attack settings. We also examine attacks to adversarially trained models and discuss the effect of local uniform noises on these models. Furthermore, we show that our method achieves relatively high attack success rates and query efficiency on average against input-transformation-based defense methods, and is virtually unaffected by these defense methods.

Keywords: Black-box adversarial attacks · AI security

1 Introduction

1.1 Backgrounds

Due to recent breakthroughs in deep learning techniques, Deep Neural Networks (DNNs) have achieved state-of-the-art classification performance in various tasks. However, it has also been shown that the classification models can still be easily affected by adversarial examples [4,5,7,8,15,28,30,32] which are malicious inputs such that small perturbations are added to benign inputs in order to fool the classifiers. Adversarial attacks can cause serious security problems

Makoto Yuito—Presently, he is with IVIS, Inc.

C. Alcaraz et al. (Eds.): ICICS 2022, LNCS 13407, pp. 303–323, 2022.
https://doi.org/10.1007/978-3-031-15777-6_17

because DNNs are deployed in the real world in various applications. For example, Deng et al. [12] analyze adversarial attacks on driving models, and show that these regression models are also very vulnerable to adversarial attacks. Sharif et al. [35] show that it is possible to impersonate another individual by having the face image wear glasses, as in Adversarial Patch [6]. Therefore, in order to design robust models, it is necessary to investigate the potential risks and identify the vulnerabilities of deep learning models. Hence adversarial attacks are an important research topic.

If an adversarial example of an image x exists, attacking a classifier turns into a search problem within a small volume around a benign image x. Recently, several algorithms have been proposed to generate adversarial examples, and these methods can be classified based on several categories.

Threat Model: One of the key differences in adversarial attacks is the setting of the attacker, and there are two primary types: white-box and black-box. In the white-box setting [7,15,28,32], the attacker is assumed to have all the knowledge about the target model. The main idea of generating adversarial examples in this setting is to apply a perturbation in the direction of the gradient of the loss w.r.t. the input x. However, in reality, an attacker is likely to have access to only a limited amount of information. In the black-box setting [4,5,8,23,30], the attacker is only allowed query access to the target model. That corresponds to an attack on a web service using a pre-trained classifier (e.g., Google Cloud Vision API [2], IBM Watson Visual Recognition [3], Amazon Rekognition [1]). In this setting, the attacker needs to compute a perturbation only from the output information obtained by querying a model, which is thus more difficult setting. The main strategies for generating adversarial examples in the black-box setting are shown in Sect. 2.

Adversarial Goal: Another important difference in adversarial attacks is whether the attacker aims to misclassify the input x to a class other than the true class y (untargeted), or to misclassify the classification result to a specific target class $t(\neq y)$ (targeted). Targeted attacks, especially on classifiers with a large number of classes, are quite a difficult task.

Distance Metric: Adversarial examples are inputs with slight perturbations that are carefully crafted to cause the classifier to misclassify them. It is commonly used ℓ_p-distances between adversarial and benign examples with $p \in \{0, 2, \infty\}$.

We focus on score-based black-box adversarial attacks. Existing query-based black-box attack methods have already achieved a high attack success rate, and the main effort is now focusing on reducing the number of queries. Attacks with low queries, i.e., methods with better query efficiency, can save attackers a great deal of cost in both time and money. For example, the Google Cloud Vision API [2] limits the number of requests per minute to $1,800$. High query efficiency attack methods are also effective in deceiving systems [10] that recognize the behavior of submitting many similar queries in short time as fraudulent, which is one of our motivations.

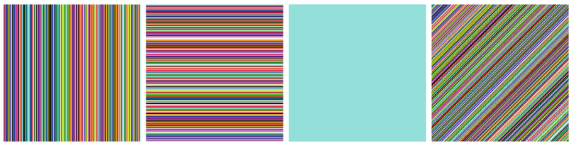

Fig. 1. Sample images of each random pattern noise (RPN) in our sampling space (from the left is vertical-wise, horizontal-wise, uniform, diagonal-wise local uniform noise).

1.2 Our Contribution

In this paper, we propose a simple but effective hyperparameter-free score-based black-box ℓ_∞-adversarial attack in computer vision. The core technique of our approach is to use susceptibility of Convolutional Neural Networks (CNNs) to noise with regional homogeneity [24,46], and specifically to construct adversarial perturbations by combining patterned noises such as vertical-wise and horizontal-wise (see Fig. 1). This idea is incorporated into an iterative random search method to sequentially update the perturbations. In a pre-specified non-orthogonal search direction, we modify the perturbation with randomly selected local uniform noises, check whether it is moving towards or away from the decision boundary using a confidence score, and repeat the perturbation update. With each update, the image moves further away from the original image and towards the decision boundary.

In Sect. 4, we conduct comparative experiments with several existing ℓ_∞-attacks using naturally and adversarially trained models and input-transformation-based defense methods.

In the experiments on the naturally trained models in Sect. 4.1, we use CIFAR-10 and ImageNet datasets to perform comparative experiments with Parsimonious, SignHunter and Square Attack. As a result, we show that our method achieves high attack success rates in both untargeted and targeted attack settings, especially in low query budgets. Specifically, in the untargeted attack on CIFAR-10, our method achieves the average query efficiency of 1.8 times while achieving a higher attack success rate than that of Square Attack. In the untargeted attack on ImageNet, our method also achieves 1.4 times higher average query efficiency than that of Square Attack.

In Sect. 4.2, we evaluate our method against several defensive models based on adversarial training that classify MNIST and CIFAR-10 datasets. In the benchmark Madry et al.'s and TRADES models on MNIST, our method achieves higher attack success rates than the other black-box methods. However, in other Clean Logit Pairing (CLP) and Logit Squeezing (LSQ) models, the results of our method are inferior to those of other black-box attacks, especially in terms of attack success rate. From this result we clarify the effect of local uniform noise in each defensive model.

In Sect. 4.3, we show attacks to several input-transformation-based defense methods that adopt the naturally trained models classifying CIFAR-10 and ImageNet as a backbone. Our method achieves an attack performance of over 90% on CIFAR-10 and over 70% on ImageNet, despite relatively small query budgets. Therefore, our method maintains a high attack success rate with or without the protection of defense methods.

Overall, our method achieves high attack performance on a wide range of target models in a hyperparameter-free manner, making it a realistic method for attackers. We also observe that our method suffers from gradient masking, and our definition of local uniform noise is highly convergent for defensive models other than gradient masking. Finally, in Sect. 4.4, we experimentally verify the effectiveness of our definition of local uniform noises and show that all of them contribute to the attack performance.

2 Related Work

There are a few different settings for adversarial attacks in the black-box setting. This section describes the differences between these settings and the main strategies. Then, we show our contribution by comparing with them.

2.1 Transfer-Based Black-Box Attacks

Most of the existing adversarial attacks assume the white-box setting, where the attacker has full access to the model architecture and the ability to perform backpropagation to obtain gradient information. On the other hand, white-box attacks can be pseudo-black-boxed by using transferability [38], called transfer-based black-box attacks. Transferability is a property that adversarial examples generated for a classifier can be used as for another same type classifiers. Papernot et al. [31] proposed a method to learn a surrogate model by querying the target model. By using the surrogate model with decision boundaries similar to the target model, they can simulate a white-box adversarial attack [15,32]. However, transfer-based attacks have some problems. First, although transfer-based attacks are theoretically possible in a decision-based setting, they often require carefully designed surrogate models, or even require many queries to extract the target model. Next, the generated adversarial examples do not always transfer well [36]. Recent studies have also proposed input transformation methods [13,25,42] to improve the transferability of adversarial examples, and showed black-box attack performances. Although such a method [25] achieves particularly high transferability, they ignore the task of extracting models and only show the attack success rates between each network architecture.

2.2 Score-Based Black-Box Attacks

In score-based black-box attacks, the attacker can obtain the predicted probabilities for each class by querying the inputs to the target model. The attacker

solves an optimization problem to compute the adversarial perturbations while directly observing the output from the target model.

Gradient Estimation Based Methods. The ZOO method, proposed by Chen et al. [9], generates adversarial examples by estimating the gradient of the classifier using a coordinate-wise finite difference method. The AutoZOOM, a modified version of ZOO, was proposed by Tu et al. [39], which uses random gradient estimation and dimensionality reduction techniques to significantly improve query efficiency while maintaining attack performance. However, it still requires an enormous number of queries to the target model (13,525 queries on average for the targeted attack on ImageNet). Hence, gradient estimation-based methods are considerably less efficient, especially for models with high-dimensional inputs.

Gradient-Free Methods. The Parsimonious Attack proposed by Moon et al. [30] solves a discrete optimization problem with local search and the greedy algorithm. On a perturbation divided into a set of n^2 square tiles, Parsimonious finds the sign of each tile by local search, and then uses the greedy algorithm to find a better solution. The SignHunter Attack proposed by Al-Dujaili et al. [4] sequentially estimates the sign of gradient in $1/2^n$ regions of the perturbation in deterministic order. Several attack methods including these [4,29,30] reduce the dimensionality of the search space of the perturbation by modifying neighboring pixels in the perturbation at once, making the computation more efficient. Andriushchenko et al. proposed the Square Attack [5], which achieved state-of-the-art attack success rates and query efficiency. Square Attack solves optimization problems by random search, which directly updates the perturbation with randomly generated square-shaped noise, as opposed to methods that invert the sign of the perturbation, such as Parsimonious and SignHunter. The DeepSearch proposed by Zhang et al. [47] generates adversarial examples close to the original images by reducing the ℓ_∞ distance of the perturbation, while using hierarchical grouping strategy like Parsimonious. However, we do not compare our method with DeepSearch because the attack success rate and query efficiency are not high (similar to those of Parsimonious) although the ℓ_∞ distance of the perturbation generated by DeepSearch is small.

On the other hand, several studies have improved query-based attacks, in which the attacker generates adversarial examples in transfer-based and query-based manner using a surrogate white-box model that is either pre-trained or trained by the attacker himself. The Subspace Attack by Guo et al. [18] uses the gradient of the surrogate model as a heuristic search direction for finite difference gradient estimation. Huang et al. proposed TREMBA [19], which learns an embedding space that can generate adversarial perturbations for a surrogate model, and significantly reduces queries compared to NES and AutoZOOM. Feng et al. [14] improved the transfer performance from the surrogate model to the target model. Their proposed $\mathcal{CG}$-Attack is robust to biases between the surrogate model and the target model by transferring partial parameters of the adversarial distribution of the surrogate model while learning the untransferred

parameters based on queries to the target model. The SWITCH proposed by Ma et al. [27] continues to select loss-maximizing perturbations whenever possible when images perturbed by gradients generated from a surrogate model do not satisfy the optimization objective. Yatsura et al. proposed a meta-learning method [45] to be used in combination with random search based attacks. Their learned controller improves the attack performance by online adjustment of the parameters of the proposal distribution at each iterate during the attack. However as explained in Sect. 2.1, we do not compare our method to these methods since the attacker needs to construct a surrogate model in advance and the computational cost is high.

2.3 Defense Methods

As adversarial attacks become more prevalent, many recent studies have also focused on building defense models against them. There are several lines of research in the literature, and the defense methods are roughly consisted of two groups: input-transformation-based defense methods and adversarial training.

The input-transformation-based defense methods include denoising, input randomization, and input transformation. These methods attempt to mitigate the effects of perturbations in adversarial examples by adding image processing-like changes to an input image. Specifically, the denoising methods include low-pass filtering [34] and autoencoders [16], which attempt to remove adversarial perturbations from adversarial examples. The input randomization methods including resizing and padding [41] and the input transformation methods including JPEG Compression [17,26] attempt to mitigate the effect of adversarial perturbations.

On the other hand, adversarial training [21,28,48] aims to obtain robustness by training the model with adversarial examples, which is a more costly but more effective method than image processing defenses. In general, it is known that adversarial training defenses are more robust than other defenses in the case of MNIST and CIFAR-10. Furthermore, Madry et al. [28] show that PGD [28] is a universal first-order adversarial attack, which means that adversarial training with PGD-generated adversarial examples is resistant to many other first-order attacks. The PGD-generated adversarial examples are the basis for many adversarially trained models, including [21,28,48]. The model of Madry et al. [28] provides robust adversarial training by min-max optimization. TRADES [48] focuses on the trade-off between robust error and natural error and trains to improve both. Adversarial Logit Pairing [21] learns by matching the logit of a benign image with the corresponding logit of adversarial examples, while acquiring ancillary information such as their similarity to each other.

2.4 Differences Among Other Black-Box Methods and Our Method

We discuss more about the existing methods presented in Sect. 2.2 and clarify the differences between them and our method. First, regarding the optimization method of the perturbation, it can be observed that Parsimonious [30] has

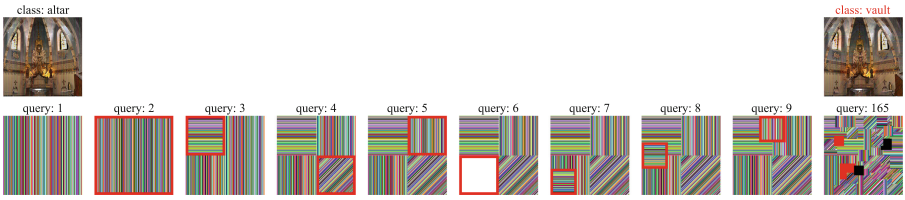

Fig. 2. An example of a sequence of adversarial perturbations on ImageNet generated by our method at each iterate. The left column shows the adversarial perturbation and the adversarial example for the first query (the attack has not yet succeeded at this point), and the right column shows those for the 165th query where the attack was successful (the class changed from *altar* to *vault*). In addition, the transition of adversarial perturbations after the first query is shown between them. The red boxes indicate the block range b determined by the SplitBlock function, i.e., the region where the noise is modified at each iterate. In the second query, we change the perturbation with a randomly picked RPN for a 1×1 region, i.e., the entire image region, and if the loss is lowered, we update the perturbation to this. In the third to sixth queries, the search is performed in 2×2 regions. After that, the perturbation update process is repeated while gradually increasing the number of segmented regions. (Color figure online)

many useless queries, partly because it uses the local search. SignHunter [4] is a deterministic search and can guarantee the attack success rate for the number of queries, but it is not very efficient. Since the convergence of the iterative random search used in Square Attack [5] is much higher than that of Parsimonious and SignHunter, an iterative random search is also used in our method.

As for the components of the perturbation, the perturbation of Parsimonious and SignHunter consist of a uniform noise in a specific segmentation range (square or rectangle shape), while the perturbation of Square Attack consists of a vertical-wise initialization and a uniform noise of a square of a certain size. On the other hand, our method places not only square-shaped but also vertical-wise, horizontal-wise and diagonal-wise uniform noise on the segmented area of squares in the image. Furthermore, while Parsimonious and Square Attack have hyperparameters that need to be tuned depending on the setting of the attack and the target model, our method does not need any hyperparameters. This feature is a great advantage in black-box attacks because it can be easily implemented in any setting.

3 Our Methods

In this section, we first recall the definitions of the threat model in the adversarial attacks and describe an optimization framework for finding adversarial perturbations against classification models. Then, we describe our black-box ℓ_∞-adversarial attack using random pattern noises and random search.

3.1 Optimization Framework

Formally, we define a classifier $f : X \to \mathbb{R}^K$ where $x \in X$ is the input image, $y \in Y = \{1, 2, \ldots, K\}$ is the output space and $f(x)$ denotes the predicted score of each class in Y. In the untargeted setting, the goal of the attacker is to find a perturbation δ such that an adversarial example $(x+\delta)$ is misclassified to classes other than the true class y, i.e., $\arg\max_{k \in Y} f_k(x+\delta) \neq y$. Additionally, the attacker also seeks to minimize ℓ_p distance, i.e.,

$$\arg\max_{k \in Y} \; f_k(x+\delta) \neq y \quad \text{s.t.} \quad \|\delta\|_p \leq \epsilon \text{ and } (x+\delta) \in X, \tag{1}$$

where $\|\cdot\|_p$ is the ℓ_p-distance norm function and ϵ is the radius of ℓ_p-ball. The task of finding a perturbation δ can be handled as a constrained optimization problem. Therefore, ℓ_p-bounded untargeted attacks aims at optimizing the following objective:

$$\min_{\delta : \|\delta\|_p \leq \epsilon} \; L(f(x+\delta), y) \tag{2}$$

where L is a loss function (typically the cross-entropy loss) and y is the true label of x. Equation 2 mostly works to minimize the score for label y. We also study the adversary in targeted setting. In the targeted setting, the attacker aims $\arg\max_{k \in Y} f_k(x+\delta) = t$ for a target label $t (\neq y)$ chosen from Y and optimizes the perturbation by minimizing the loss $L(f(x+\delta), t)$. A black-box targeted attack on a network with many output classes (large K) will be a rather difficult task.

3.2 Algorithm

In this section, we present our black-box ℓ_∞-attack. We assume that the attacker has an image $x \in X$ and a black-box classifier f. An output $f(x)$ is the predicted probabilities over K-classes w.r.t. input image x. In the untargeted setting, our goal is to find a perturbation $\delta \in \{-\epsilon, \epsilon\}^d$ such that $\arg\max \; f(x+\delta) \neq y$ under the ℓ_∞-perturbation constraint, where $\epsilon \in \mathbb{R}^+$ is the radius of ℓ_∞-ball. Our method is based on a random search [33] which is a well known iterative technique in optimization problems. If we apply this technique to the adversarial attacks, it acts as sequential updates of the perturbation. If the loss value $L(f(x+\delta^*), y)$ w.r.t. the perturbed image $(x+\delta^*)$ with the updated perturbation δ^* is lower than the prior loss value $L(f(x+\delta), y)$, this update is adopted to the current perturbation, otherwise it is discarded.

The core technique of our approach is that the perturbation is composed of noises with regional homogeneity. There are studies [24,46] showing the vulnerability of CNNs to local uniform noises. In particular, Li et al. [24] investigate how effective local homogeneous noise is for defensive models against adversarial attacks. They find that adversarial perturbations made for defensive models exhibit more homogeneous patterns than those made for naturally trained models. We therefore investigate whether local homogeneous noises can be applied

Algorithm 1. Our Method with Random Search

Input: classifier f, original image $x \in X$, true class y, image size w, image channels c, ℓ_∞-radius ϵ, max number of iterations N
Output: adversarial perturbation $\delta \in \{-\epsilon, \epsilon\}^d$

$\delta \leftarrow$ initial perturbation (vertical-wise),
$x_{adv} \leftarrow (x + \delta).Clip(0, 1)$
$l \leftarrow L(f(x_{adv}), y),\ i \leftarrow 1$
if *attack is already successful* **then**
  break
end if
$\mathcal{B} \leftarrow \mathsf{SplitBlock}(w)$
while $i < N$ **and** *attack is not successful* **do**
  $b \leftarrow \mathcal{B}^{(i\%len(\mathcal{B}))}$
  $\delta^* \leftarrow \mathsf{RPNSampling}(\delta, b, w, c, \epsilon)$
  $x_{adv} \leftarrow (x + \delta^*).Clip(0, 1)$
  $l^* \leftarrow L(f(x_{adv}), y)$
  if $l^* < l$ **then**
    $\delta \leftarrow \delta^*,\ l \leftarrow l^*$
  end if
  $i \leftarrow i + 1$
end while

Algorithm 2. RPNSampling

Input: perturbation δ, block area to be modified b, image size w, image channels c, ℓ_∞-radius ϵ
Output: new updated $\delta^* \in \{-\epsilon, \epsilon\}^d$

$\delta^* \leftarrow \delta$
sample RPN uniformly $\gamma \in \{\delta_{vert}, \delta_{horiz}, \delta_{uni}, \delta_{diag}\}$
for $i = 1, \ldots, c$ **do**
  $\delta^*_{b,i} \leftarrow \gamma_{b,i}$
end for

Algorithm 3. SplitBlock

Input: image size w
Output: a sequence of block areas $\mathcal{B}$

$\mathcal{B} = \emptyset$
for $i = 1, \ldots, w$ **do**
  Split the whole area of image into i^2 square shaped blocks $\{b_1, b_2, \ldots, b_{i^2}\}$ with size w/i
  $\mathcal{B} \leftarrow \mathcal{B} \cup$ shuffled $\{b_1, b_2, \ldots, b_{i^2}\}$
end for

to generate adversarial examples (Note that, they [24] aim to generate universal adversarial perturbations, which is a deceptive perturbation for arbitrary images, and is a different objective from ours, so it is not comparable). Specifically, our method constructs perturbations with four patterned noises: vertical-wise, horizontal-wise, uniform, and diagonal-wise (henceforth, collectively referred to as random pattern noise, RPN). This represents a major difference from Square Attack [5], which updates perturbations only with uniform noise in the form of squares.

Algorithmic Scheme with Random Search. Our proposed schemes are presented in Algorithms 1, 2 and 3. First, we set a initial perturbation to the vertical-wise one. A vertical-wise initialization is a technique used in [5]. Then, we obtain the current loss by querying the perturbed image $(x+\delta)$. Since we are interested in query efficiency, the algorithm stops as soon as an adversarial perturbation is found. Therefore, the process is terminated if the attack is already successful

Table 1. Results of both untargeted and targeted attacks on Madry et al.'s naturally trained model [28] classifying CIFAR-10. We set the norm bound $\epsilon_\infty = 0.031$ and a limit of queries to 10 k.

Attack	Success rate		Avg. queries		Med. queries	
	Untargeted	Targeted	Untargeted	Targeted	Untargeted	Targeted
Parsimonious [30]	93.3%	97.3%	329	631	244	476
SignHunter [4]	88.9%	95.6%	157	370	73	311
Square Attack [5]	93.0%	96.7%	131	354	67	253
Ours	**96.4%**	**98.3%**	**72**	**242**	**28**	**132**

at the first query point (step 3 in Algorithm 1). After that, we decide the set of block areas to be modified using the SplitBlock algorithm in Algorithm 3. In a random search loop, first the algorithm picks a block area b and obtains the new perturbation δ^* updated for the area through RPNSampling in Algorithm 2. Then, an adversarial example x_{adv} is generated by adding the perturbation to the benign image. Note that, all perturbed images are clipped in the domain $[0, 1]^d$. If the resulting loss corresponding to the perturbed image $(x + \delta^*)$ with the updated perturbation is lower than the current loss, the change is applied. The process is performed at most N (the maximum number of iterations) times and the attack is failure if we cannot find the adversarial perturbation until N times. Figure 2 shows a sequence of candidates of adversarial examples at each iterate generated by our method. A candidate is generated at each iterate, and the perturbation is updated if the loss at that time is lower than the previous one.

RPN Sampling. Our RPNSampling algorithm presented in Algorithm 2 returns a new perturbation δ^* updated for a given block area b to be modified. As the variation of RPNs, we focus on vertical-wise, horizontal-wise, uniform and diagonal-wise perturbations. We show the samples of each RPN in Fig. 1. In this algorithm, one of the four RPNs $\delta_{vert}, \delta_{horiz}, \delta_{uni}, \delta_{diag} \in \{-\epsilon, \epsilon\}^d$, which are randomly generated each time, is sampled as γ. The algorithm then changes the new perturbation δ^* to γ only in the region of block area b. From Fig. 2, it can be observed that one of the randomly generated RPNs is picked at each iterate and changed to that RPN only in certain regions. The effectiveness of each RPN is experimentally verified in Sect. 4.4.

Split Block. The SplitBlock algorithm shown in Algorithm 3 returns a set of elements which are block areas to be modified in the perturbation. The purpose of this function is to decide a low-dimensional space for a perturbation. In general, the input space of a deep learning classifier is very high-dimensional. Therefore, the optimization in the high-dimensional domain requires a very large number of queries and is inefficient. The optimization can be done efficiently by narrowing

down the search space for solutions by making changes in some regions at a time as a group. The dimensionality reduction techniques are used in many existing methods [4,29,30], and we observe that the main difference lies in the number of region partitions.

Given image size w, the SplitBlock equally divides the perturbation into n^2 ($n \in \{1, \ldots, w\}$) square regions. Then, each divided area including its coordinates is stored in the order of the region size in the set.

While Square Attack [5] updates the perturbation by randomly selecting a square shaped region $s \times s$ of size $s(< w)$ from the image size $h \times w$, our method updates it regularly for each of the n^2 equally divided square regions. After updating all the n^2 regions, we move on to search in $(n+1)^2$ regions. This can be observed in Fig. 2. In the testing phase, we show that the non-orthogonal search direction and n^2 partitions provide a wider change area in the low query budget, which is a factor to achieve high query efficiency.

4 Experiments

In this section, we evaluate our method by comparing it with other ℓ_∞-attack methods: Parsimonious [30], SignHunter [4] and Square Attack [5]. We consider the ℓ_∞-threat model and execute attacks on both untargeted and targeted attack settings, then quantify the performance in terms of attack success rates, average queries and median queries. The attack success rate is calculated by the proportion of adversarial images which successfully fool the model. The mean and median queries are the mean and median number of queries for successful adversarial images.

In Sect. 4.1, we show results based on naturally trained models, i.e., models that are not hardened against adversarial attacks. In Sect. 4.2 and 4.3, we show results based on robust models of adversarially training and models with input-transformation-based defenses. In Sect. 4.4, we evaluate our method a little more by ablation study. Specifically, we experimentally investigate how much each of our defined RPNs contributes to the attack performance.

4.1 Experiments on Naturally Trained Models

Datasets and Target Models. We evaluate our method on CIFAR-10 [22] and ImageNet [11] datasets. CIFAR-10 is $32 \times 32 \times 3$ dimensional images having 10 classes. For CIFAR-10, we randomly choose $1,000$ images from the test set for evaluation, all of which are initially correctly recognized by the target model. ImageNet has $1,000$ classes. Since the size of images of ImageNet dataset is not fixed, we re-scale these images to $299 \times 299 \times 3$ (default input size of Inception-v3 model explained below). For ImageNet, we randomly choose $1,000$ images belonging to $1,000$ categories from ILSVRC 2012 validation set, all of which are initially correctly recognized by the target model. All images are normalized in $[0, 1]$ scale, and for all experiments, we clip the perturbed image into the input domain $[0, 1]^d$ for all algorithms by default.

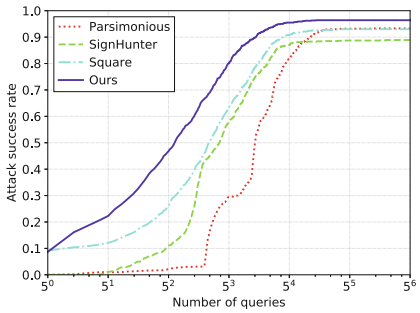

Fig. 3. Cumulative distribution of the number of queries required for untargeted attacks on CIFAR-10.

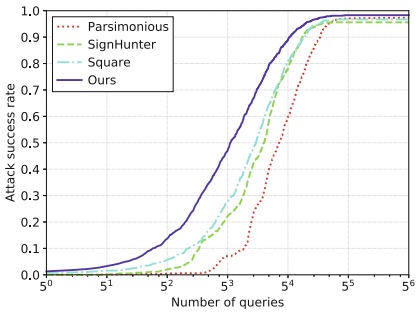

Fig. 4. Cumulative distribution of the number of queries required for targeted attacks on CIFAR-10.

Table 2. Results of both untargeted and targeted attacks on Inception-v3 classifying ImageNet. We set the norm bound $\epsilon_\infty = 0.031$ and a limit of queries to 10 k.

Attack	Success rate		Avg. queries		Med. queries	
	Untargeted	Targeted	Untargeted	Targeted	Untargeted	Targeted
Parsimonious [30]	96.1%	78.4%	1082	3495	389	2807
SignHunter [4]	94.9%	72.4%	966	3656	204	3222
Square Attack [5]	98.5%	**90.9%**	568	2592	96	1716
Ours	**98.6%**	90.2%	**416**	**2116**	**49**	**1312**

For the experiments on CIFAR-10, we use Madry et al.'s naturally trained model [28]. The model architecture and weights are available at here[1]. For the experiments on ImageNet, we use the pre-trained model provided as an application in Keras[2]. We select the Inception-v3 [37] pre-trained model in our experiments because we can see in [5] that it is robuster than some other models for ImageNet against adversarial attacks.

Method Setting. Since it is standard in the literature, we give a budget of 10 k queries per image to find an adversarial perturbation. We set the maximum ℓ_∞-perturbation of the adversarial image to $\epsilon = 0.031$ ($\approx 8/255$) on both CIFAR-10 and ImageNet. Query budgets and the maximum distortion ϵ are parameters specific to the threat model of adversarial attacks, so they are generally not considered as hyperparameters. In targeted attacks, we set the target class to $y_{target} = (y_{true} + 1)\ mod\ K$, where y_{true} is the true class, and K is the number of classes.

[1] https://github.com/MadryLab/cifar10_challenge.
[2] https://keras.io/api/applications/.

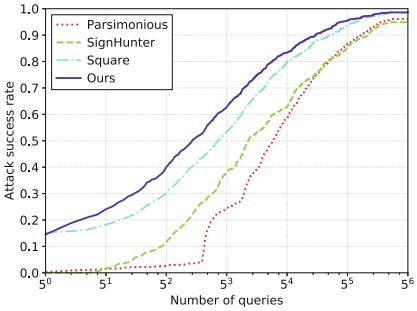

Fig. 5. Cumulative distribution of the number of queries required for untargeted attacks on ImageNet.

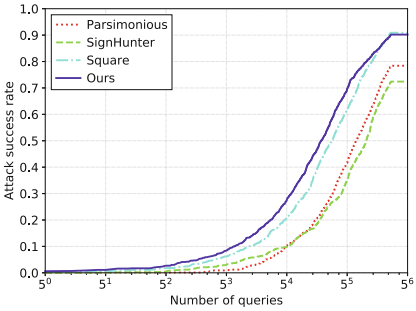

Fig. 6. Cumulative distribution of the number of queries required for targeted attacks on ImageNet.

Results on CIFAR-10. We show the results in Table 1. Our method achieves the highest attack success rates on both untargeted and targeted settings. Also at the same time, we improve the number of queries required to fool the classifiers compared to other three methods. Compared to the state-of-the-art method, Square Attack, our method achieves a higher attack success rate, 1.5 to 1.8 times higher average query efficiency, and 1.9 to 2.4 times higher median query efficiency. We also plotted the cumulative success rates in terms of the required budget in Figs. 3 and 4. Especially in low-query budgets, our method remarkably outperforms the other methods. Additionally, the success rates of Square Attack and our method at 1 query indicate the strength of the vertical-wise initialization. As hyperparameters for the comparison methods, we set $block\ size = 4$ and $batch\ size = 64$ for Parsimonious and $p = 0.05$ for Square Attack by default.

Results on ImageNet. The results are presented in Table 2, and Figs. 5 and 6. Although our method does not achieve the highest attack success rate in the targeted attack setting, it achieves higher attack success rate and query efficiency in the untargeted attack setting. As Figs. 5 and 6 show, our method achieves the highest attack success rate up to 5^5 queries in both untargeted and targeted attack settings. Additionally, we can see from Table 2 that more than half of the images are successfully attacked for the untargeted attack with 49 queries, which is about half of the median query of Square Attack. These results indicate the high query efficiency in low query budgets of our method. As hyperparameters for the comparison methods, we set $block\ size = 32$ and $batch\ size = 64$ for Parsimonious and $p = 0.05$ for Square Attack.

Table 3. Results on adversarially trained models of Madry et al. [28], TRADES [48], CLP and LSQ [21] on MNIST, and CLP and LSQ [21] on CIFAR-10. We set the norm bound ϵ_∞ and a limit of queries to 0.3 and 10 k respectively for MNIST and 0.062 ($\approx 16/255$) and 10 k respectively for CIFAR-10. The percentages in the model column indicate the natural accuracy in the test data for each model.

Dataset	Model	Attack	Success rate	Avg. queries	Med. queries
MNIST	Madry et al. [28] (99.0%)	Parsimonious	11.0%	310	58
		SignHunter	7.5%	**217**	**28**
		Square Attack	11.1%	496	204
		Ours	**11.3%**	504	73
	TRADES [48] (100.0%)	Parsimonious	7.4%	338	60
		SignHunter	5.5%	**198**	**53**
		Square Attack	**7.5%**	450	228
		Ours	**7.5%**	296	81
	CLP [21] (99.3%)	Parsimonious	87.3%	581	65
		SignHunter	24.2%	741	**6**
		Square Attack	**92.8%**	**353**	63
		Ours	80.1%	638	122
	LSQ [21] (99.1%)	Parsimonious	83.8%	418	79
		SignHunter	23.0%	852	**7**
		Square Attack	**90.1%**	**248**	68
		Ours	74.3%	666	117
CIFAR-10	CLP [21] (74.2%)	Parsimonious	99.5%	285	117
		SignHunter	**99.9%**	**109**	39
		Square	99.5%	186	41
		Ours	99.7%	178	**33**
	LSQ [21] (85.5%)	Parsimonious	77.5%	960	199
		SignHunter	83.4%	**354**	34
		Square	**85.0%**	533	29
		Ours	80.5%	627	**26**

4.2 Experiments on Adversarially Trained Models

Here we evaluate our method to robust models based on adversarial training.

Datasets and Target Models. We use some robust models classifying MNIST [44] and CIFAR-10 datasets as the same as the experiment of Square Attack [5]. MNIST is $28 \times 28 \times 1$ dimensional grayscale handwritten numeric dataset. In the experiments on MNIST, we randomly sample $1,000$ images from the test set, all of which are initially correctly recognized by Madry et al.'s naturally trained model [28]. In the experiments on CIFAR-10, we use the same test data in Sect. 4.1.

In line with the experiments in [5], we use the ℓ_∞-adversarially trained models of Madry et al. [28], TRADES [48], Clean Logit Pairing (CLP) [21] and Logit Squeezing (LSQ) [21] for MNIST, and the ℓ_∞-adversarially trained models of CLP and LSQ for CIFAR-10.

Method Setting. We give a budget of 10 k queries per image to find an adversarial perturbation. We set the maximum ℓ_∞-perturbation of the adversarial image to $\epsilon = 0.3$ on MNIST, and $\epsilon = 0.062$ ($\approx 16/255$) on CIFAR-10. All experiments in this section are done in the untargeted setting.

Results on MNIST. Table 3 shows the results. In Madry et al.'s and TRADES models, SignHunter achieves better query efficiency, but has a lower attack success rate on average than the other methods. Comparing the methods with similar attack success rates, our method achieves higher attack success rates and better query efficiencies. Although our method does not achieve better performance than other methods in CLP and LSQ models, our method achieves better performance in Madry et al.'s and TRADES models, where the original robust accuracy is higher. This indicates the potential attack power of our method. As the hyperparameter for Parsimonious, we set $block\ size = 4$ and $batch\ size = 64$. As the hyperparameter for Square Attack, we set $p = 0.8$ for Madry et al.'s and TRADES models and $p = 0.3$ for CLP and LSQ models.

Results on CIFAR-10. The results are shown at the bottom of Table 3. All methods have high attack success rates overall, and there is not as large a difference in attack performance due to the shape of the uniform noise as for MNIST. In both models, our method achieves the highest median query efficiency, although not the highest average query. This suggests that the query efficiency in low query budgets of our method is high. As hyperparameters for the comparison methods, we set $block\ size = 4$ and $batch\ size = 64$ for Parsimonious and $p = 0.3$ for Square Attack.

On the Difference in Attack Success Rates in CLP and LSQ Models. It may be concluded that the difference in the attack performance of the methods in CLP and LSQ models classifying MNIST comes from the form of local uniform noise generated by each method. SignHunter considers the image as a one-dimensional vector and flips the sign in a particular segmentation range, so that a rectangular noise can be seen in the image. On the other hand, Parsimonious and Square Attack make most of the noise consist of square shaped uniform noise. The results in Table 3 show a large margin in terms of attack success rate of the attacks between these two patterns. This suggests that CLP and LSQ models are particularly vulnerable to square shaped uniform noise, which causes the large differences.

Table 4. Results on input-transformation-based defenses: Bit-Red [43], JPEG [17], FD [26], and ComDefend [20]. Each defense method adopts the backbones of Madry et al.'s naturally trained model classifying CIFAR-10 and Inception-v3 pre-trained model classifying ImageNet, respectively. We use 50 randomly selected images and set a limit of queries to 200, the norm bound ϵ_∞ to 0.031 for CIFAR-10 and 0.062 for ImageNet.

Dataset	Defense	Attack	Success rate	Avg. queries	Med. queries
CIFAR-10	Bit-Red [43] (78.0%)	Parsimonious	71.8%	61	71
		SignHunter	84.6%	32	16
		Square Attack	**92.3%**	27	14
		Ours	**92.3%**	**23**	**12**
	JPEG [17] (82.0%)	Parsimonious	48.8%	95	75
		SignHunter	73.2%	63	52
		Square Attack	85.4%	31	19
		Ours	**92.7%**	**29**	**12**
	FD [26] (86.0%)	Parsimonious	81.4%	73	70
		SignHunter	83.7%	38	28
		Square Attack	97.7%	28	8
		Ours	**100.0%**	**24**	**6**
ImageNet	Bit-Red [43] (78.0%)	Parsimonious	51.3%	86	74
		SignHunter	74.4%	60	35
		Square Attack	**84.6%**	35	22
		Ours	82.1%	**33**	**12**
	JPEG [17] (82.0%)	Parsimonious	41.7%	82	69
		SignHunter	66.7%	89	73
		Square Attack	77.1%	37	13
		Ours	**83.3%**	**36**	**8**
	FD [26] (86.0%)	Parsimonious	66.0%	55	67
		SignHunter	74.5%	35	18
		Square Attack	93.6%	27	7
		Ours	**97.9%**	**21**	**4**
	ComDefend [20] (94.0%)	Parsimonious	27.7%	80	78
		SignHunter	61.7%	72	61
		Square Attack	**74.5%**	45	25
		Ours	**74.5%**	**37**	**13**

4.3 Experiments on Input-Transformation-Based Defenses

In this section, we attack against input-transformation-based defense methods other than adversarial training.

Datasets and Target Models. Since the basic input-transformation-based defense methods are input-independent, they can be applied to various models to easily improve the defense performance against adversarial attacks. We consider

four defense methods: Bit-Depth Reduction (Bit-Red) [43], JPEG Compression (JPEG) [17], Feature Distillation (FD) [26], and ComDefend [20]. All of these defense methods are input-transformation-based methods that apply a transformation to the input image to mitigate the effects of adversarial perturbation. We conduct attack experiments on models applying each defense method to Madry et al.'s naturally trained model for classifying CIFAR-10 and Inception-v3 pre-trained model for classifying ImageNet, respectively. Since ComDefend requires a separate pre-trained model for defense and is not available in CIFAR-10, we only consider ImageNet for this method. For the test data on both CIFAR-10 and ImageNet, we randomly sample 50 images from those used in Sect. 4.1, and we generate adversarial examples of these images.

Method Setting. Considering a more realistic setting, we give a budget of 200 queries, which is much less than the number of queries in the experiment in Sect. 4.1. We set the maximum ℓ_∞-perturbation of the adversarial image to $\epsilon = 0.031$ ($\approx 8/255$) on CIFAR-10, and $\epsilon = 0.062$ ($\approx 16/255$) on ImageNet. The amount of perturbation distortion on ImageNet is based on VMI-CT-FGSM [40]. All experiments in this section are done in the untargeted setting.

Results on CIFAR-10. The results are shown in upper part of Table 4. Our method outperforms the other black-box attacks against all three input-transformation-based defenses. Our method achieves an attack success rate of more than 90% for all defense methods, and when compared to the results for the case without defense methods in Sect. 4.1, it can be seen that our method is almost unaffected by these defenses. Overall, our method achieves better performance in situations where the attacker is given only a small query budget. As hyperparameters for Parsimonious, we set $block\ size = 4$ and $batch\ size = 64$.

Results on ImageNet. The results are shown in the lower part of Table 4. Our method achieves better attack performance except the attack success rate on Bit-Red. In particular, the median query of our method is about half that of Square Attack in most settings, which indicates a relatively high query efficiency of our method. The defense methods such as input transformation are very easy to apply to ImageNet with high dimensionality and are considered more realistic than adversarial training. However, such a simple defense method is not sufficient to prevent adversarial attacks. In terms of the amount of perturbation distortion in the adversarial image, these defense methods may be more robust for smaller amounts of that. As hyperparameters for Parsimonious, we set $block\ size = 32$ and $batch\ size = 64$.

4.4 Ablation Study

In this subsection, we evaluate our methodology a little more. We perform a simple ablation study to show how the individual RPNs (in Sect. 3.2) improve

Table 5. Ablation study of our method which shows how the individual RPNs (in Sect. 3.2) improve the performance. Our final method is highlighted in blue, and the results are shown below when each RPN was removed from the "All" sampling space.

Sampling space	Success rate	Avg. queries	Med. queries
All	**90.2%**	**2116**	**1312**
All − vertical-wise	88.7%	2208	1366
All − horizontal-wise	89.8%	2263	1390
All − uniform	88.8%	2237	1314
All − diagonal-wise	88.2%	2271	1468

the performance of our attack. The comparison is done for an ℓ_∞-threat model of radius $\epsilon = 0.031$. We use $1,000$ test images and carry out targeted attacks against the Inception-v3 model pre-trained on ImageNet with a 10 k query budget. Results are shown in Table 5. The "All" in sampling space column means that the RPN is sampled from the all sampling space (vertical-wise, horizontal-wise, uniform and diagonal-wise), which is our final method we used in our experiments. The results when each RPN is removed from the all sampling space are shown below that. In terms of attack success rate and query efficiency, we can see that all RPNs contribute to the attack performance. In particular, when the diagonal-wise pattern is removed, the attack success rate and query efficiency are greatly degraded. In addition, based on the results, further analysis of the noise patterns will be a future challenge, assuming that the addition of new RPNs will improve the attack performance.

5 Conclusion

We proposed a query-efficient black-box attack using an iterative random search and random pattern noises. In our experiments, we show that our method achieves higher success rates than existing methods in both untargeted and targeted attacks, especially in low-query budgets. In the experiments on defensive models, we show that our method achieves high attack performance in most settings. Since our method is hyperparameter-free, it is practical and easy to apply for attackers.

References

1. Amazon Rekognition. https://aws.amazon.com/rekognition/
2. Google Cloud Vision API. https://cloud.google.com/vision/
3. IBM Watson Visual Recognition. https://www.ibm.com/cloud/watson-visual-recognition
4. Al-Dujaili, A., O'Reilly, U.M.: Sign bits are all you need for black-box attacks. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=SygW0TEFwH

5. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 484–501. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_29
6. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch. CoRR abs/1712.09665 (2017)
7. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy 2017, pp. 39–57 (2017)
8. Chen, J., Jordan, M.I., Wainwright, M.J.: HopSkipJumpAttack: a query-efficient decision-based attack. In: IEEE Symposium on Security and Privacy 2020, pp. 1277–1294 (2020)
9. Chen, P., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.: Zoo: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the AISec@CCS 2017, pp. 15–26 (2017)
10. Chen, S., Carlini, N., Wagner, D.A.: Stateful detection of black-box adversarial attacks. In: Proceedings of the SPAI 2020, pp. 30–39 (2020)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR 2009 (2009)
12. Deng, Y., Zheng, J.X., Zhang, T., Chen, C., Lou, G., Kim, M.: An analysis of adversarial attacks and defenses on autonomous driving models. In: PerCom 2020, pp. 1–10 (2020)
13. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: Proceedings of the CVPR 2019, pp. 4312–4321 (2019)
14. Feng, Y., Wu, B., Fan, Y., Li, Z., Xia, S.: Efficient black-box adversarial attack guided by the distribution of adversarial perturbations. CoRR abs/2006.08538 (2020)
15. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR 2015 (2015)
16. Gu, S., Rigazio, L.: Towards deep neural network architectures robust to adversarial examples. In: ICLR 2015 (2015)
17. Guo, C., Rana, M., Cissé, M., van der Maaten, L.: Countering adversarial images using input transformations. In: ICLR 2018 (2018)
18. Guo, Y., Yan, Z., Zhang, C.: Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) NeurIPS 2019, pp. 3820–3829 (2019)
19. Huang, Z., Zhang, T.: Black-box adversarial attack with transferable model-based embedding. In: ICLR 2020 (2020)
20. Jia, X., Wei, X., Cao, X., Foroosh, H.: ComDefend: an efficient image compression model to defend adversarial examples. In: Proceedings of the CVPR 2019, pp. 6084–6092 (2019)
21. Kannan, H., Kurakin, A., Goodfellow, I.J.: Adversarial logit pairing (2018)
22. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
23. Li, H., Xu, X., Zhang, X., Yang, S., Li, B.: QEBA: query-efficient boundary-based blackbox attack. In: Proceedings of the CVPR 2020, pp. 1218–1227 (2020)

24. Li, Y., Bai, S., Xie, C., Liao, Z., Shen, X., Yuille, A.: Regional homogeneity: towards learning transferable universal adversarial perturbations against defenses. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12356, pp. 795–813. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58621-8_46
25. Lin, J., Song, C., He, K., Wang, L., Hopcroft, J.E.: Nesterov accelerated gradient and scale invariance for adversarial attacks. In: ICLR 2020 (2020)
26. Liu, Z., et al.: Feature distillation: DNN-oriented JPEG compression against adversarial examples. In: CVPR 2019, pp. 860–868 (2019)
27. Ma, C., Cheng, S., Chen, L., Yong, J.: Switching gradient directions for query-efficient black-box adversarial attacks. CoRR abs/2009.07191 (2020)
28. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR 2018 (2018)
29. Meunier, L., Atif, J., Teytaud, O.: Yet another but more efficient black-box adversarial attack: tiling and evolution strategies (2019)
30. Moon, S., An, G., Song, H.O.: Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In: ICML 2019, pp. 4636–4645 (2019)
31. Papernot, N., McDaniel, P.D., Goodfellow, I.J., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the AsiaCCS 2017, pp. 506–519 (2017)
32. Papernot, N., McDaniel, P.D., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: IEEE EuroS&P 2016, pp. 372–387 (2016)
33. Rastrigin, L.A.: The convergence of the random search method in the extremal control of many-parameter system. Autom. Remote Control **24**(10), 1337–1342 (1963). https://scholar.google.com/scholar?cluster=1484480983410715230
34. Shaham, U., et al.: Defending against adversarial images using basis functions transformations. CoRR abs/1803.10840 (2018)
35. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the ACM CCS 2016, pp. 1528–1540 (2016)
36. Su, D., Zhang, H., Chen, H., Yi, J., Chen, P., Gao, Y.: Is robustness the cost of accuracy? – A comprehensive study on the robustness of 18 deep image classification models. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018, pp. 644–661 (2018)
37. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the CVPR 2016, pp. 2818–2826 (2016)
38. Szegedy, C., et al.: Intriguing properties of neural networks. In: ICLR 2014 (2014)
39. Tu, C., et al.: Autozoom: autoencoder-based zeroth order optimization method for attacking black-box neural networks. In: Proceedings of the AAAI 2019, pp. 742–749 (2019)
40. Wang, X., He, K.: Enhancing the transferability of adversarial attacks through variance tuning. In: Proceedings of the CVPR 2021, pp. 1924–1933 (2021)
41. Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.L.: Mitigating adversarial effects through randomization. In: ICLR 2018 (2018)
42. Xie, C., et al.: Improving transferability of adversarial examples with input diversity. In: Proceedings of the CVPR 2019, pp. 2730–2739 (2019)
43. Xu, W., Evans, D., Qi, Y.: Feature squeezing: detecting adversarial examples in deep neural networks. In: NDSS 2018 (2018)

44. Yann, L., Corinna, C.: The MNIST database of handwritten digit (1998)
45. Yatsura, M., Metzen, J.H., Hein, M.: Meta-learning the search distribution of black-box random search based adversarial attacks. CoRR abs/2111.01714 (2021)
46. Yin, D., Lopes, R.G., Shlens, J., Cubuk, E.D., Gilmer, J.: A Fourier perspective on model robustness in computer vision. In: NeurIPS 2019, pp. 13255–13265 (2019)
47. Zhang, F., Chowdhury, S.P., Christakis, M.: DeepSearch: a simple and effective blackbox attack for deep neural networks. In: Devanbu, P., Cohen, M.B., Zimmermann, T. (eds.) ESEC/FSE 2020, pp. 800–812 (2020)
48. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: ICML 2019 (2019)