Decision-Based Black-Box Attack Specific to Large-Size Images

Dan Wang and Yuan-Gen Wang(✉)

School of Computer Science and Cyber Engineering, Guangzhou University,
Guangzhou, China
wangyg@gzhu.edu.cn

Abstract. Decision-based black-box attacks can craft adversarial examples by only querying the target model for hard-label predictions. However, most existing methods are not efficient when attacking large-size images due to optimization difficulty in high-dimensional space, thus consuming lots of queries or obtaining relatively large perturbations. In this paper, we propose a novel decision-based black-box attack to generate adversarial examples, which is Specific to Large-size Image Attack (SLIA). We only perturb on the low-frequency component of discrete wavelet transform (DWT) of an image, reducing the dimension of the gradient to be estimated. Besides, when initializing the adversarial example of the untargeted attack, we remain the high-frequency components of the original image unchanged, and only update the low-frequency component with the randomly sampled uniform noise, thereby reducing the distortion at the beginning of the attack. Extensive experimental results demonstrate that the proposed SLIA outperforms state-of-the-art algorithms when attacking a variety of different threat models. The source code is publicly available at https://github.com/GZHU-DVL/SLIA.

1 Introduction

At present, deep neural networks (DNNs) have been widely applied in various fields due to their ability to efficiently solve complex tasks. However, DNN is highly uninterpretable, making it difficult to control [1]. The safety of its applications in specific fields deserves attention, such as military, autonomous driving, and medical treatment. The concept of adversarial example was first proposed by Szegedy et al. [1] in 2014. That is, adding a small perturbation to an original image can generate an adversarial example that makes the DNN model misclassified with high confidence. According to the accessible knowledge of the structure and parameters of the target model, adversarial attack can be divided into white-box attack and black-box attack. Since the black-box attack is more practicable, thus attracting more attentions than the former [2]. The black-box attack includes the transfer-based [3] and query-based attack [4].

In the transfer-based black-box attack, Dong et al. [5] make the generated adversarial examples more transferable by increasing the momentum in the gradient direction. However, this approach has low attack success rate. In [11], Papernot et al. propose a dataset expansion method based on the Jacobian matrix to

L. Wang et al. (Eds.): ACCV 2022, LNCS 13842, pp. 357–372, 2023.
https://doi.org/10.1007/978-3-031-26284-5_22

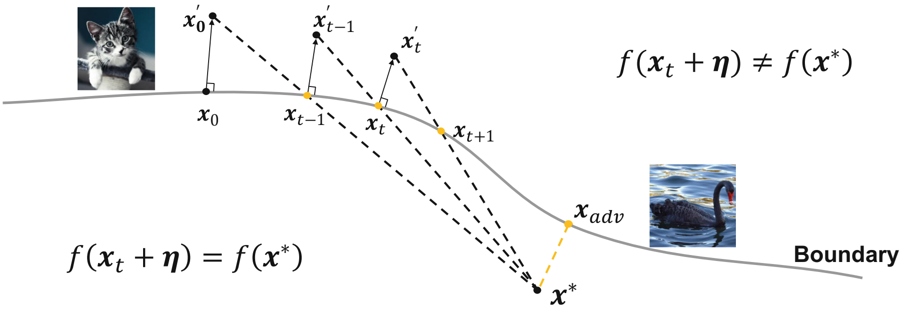

Fig. 1. Pipeline of SLIA. The attack is initialized with an example that has already been adversarial, and then generating adversarial examples iteratively move along the decision boundary. Taking targeted attack as an example, we aim to obtain an adversarial image that visually looks like a cat but be misclassified as a black swan.

iteratively expand and improve surrogate model. However, when the dimension of the sampled image is large, the calculation of the Jacobian matrix will consume huge resources. Besides, it is difficult to completely imitate the decision boundary of the attacked model, which causes a low attack success rate.

Since the surrogate model cannot fully imitate the target model, many researchers tend to directly estimate the structure and parameter information of the target model. The focus of black-box attacks is gradient estimation by querying model. Chen et al. [4] utilize the finite difference based Zero-Order Optimization (ZOO) algorithm to estimate the gradient of the loss function by accessing predicted probabilities of the target model. This method needs to estimate each pixel one by one, which requires numerous queries to generate accurate gradient estimation in each iteration, causing the low attack efficiency. Bhagoji et al. [13] use the finite difference method and the random grouping method to reduce the amount of calculation. However, the reduced calculation causes the low attack success rate on the large-size image dataset.

When the model's prediction probabilities are accessible, attackers will typically prefer score-based attack. While in more realistic scenarios where only top-1 class predictions are available, attackers will have to resort to decision-based attack. The concept of boundary-based black-box attack was first proposed by Brendel et al. [19]. It only needs to utilize the final classification output of the model to craft adversarial example. The method works by randomly walking in the direction of the original example along the decision boundary until it is closest to the original example, while remaining adversarial. This attack requires less model knowledge but can achieve comparable attack effects to white-box attack. However, the perturbation sampling strategy in [19] has great randomness, and the convergence of perturbation cannot be guaranteed. To address this problem, [6,20] were proposed to carry out decision-based black-box attack. However, these attacks often require numerous queries to converge or have large perturbations under a given number of query budget, which makes the attack process consume heavy computation, especially when attacking large-size images.

To improve the query efficiency, we propose a decision-based boundary adversarial attack, which is specific to large-size images, termed SLIA. SLIA optimizes both l_2-norm and l_∞-norm distortion. The main contributions of this paper are as follows: (1) We propose a decision-based black-box attack for large-size images (named SLIA), wherein adversarial images can be crafted by sending a few queries to the model; (2) When performing untargeted attack, SLIA replaces the low-frequency component of the original image with random uniform noise, and reconstructs it back to the original image space with high-frequency components. This can fool the model while retaining as much key information of the original image as possible; (3) SLIA performs discrete wavelet decomposition on adversarial example at the boundary, only estimates and updates the gradient of low-frequency component, greatly reduces the number of dimensions to be estimated with fewer model queries. Experiments show that our algorithm can be successfully used to attack different ImageNet models with less distortion than state-of-the-art algorithms under the same number of queries.

2 Related Work

According to the available knowledge of the network model, adversarial attack is classified into white-box attack and black-box attack. In a white-box setting, the attacker has all knowledge about the network. Since Szegedy et al. [1] discovered vulnerability of DNNs, various white-box attacks [8–10,12] have been developed. In practice, the attacker may not be able to access the structure and parameters of the model, which is more in line with the actual attack situation. Hence black-box attacks have received more attention recently. It is often divided into three families: transfer-based, score-based, and decision-based attacks.

2.1 Transfer-Based Black-Box Attacks

Transfer-based black-box attack algorithms are mainly based on the phenomenon of transferability: adversarial example against a certain model is often misclassified by other models. Papernot et al. [10,11] trained a local substitute model by querying the target model and used backpropagation gradient from the substitute network to craft adversarial examples. These examples can also successfully fool the target model with high probability. The follow-up work [3] showed that adversarial example generated on substitute network tends not to have better transferability for targeted attack, but can be developed on an ensemble of models. However, query-based algorithms that directly estimate the gradient of the target network outperform these methods. In addition, it is difficult to find a suitable surrogate model to learn the decision boundary of the target model.

2.2 Score-Based Black-Box Attacks

In the score-based black-box setting, the attacker utilizes the corresponding predicted probabilities to make adversarial examples by querying the target model.

Chen et al. [21] applied zeroth order optimization and coordinate descent to estimate the gradient, but requred a large number of queries on the target model. The method in [6] performs gradient estimation via Natural Evolutionary Strategy (NES) and then uses Projected Gradient Descent (PGD) [7], further reduces the query complexity.

2.3 Decision-Based Black-Box Attacks

As an important category of adversarial attacks, an initial attempt named Boundary Attack [19] is highly relevant to real-world applications. It starts from an adversarial point and tries to reduce the distortion by walking towards the original image along the decision boundary while keeping adversarial. The main issue is the trade-off between the number of queries and the quality of adversarial example. HopSkipJumpAttack [21] significantly improves the former [19] in terms of query efficiency. This method can balance both the accuracy of gradient estimation and query complexity well. However, when attacking large-size images, the number of queries required to produce adversarial examples still is in the tens of thousands.

3 Problem Definition

We consider an image classifier $f : \boldsymbol{x} \rightarrow c$, where $\boldsymbol{x} \in \mathbb{R}^n$ is a normalized RGB image and c is its corresponding true label such as the top-1 classification label. $F(\boldsymbol{x})$ is a k-dimensional vector, referring to the probability distribution over classes. $c := \arg\max_{c\in[k]} F_c(\boldsymbol{x})$ represents the label of $\boldsymbol{x}$. Given an original image $\boldsymbol{x}^*$, c^* represents its label. Denote the adversarial perturbation as $\boldsymbol{\mu} \in \mathbb{R}^n$, the goal of untargeted attack is to make the model misclassified wherein $c(\boldsymbol{x}^* + \boldsymbol{\mu}) \neq c^*$, and targeted attack aims to change the original classifier decision c^* into a pre-specified class c^+.

The process of generating adversarial examples can be formulated as an optimization problem by defining the function $\mathcal{L}$:

$$\mathcal{L}_{\boldsymbol{x}^*}(\boldsymbol{x}) := \begin{cases} \max\limits_{c\neq c^*} F_c(\boldsymbol{x}) - F_{c^*}(\boldsymbol{x}) & \text{(Untargeted)} \\ F_{c^+}(\boldsymbol{x}) - \max\limits_{c\neq c^+} F_c(\boldsymbol{x}) & \text{(Targeted)} \end{cases} \tag{1}$$

Gradient-based methods can be used to efficiently optimize this problem under the white-box setting. However, in the decision-based black-box attack, models only provide attackers with a hard label, even without any output probabilities. In other words, only the value of sign($\mathcal{L}$) is available, while the value of $\mathcal{L}$ is unknown. We denote the indicator function $\mathcal{I}$ as:

$$\mathcal{I}_{\boldsymbol{x}^*}(\boldsymbol{x}) = \operatorname{sign}\left(\mathcal{L}_{\boldsymbol{x}^*}(\boldsymbol{x})\right) = \begin{cases} 1 & \text{if} \quad \mathcal{L}_{\boldsymbol{x}^*}(\boldsymbol{x}) > 0 \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

In our decision-based attack, the goal of the adversary is to find an adversarial perturbation $\boldsymbol{\mu}$ which satisfies $\mathcal{I}(\boldsymbol{x}^* + \boldsymbol{\mu}) = 1$ by sending queries to model. That

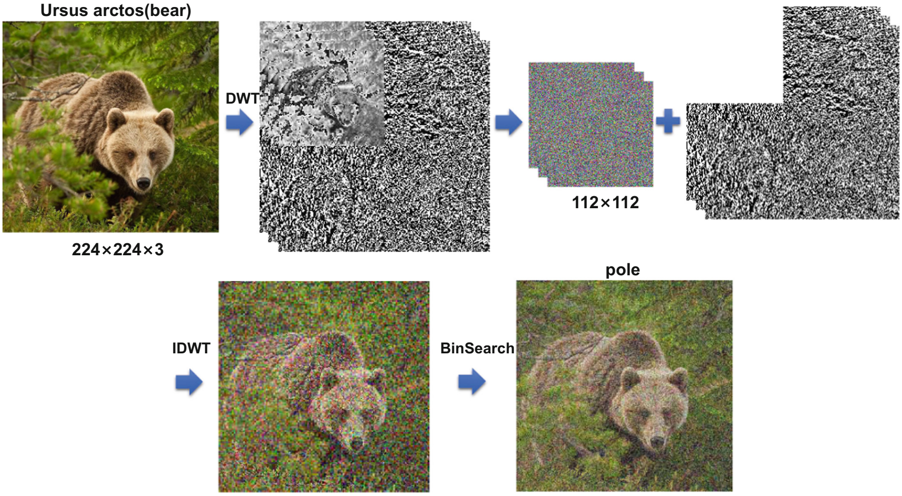

Fig. 2. Initialization for untargeted attacks.

is, only when $\mathcal{I}(\boldsymbol{x}^* + \boldsymbol{\mu}) = 1$ can it be considered as a successful attack. Generating adversarial examples under decision-based black-box setting can be defined as the following optimization problem:

$$\min \mathcal{D}(\boldsymbol{x}^*, \boldsymbol{x}^* + \boldsymbol{\mu}) \quad s.t. \quad \mathcal{I}(\boldsymbol{x}^* + \boldsymbol{\mu}) = 1 \tag{3}$$

where $\mathcal{D}(\cdot, \cdot)$ is l_2-norm or l_∞-norm distance metric. We strive to find an example with as little distortion as possible from the original example under the condition of guaranteed adversarial.

4 Decision-Based Black-Box Attack Specific to Large-Size Images (SLIA)

In this section, we propose to utilize discrete wavelet transform (DWT) to decompose the low-frequency component of the attacked image, and only adds perturbation to this part, while maintaining a 100% attack success rate. The pipeline of SLIA is shown in Fig. 1, which includes three steps: gradient estimation by querying the model, moving along the estimated gradient direction, and projecting new example to the decision boundary by binary search towards the original example. Details of each step are given below.

4.1 Initialization

Our SLIA starts from an adversarial image outside the boundary, and gradually reduces the distortion by moving towards the original image along the decision boundary while remaining adversarial.

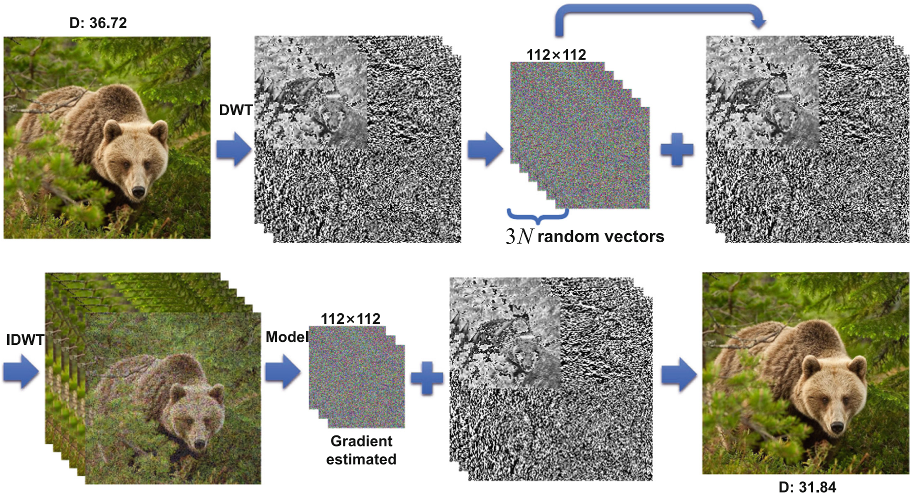

Fig. 3. Overview of estimating gradient at decision boundary.

Given a correctly classified original image $\boldsymbol{x}^*$, the first step is to generate an initial adversarial example: (1) As shown in Fig. 2, for untargeted attacks, we perform 1-level discrete wavelet decomposition on the original image. Then the low-frequency component $\boldsymbol{LL}^*$ is reset to a random uniform noise $\boldsymbol{u} \sim \mathcal{U}(\min(\boldsymbol{LL}^*), \max(\boldsymbol{LL}^*))$. Next, we combine the low-frequency noise with the original high-frequency components to reconstruct the image through inverse DWT. We make queries to the target model, until the new image is misclassified. Different from the previous attack methods that use a uniform random noise as the initialization image, the advantage of SLIA is that the new image can retain more original image information without causing large distortion. Finally, we project it to the boundary through the binary search algorithm and identify it as the initial adversarial example $\boldsymbol{x}_0$; (2) For targeted attacks, the image is randomly selected from a pre-specified class which is different from the class of the original image. Similarly, we leverage the binary search algorithm to search for the decision boundary, and take the image as initial adversarial example $\boldsymbol{x}_0$.

4.2 Gradient Direction Estimate at the Decision Boundary

In this subsection, we will elaborate the gradient estimation part in the proposed method in detail. Suppose that at the t-th iteration, the adversarial example on the boundary is $\boldsymbol{x}_t$. As shown in Fig. 3, $\boldsymbol{x}_t$ is decomposed into low-frequency and high-frequency components by DWT. Therefore, the gradient direction of loss function $\mathcal{L}$ at this point is estimated by sending queries to the target model,

$$\nabla\mathcal{L}(\boldsymbol{x}_t) := \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}_{\boldsymbol{x}^*}[\mathrm{IDWT}(\boldsymbol{LL}_t + \delta\boldsymbol{\eta}_i, \boldsymbol{HL}_t, \boldsymbol{LH}_t, \boldsymbol{HH}_t)]\boldsymbol{\eta}_i, \tag{4}$$

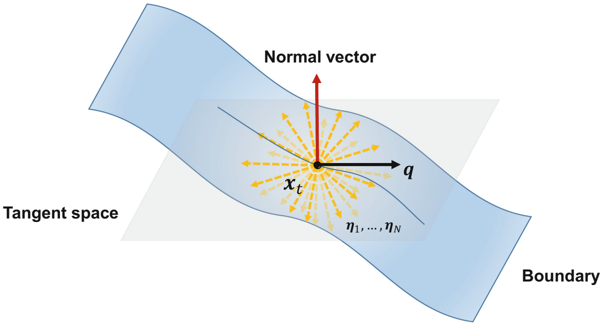

Fig. 4. Illustration of estimating gradient at $\boldsymbol{x}_t$ by sampling N Gaussian noises. $\boldsymbol{q}$ is an arbitrary in the tangent space.

where δ and t are probe step size which are a small positive parameter and t is the current number of iteration. $\boldsymbol{\eta}_{i=1}^{N}$ are normalized random noise vectors drawn from the Gaussian distribution over the 1/4-dimensional sphere as shown in Fig. 4 ($\boldsymbol{x}^* \in \mathbb{R}^n$). $\delta\boldsymbol{\eta}_{i=1}^{N}$ is added to the low-frequency component. By combining with high-frequency components of $\boldsymbol{x}_t$, inverse DWT is utilized to reconstruct N samples with unknown labels.

We determine the directions of the noise vectors by accessing the model to observe whether these samples have the same labels as the original example: (1) If $\mathcal{I}_{\boldsymbol{x}^*} = -1$, the noise vector will be updated to its opposite direction; (2) If $\mathcal{I}_{\boldsymbol{x}^*} = 1$, the noise vector will remain unchanged. Finally, we average the above noises and use the mean as the normal vector of tangent hyperplane, *i.e.*, the gradient direction $\nabla\mathcal{L}(\boldsymbol{x}_t)$ at the decision boundary.

Due to the flatness of the boundary, it is theoretically likely that the noise vectors are symmetrically distributed on both sides of the decision boundary. Therefore, the updated and unchanged noise vectors can be clustered around the true gradient as much as possible, the mean vector is also closer to the true gradient.

The gradient estimation in SLIA is essentially a Monte Carlo estimation method. When the dimension of the gradient to be estimated is large, using the Monte Carlo method requires more sampling points to make the estimated gradient closer to the true gradient. In an RGB color image, each pixel is represented by three channels. Moreover, as the size of the image becomes larger, the dimensionality of the image increases dramatically (e.g., the data dimension on ImageNet is over 150k), resulting in a low accuracy of estimating gradient. To reduce the dimension of the gradient to be estimated and further minimize the visual effect of adversarial perturbation, SLIA applies DWT to decompose the sample into low-frequency components and high-frequency components. Note that most of the key content-defining information in natural images exists at the low-frequency end of the spectrum, while high-frequency signals are often associated with noise. That is, adversarial examples are more likely to be generated by adding noise to low-frequency component. Therefore, we keep the high-frequency components unchanged, and only perturb the low-frequency component, which reduces the dimension to be perturbed to 1/4 of the original image. Adding perturbation to the low-frequency information has several advantages: (1) Only the low-frequency component is perturbed, the dimension of the gradient to be estimated is reduced to 1/4 of the original image, which means that the same number of sampling points can obtain higher estimation accuracy; (2) Only adding perturbation to the low-frequency component, the perturbation is distributed in multiple pixels, which is not easy to form salt and pepper noise and has less visual impact.

4.3 Move Along Estimated Gradient Direction

In this part, we will move one step along the gradient direction estimated in Eq. (4) to obtain an example located in the adversarial area,

$$5\boldsymbol{x}_t' = \boldsymbol{x}_t + \epsilon_t \cdot \frac{\nabla \mathcal{L}(\boldsymbol{x}_t)}{\|\nabla \mathcal{L}(\boldsymbol{x}_t)\|_2}, \tag{5}$$

where ϵ_t is perturbation magnitude at t-th iteration. It is computed from the distortion result of the last iteration and the geometric progression related to current iteration number t. We multiply the normalized estimated gradient by ϵ_t, and add it to $\boldsymbol{x}_t$ to obtain an adversarial example $\boldsymbol{x}_t'$, which is slightly away from the boundary, shown in Fig. 1. Note that $\boldsymbol{x}_t'$ is at the opposite side of the boundary to $\boldsymbol{x}^*$.

4.4 Project to Decision Boundary

Since the proposed gradient direction estimation works only at the boundary, we adopt binary search algorithm to quickly find the decision boundary and project $\boldsymbol{x}_t'$ to it. We use the following formula to adjust the value of the parameter γ to control the relative position of the adversarial example from the original example, until the stopping condition is satisfied. Hence, we move the adversarial image $\boldsymbol{x}_t'$ towards the direction of the original image $\boldsymbol{x}^*$ via

$$\boldsymbol{x}^{t+1} = \gamma_t \cdot \boldsymbol{x}^* + (1 - \gamma_t) \cdot \boldsymbol{x}_t', \tag{6}$$

where γ_t is a changing positive parameter between 0 and 1 so that $\boldsymbol{x}_t'$ projected back to the decision boundary. We denote the example projected back on the boundary as $\boldsymbol{x}^{t+1}$, and let it enter to the next iteration as a new boundary adversarial example. The pseudo code of the complete process in generating adversarial images is outlined in Algorithm 1.

5 Experiments

5.1 Experimental Settings

Dataset and Target Models. We experiment on ImageNet [18], a public large-scale labeled image dataset, to demonstrate the efficiency of our proposed

Algorithm 1 Boundary attack specific to large-size images (SLIA)

```
Input: Indicator function $\mathcal{I}$, the original example $\boldsymbol{x}^*$, the number of
  normalized random noises $N$, iteration number $T$, constraint $l_p$ (p=0 or p=$\infty$),
  attack objective (untargeted or targeted), stopping threshold of binary search.
Output: Adversarial example.
if objective is untargeted then
    $\boldsymbol{LL}^*$, $\boldsymbol{LH}^*$, $\boldsymbol{HL}^*$, $\boldsymbol{HH}^* \leftarrow$ DWT($\boldsymbol{x}^*$).
    Sample noise $\boldsymbol{u} \sim \mathcal{U}(\min(\boldsymbol{LL}^*), \max(\boldsymbol{LL}^*))$.
    while $\mathcal{I}$(IDWT($\boldsymbol{u}$, $\boldsymbol{LH}^*$, $\boldsymbol{HL}^*$, $\boldsymbol{HH}^*$)) $= -1$ do
        Sample noise $\boldsymbol{u} \sim \mathcal{U}(\min(\boldsymbol{LL}^*), \max(\boldsymbol{LL}^*))$.
    end
    $\boldsymbol{x}_{initail}$ = IDWT($\boldsymbol{u}$, $\boldsymbol{LH}^*$, $\boldsymbol{HL}^*$, $\boldsymbol{HH}^*$).
else
    A randomly sampled image $\boldsymbol{x}_{initail}$ belonging to the target class.
end
Search starting point $\boldsymbol{x}_0$ = BinarySearch($\boldsymbol{x}_{initail}$, $\boldsymbol{x}^*$, $\mathcal{I}$) which lies on the
  boundary.
for $t = 0$ to $T - 1$ do
    $\boldsymbol{LL}_t$, $\boldsymbol{LH}_t$, $\boldsymbol{HL}_t$, $\boldsymbol{HH}_t \leftarrow$ DWT($\boldsymbol{x}_t$).
    Sample $N$ noise vectors: $\boldsymbol{\eta}_{i=1}^N \sim \mathcal{N}(0,1)$.
    Estimate gradient direction of $\boldsymbol{LL}_t$: $\nabla\mathcal{L}(\boldsymbol{x}_t)$ with the rule defined in Eq.(??).
    if constraint is $l_\infty$ then
        $\nabla\mathcal{L}(\boldsymbol{x}_t) = \mathrm{sign}(\nabla\mathcal{L}(\boldsymbol{x}_t))$.
    end
    Initialize $\epsilon_t = \|\boldsymbol{x}_t - \boldsymbol{x}^*\|_p/\sqrt{t} \times 4$ for obtaining attack step size.
    while $\mathcal{I}(\boldsymbol{x}_t + \epsilon_t \cdot \frac{\nabla\mathcal{L}}{\|\nabla\mathcal{L}\|_p}) = -1$ do
        $\epsilon_t = \epsilon_t$ / 2.
    end
    Compute $\boldsymbol{x}'_t = \boldsymbol{x}_t + \epsilon_t \cdot \frac{\nabla\mathcal{L}}{\|\nabla\mathcal{L}\|_p}$.
    Update adversarial image $\boldsymbol{x}_{t+1}$= BinarySearch($\boldsymbol{x}'_t$, $\boldsymbol{x}^*$, $\mathcal{I}$) on the boundary.
end
Return an adversarial example $\boldsymbol{x}_{T-1}$;
```

method. For ImageNet, we randomly sample 100 correctly classified test images, evenly distributed among 10 randomly selected classes. The whole images are clipped into [0,1] by default for all experiments. We perform both untargeted attacks and targeted attacks to a random class against three prevailing models: ResNet-50 [22], VGG16 [23] and DenseNet-201 [24]. All models are pretrained on ImageNet and provided by Keras online[1].

Compared Baseline Methods. To demonstrate the effectiveness of our method, we compare SLIA with several state-of-the-art decision-based attacks including Boundary Attack method [19], HopSkipJumpAttack (HSJA [21] and

[1] https://keras.io/applications/#resnet50.
https://keras.io/applications/#vgg16.
https://keras.io/applications/#densenet201.

Table 1. Mean l_2-norm distortions for performing untargeted and targeted attacks with different query budgets.

Objective	Victim Model	Method	1 K	5 K	10 K	20 K
Untargeted	ResNet-50	Boundary Attack [19]	54.67	27.03	14.89	10.34
		HopSkipJumpAttack [21]	28.69	9.12	5.46	3.31
		LHS-BA [25]	23.84	6.39	4.92	3.20
		Ours	**14.73**	**4.85**	**3.59**	**2.98**
	VGG16	Boundary Attack [19]	60.06	24.76	18.63	14.83
		HopSkipJumpAttack [21]	26.35	12.22	9.78	7.97
		LHS-BA [25]	22.84	10.20	7.51	7.32
		Ours	**13.41**	**5.11**	**3.68**	**2.86**
	DenseNet-201	Boundary Attack [19]	78.83	33.29	15.90	10.64
		HopSkipJumpAttack [21]	35.20	7.74	4.52	2.92
		LHS-BA [25]	27.09	7.36	3.74	2.28
		Ours	**17.64**	**6.83**	**2.84**	**0.80**
Targeted	ResNet-50	Boundary Attack [19]	83.10	49.24	31.85	22.59
		HopSkipJumpAttack [21]	54.85	27.54	17.04	9.34
		LHS-BA [25]	50.29	26.81	16.70	9.25
		Ours	**49.10**	**26.21**	**16.16**	**9.06**
	VGG16	Boundary Attack [19]	97.23	58.94	39.27	28.25
		HopSkipJumpAttack [21]	67.36	40.49	27.47	18.17
		LHS-BA [25]	60.64	36.72	25.70	16.38
		Ours	**56.25**	**26.27**	**15.08**	**10.18**
	DenseNet-201	Boundary Attack [19]	92.78	54.86	26.41	17.03
		HopSkipJumpAttack [21]	67.92	30.63	15.79	8.62
		LHS-BA [25]	61.85	27.49	15.66	8.40
		Ours	**54.32**	**19.56**	**13.13**	**7.70**

Latin Hypercube Sampling based Boundary Attack (LHS-BA) [25]. We mainly focus on attack method LHS-BA, which outperforms all of other Boundary Attack [19], Limited Attack [6], and HSJA [21]. We use the implementation of the three algorithms with the suggested hyperparameters from the publicly available source code online. We fixed the number of queries at 1K, 5K, 10K and 20K and magnitude of the average distortion is what we mainly observe when performing untargeted and targeted attacks respectively.

Evaluation Metrics. Effective querying is the most important indicator to evaluate the decision-based adversarial attack, which requires the method to craft adversarial example with smaller model queries at the same distortion. SLIA's attack success rate is 100%, so we quantify the performance in terms of two dimensions: average l_p-norm distortion and specified query numbers. It can be formulated as:

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |\boldsymbol{x}_i|^p\right)^{\frac{1}{p}}, \tag{7}$$

Table 2. Mean l_∞-norm distortions for performing untargeted and targeted attacks with different query budgets.

Objective	Victim Model	Method	1 K	5 K	10 K	20 K
Untargeted	ResNet-50	Boundary Attack [19]	0.553	0.411	0.247	0.193
		HopSkipJumpAttack [21]	0.231	0.129	0.103	0.098
		LHS-BA [25]	0.164	0.082	0.070	0.047
		Ours	**0.089**	**0.039**	**0.032**	**0.023**
	VGG16	Boundary Attack [19]	0.475	0.349	0.257	0.124
		HopSkipJumpAttack [21]	0.291	0.185	0.121	0.087
		LHS-BA [25]	0.166	0.095	0.073	0.038
		Ours	**0.067**	**0.032**	**0.024**	**0.018**
	DenseNet-201	Boundary Attack [19]	0.431	0.318	0.234	0.109
		HopSkipJumpAttack [21]	0.267	0.132	0.107	0.076
		LHS-BA [25]	0.204	0.116	0.085	0.061
		Ours	**0.152**	**0.074**	**0.058**	**0.035**
Targeted	ResNet-50	Boundary Attack [19]	0.780	0.618	0.372	0.244
		HopSkipJumpAttack [21]	0.370	0.267	0.199	0.137
		LHS-BA [25]	0.310	0.229	0.163	0.125
		Ours	**0.253**	**0.146**	**0.120**	**0.091**
	VGG16	Boundary Attack [19]	0.739	0.584	0.301	0.236
		HopSkipJumpAttack [21]	0.441	0.238	0.186	0.133
		LHS-BA [25]	0.405	0.210	0.169	0.117
		Ours	**0.361**	**0.182**	**0.128**	**0.090**
	DenseNet-201	Boundary Attack [19]	0.683	0.553	0.291	0.255
		HopSkipJumpAttack [21]	0.410	0.216	0.175	0.117
		LHS-BA [25]	0.381	0.188	0.146	0.099
		Ours	**0.315**	**0.133**	**0.098**	**0.078**

where l_2-norm and l_∞-norm are are two most commonly used metrics in the adversarial attack field. l_2-norm means Euclidean distance between the original example and the adversarial one, and l_∞-norm represents perturbation's maximum changeable degree.

Hyperparameters. In our proposed attack, the number of iteration and the maximum queries are set to 76 and 20,000, respectively. At the t-th iteration, we compute probe step size in each gradient direction estimation by $\delta_t = \|\boldsymbol{x}_{t-1} - \boldsymbol{x}^*\|_2/n\times 4$ and $\epsilon_t = \|\boldsymbol{x}_{t-1} - \boldsymbol{x}^*\|_2/\sqrt{t}\times 4$ as perturbation step size in moving along estimated gradient direction, where $n = 224\times 224\times 3$ is the input dimension. Random vectors N is set to 100 first, and we gradually increase it by $N = N\times(t+1)^{\frac{1}{4}}$. Stopping threshold θ when performing binary search is set to $n^{-\frac{3}{2}}$.

5.2 Experimental Results

To evaluate SLIA's performance, we report mean l_2-norm and l_∞-norm distortion results in Tables 1 and 2 when performing untargeted and targeted attacks. The distortion descending curves of various algorithms under different query budgets are given in Fig. 5. Two qualitative example processes of attacking the ResNet-50 by different attack methods are shown in Figs. 6 and 7, respectively.

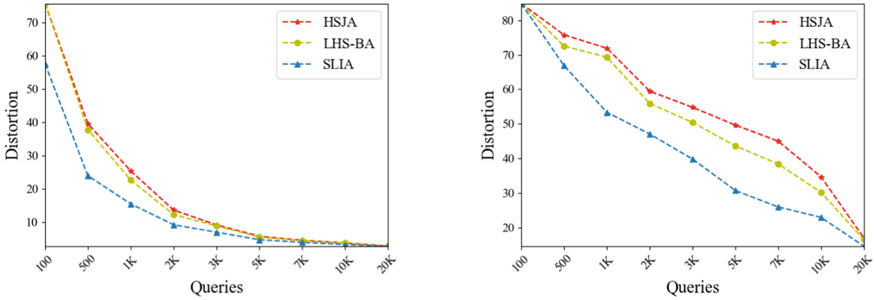

Fig. 5. l_2-norm distortions across various model queries on ImageNet with ResNet-50. 1st column: untargeted attacks. 2nd column: targeted attacks.

Untargeted Attacks. As shown in the untargeted attack section of Tables 1 and 2, it is obvious that our method outperforms existing decision-based attacks by a large margin under all fixed number of model queries. SLIA also converges in a fewer number of queries, as shown in Fig. 5.

Especially in the early stages of the attack, the advantages of SLIA are more obvious. When the number of fixed model queries does not exceed 10K: (1) Under the l_2-norm distance metric, SLIA can reduce the distortion to 56% of HSJA and about 67% of LHS-BA; (2) Under the l_∞-norm distance metric, the distortion of adversarial examples constructed via SLIA is about 64% lower than that of HSJA and about 45% lower than that of LHS-BA. Experimental data demonstrates that the adversarial examples can be crafted by our method rather quickly without using too many queries.

This is due to two reasons: (1) In the initialization part, we replace the low-frequency component of the original example with a uniform noise, and do not update other high-frequency components. In this way, more details of the original example can be preserved in the case of making the model misclassify; (2) When estimating the gradient, we consider DWT to decompose the low-frequency component of the example, and estimate the gradient of it. This greatly reduces the dimension of the gradient to be estimated to 1/4 of the original space. When sampling the same amount of Gaussian noises, the gradient can be estimated with higher accuracy than that of the original full space.

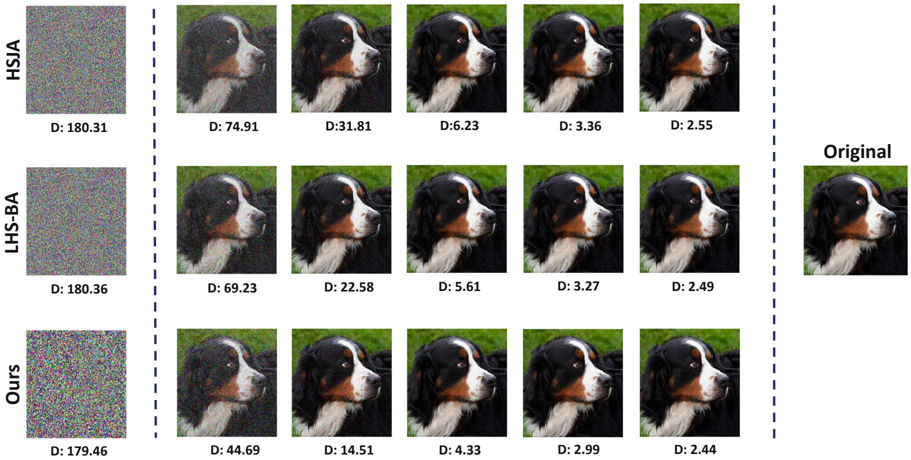

Fig. 6. Visualized trajectories of HSJA [21], LHS-BA [25] and SLIA for performing untargeted attacks on ResNet-50. 1st column: initialization. 2nd-9th columns: images after blended with original images and at 1 K, 5 K, 10 K, 20 K model queries. 10th column: original image. D is l_2-norm metric to compute the distortion between adversarial image and original image.

Targeted Attacks. We randomly select a target label and pick one image belonging to the target label. Then we use it as initialization image for all targeted attacks. The results for targeted attacks are presented in the lower parts of Tables 1 and 2. We can see that SLIA not only outperforms HSJA [21], but also surpasses the latest gradient estimation-based boundary attack LHS-BA [25]. From a qualitative example comparison using different methods shown in Fig. 7, when model queries is fixed at 5,000 (4-th column), the adversarial example crafted by SLIA is visually closer to the original example than the other two attacks. It can be seen that under a limited number of queries, SLIA is able to make adversarial examples with significantly smaller distortions from the corresponding original example. In other words, under the same distortion condition, SLIA requires fewer number of queries than the state-of-the-art methods. We can also find that SLIA requires a larger number of model queries to achieve a comparable distortion when performing targeted attacks than untargeted attacks. This phenomenon is evident on the ImageNet dataset which has many categories. There is often an order-of-magnitude difference in the average l_p-norm distortion between untargeted and targeted attacks for the same number of queries.

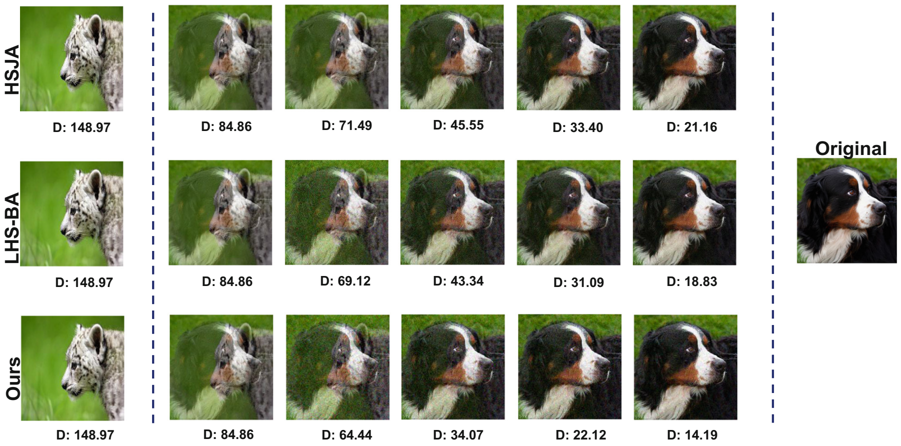

Fig. 7. Visualized trajectories of HSJA [21], LHS-BA [25] and SLIA for performing targeted attacks on ResNet-50. 1st column: initialization. 2nd-9th columns: images after blended with original images and at 1K, 5K, 10K, 20K model queries. 10th column: original image. D is l_2-norm metric to compute the distortion between adversarial image and original image.

6 Conclusion

In this work, we present a query-efficient adversarial example generation algorithm (SLIA), which is specific to ImageNet with a large image size. SLIA can be performed to ensure 100% attack success rate for settings where the attacker only has access to the final decisions of a model. We generate adversarial examples by estimating the gradient of the low-frequency component, which greatly reduces the dimension of the gradient to be estimated. When attacking a variety of different ImageNet models, the distortion can be reduced faster with our method compared to state-of-the-art attacks with different query budgets.

Acknowledgement. This work was supported in part by the NSFC (61872099, 62272116) and in part by the Scientific Research Project of Guangzhou University (YJ2021004). The authors acknowledge the Network Center of Guangzhou University for providing HPC computing resources.

References

1. Szegedy, C., et al.: Intriguing properties of neural networks. In: Proceedings of International Conference on Learning Representations (2014)
2. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Proceedings of International Conference on Learning Representations (2015)
3. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. In: Proceedings of International Conference on Learning Representations (2016)

4. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., 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)
5. Dong, Y., et al.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9185–9193. IEEE (2018)
6. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: Proceedings of International Conference on Machine Learning, pp. 2142–2151. ACM (2018)
7. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: Proceedings of International Conference on Learning Representations (2017)
8. Fan, Y., et al.: Sparse adversarial attack via perturbation factorization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12367, pp. 35–50. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58542-6_3
9. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: DeepFool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582. IEEE (2016)
10. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: Proceedings of the IEEE European Symposium on Security and Privacy (Euro S&P), pp. 372–387. IEEE (2016)
11. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519. ACM (2017)
12. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE (2017)
13. Bhagoji, A.N., He, W., Li, B., Song, D.: Practical black-box attacks on deep neural networks using efficient query mechanisms. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11216, pp. 158–174. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01258-8_10
14. Tu, C.-C., et al.: AutoZOOM: autoencoder-based zeroth order optimization method for attacking black-box neural networks. In AAAI Conference on Artificial Intelligence (2018)
15. Al-Dujaili, A., O'Reilly, U.M.: Sign bits are all you need for black-box attacks. In: Proceedings of International Conference on Learning Representations (2020)
16. Guo, C., Gardner, J.R., You, Y., Wilson, A.G., Weinberger, K.Q.: Simple black-box adversarial attacks. arXiv preprint arXiv:1905.07121 (2019)
17. Moon, S., An, G., Song, H.O.: Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In: Proceedings of International Conference on Machine Learning (2019)
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
19. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: reliable attacks against black-box machine learning models. In: Proceedings of International Conference on Learning Representations (2018)

20. Cheng, M., Le, T., Chen, P.Y., Yi, J., Zhang, H., Hsieh, C.J.: Query-efficient hard-label black-box attack: an optimization-based approach. In: Proceedings of International Conference on Learning Representations (2019)
21. Chen, J., Jordan, M.I., Wainwright, M.: HopSkipJumpAttack: a query-efficient decision-based attack. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), pp. 1277–1294. IEEE (2020)
22. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. (2014)
24. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2261–2269. IEEE (2017)
25. Wang, D., Lin, J., Wang, Y.-G.: Query-efficient adversarial attack based on Latin hypercube sampling. arXiv preprint arXiv: 2207.02391. (Accept for presentation in IEEE International Conference on Image Processing 2022)
26. Mallat, S.: The theory for multiresolution signal decomposition: the wavelet representation. In: Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 654–693. IEEE (1989)
27. Guo, C., Frank, J.S., Weinberger, K.Q.: Low frequency adversarial perturbation. In: International Conference on Uncertainty in Artificial Intelligence, pp. 1127–1137. AUAI (2019)