Projection & Probability-Driven Black-Box Attack

Jie Li[1], Rongrong Ji[1]*, Hong Liu[1], Jianzhuang Liu[2], Bineng Zhong[3], Cheng Deng[4], Qi Tian[2]
[1]Department of Artificial Intelligence, School of Informatics, Xiamen University,
[2]Noah's Ark Lab, Huawei Technologies [3]Huaqiao University [4]Xidian University
lijie.32@outlook.com, rrji@xmu.edu.cn, lynnliu.xmu@gmail.com, liu.jianzhuang@huawei.com,
bnzhong@hqu.edu.cn, chdeng.xd@gmail.com, tian.qi1@huawei.com,

Abstract

Generating adversarial examples in a black-box setting retains a significant challenge with vast practical application prospects. In particular, existing black-box attacks suffer from the need for excessive queries, as it is non-trivial to find an appropriate direction to optimize in the high-dimensional space. In this paper, we propose Projection & Probability-driven Black-box Attack (PPBA) to tackle this problem by reducing the solution space and providing better optimization. For reducing the solution space, we first model the adversarial perturbation optimization problem as a process of recovering frequency-sparse perturbations with compressed sensing, under the setting that random noise in the low-frequency space is more likely to be adversarial. We then propose a simple method to construct a low-frequency constrained sensing matrix, which works as a plug-and-play projection matrix to reduce the dimensionality. Such a sensing matrix is shown to be flexible enough to be integrated into existing methods like NES and Bandits$_{TD}$. For better optimization, we perform a random walk with a probability-driven strategy, which utilizes all queries over the whole progress to make full use of the sensing matrix for a less query budget. Extensive experiments show that our method requires at most 24% fewer queries with a higher attack success rate compared with state-of-the-art approaches. Finally, the attack method is evaluated on the real-world online service, i.e., Google Cloud Vision API, which further demonstrates our practical potentials. [1]

1. Introduction

While deep neural networks (DNNs) have proven their dominant performance on a wide range of computer vision tasks, they are shown to be vulnerable to adversarial examples [24, 38, 40]. In such a scenario, the imperceptible perturbations added to input samples can mislead the output of DNNs, which has raised serious security concerns in the literature [7, 15, 33].

Adversarial attacks can be generally categorized into the *white-box* attack and *black-box* attack. In white-box attacks, the adversary has the full knowledge of the victim model including network architecture and parameters, and can efficiently achieve an almost 100% attack success rate within a few iterations guided by the gradient descent [8, 16, 29, 31]. However, white-box attacks are less practical for commercial systems like Google Cloud Vision API, where the model is inaccessible.

To this end, the black-box attacks, including transfer-based attacks and query-based attacks, are more practical where the adversary is only able to craft model inputs and obtain corresponding outputs. Transfer-based attacks [13, 25, 32] adopt the adversarial perturbation crafted from a surrogate white-box model and transfer it to the black-box victim model. They require less time consumption, but suffer from low attack performance, since the target model may be very different from the surrogate. To achieve a high attack success rate, recent works [9, 10] queried iteratively to estimate the gradients and then perform the white-box attacks, or approach the decision boundary first and then wander along it. Due to the high-dimensional input space, it is hard to find a feasible direction to optimize, which results in numerous queries and a high cost of time and money. Some extra efforts have been put on reducing the dimension of solution space, like utilizing latent variable space of autoencoder [39] or adopting low-resolution images [21]. Essentially, these methods reduce the solution space from the perspective of the spatial domain, and the reduction of dimension is limited since images with quite low-resolution will be unserviceable, which makes these methods still inefficient.

In this paper, we propose Projection & Probability-driven Black-box Attack (PPBA) towards achieving a high

*Corresponding author.
[1]The code for reproducing our work is available at https://github.com/theFool32/PPBA

attack success rate with few queries. Optimization in a high dimension is difficult and urges for a smaller solution space [11, 28]. On the other hand, some recent works [17, 34] have experimentally verified that adversarial perturbations tend to lie in the low-frequency space, which is a subspace of the original solution space. These both motivate us to form a smaller search space from the frequency perspective. Considering that, we first reduce the query number via reducing the solution space with a low-frequency constrained projection matrix. In particular, we view this problem as recovering adversarial perturbations using compressed sensing with a sensing matrix. This sensing matrix can be crafted by applying the inverse Discrete Cosine Transform (DCT) [1] on the standard basis and selecting the low-frequency parts. The sensing matrix is plug-and-play as a projection matrix. With the elaborate sensing matrix, we reduce the dimension of the solution space from that of the image space (*e.g.*, $224 \times 224 \times 3 = 150,528$) to a very small one (*e.g.*, $1,500$). Based on this sensing matrix, we then propose a more suitable attack strategy driven by probability. We care merely about the direction of each dimension, upon which we quantize the value of every iteration into a triplet. Then, a probability-driven strategy is kicked in to take advantage of information throughout the iteration process to perform a random walk optimization.

Extensive experiments show the efficiency of the proposed PPBA method. By integrating the proposed low-frequency sensing matrix into various existing methods, we verify that it is flexible enough, which can reduce 9.6% queries with a higher attack success rate for VGG-16 [35] on ImageNet [12]. PPBA further improves the performance over the state-of-the-art methods [18, 20, 21] with at least 11% fewer queries for Inception v3 [37] on ImageNet. Finally, we evaluate PPBA on the real-world image classifier, *i.e.*, Google Cloud Vision API, and show that our method can efficiently corrupt it with an 84% success rate.

Concretely, the contributions of this work are as follows:

- We view generating adversarial perturbations as recovering sparse signals and propose a low-frequency sensing matrix to efficiently reduce the dimension of the solution space. The sensing matrix is plug-and-play and can be integrated into existing methods functioned as a projection matrix.
- Based on this projection matrix, a probability-driven attack method is proposed, which suits the sensing matrix more and makes the best use of the information throughout the whole iteration process.
- The proposed PPBA method achieves higher performance on different neural networks [19, 35, 37] pre-trained on ImageNet [12], compared with state-of-the-art methods [18, 20, 21], and can fool real-world systems efficiently.

2. Related Work

2.1. White-Box Attacks

The adversary under the white-box settings has full knowledge of the victim model. Szegedy *et al.* [38] first demonstrated that intentionally perturbated images, *e.g.* by adding quasi-imperceptible adversarial perturbations, can fool neural networks. These adversarial perturbations can be crafted with box-constrained L-BFGS [26]. Subsequently, various methods have been proposed to generate such perturbations. For example, Goodfellow *et al.* [16, 23] took a linear view of adversarial examples and proposed fast ways of generating them in one step or iteratively. Moosavi-Dezfooli *et al.* [31] attempted to find adversarial examples from the decision boundary. Carlini *et al.* [8] compared different objective functions and proposed a powerful C&W attack method. Note that these methods perform optimization with the gradient information, which cannot be applied to black-box attacks directly.

2.2. Black-Box Attacks

White-box attacks are unrealistic for many real-world systems, where neither model architectures nor parameters are available. Under this scenario, black-box attacks are necessary. In black-box attacks, the adversary is unable to access the target victim model, and only the model inputs and its corresponding outputs can be fetched. In this paper, we assume that the outputs include prediction confidences since it is a common setting for popular online systems, *e.g.*, Google Cloud Vision, Clarifai, and Microsoft Custom Vision. There are two types of black-box attack methods, *i.e.*, transfer-based attacks and query-based attacks:

Transfer-Based Attacks. Since models trained on the same dataset may share similar decision boundaries, adversarial examples can transfer across models to some degree. Considering that, the adversary performs a standard white-box attack on accessible local models to construct adversarial examples, which are expected to be transferred to the inaccessible target model. One type of such attack assumes that the local model and target model are trained with data from similar distributions, and no query on the target model is needed [27, 30, 38]. Another type of such attack is to distill the target model with a surrogate model [25, 32], which requests a large number of queries to train the local model and is thus inefficient. Although there exist many works focusing on improving the transferability of adversarial examples [13, 41, 42], the attack success rates of transfer-based attacks are still less competitive to query-based attacks.

Query-Based Attacks. Query-based attacks define an objective function and update the perturbation iteratively to optimize this function. Each iteration requires one or more queries to determine the next step. Authors in [13, 36] constructed the adversarial perturbations with an evolution-

ary algorithm. The efficiency of evolutionary algorithms is highly dependent on the dimension of the inputs and the size of the solution space, which makes these algorithms time-consuming. The authors in [3, 9] proposed the decision-based attack that initiates perturbations from a target image or with a large norm to guarantee adversarial and then reduces the norm iteratively along the decision boundary. Despite a high success rate, this kind of method requires a large number of queries wandering along the decision boundary as the boundary can be potentially complex. Another mainstream of query-based attacks is to estimate the gradients and then perform the white-box attack. Chen *et al.* [10] proposed the ZOO (zeroth-order optimization) attack that adopts the finite-difference method with dimension-wise estimation to approximate the gradient values. It takes $2d$ queries in each iteration, where d is the dimension of the input image (d can be more than $150,000$). Bhagoji *et al.* [2] attempted to reduce the query budget in each iteration via random grouping or PCA components mapping. Tu *et al.* [39] utilized a pretrained autoencoder and optimized the perturbations in the latent space. Instead of using the finite-difference method, Ilyas *et al.* [20] proposed the NES attack that adopts the natural evolution strategy to estimate gradients with random vectors. The Bandits$_{TD}$ is further proposed in [21], which incorporates time and data-dependent information with the bandit theory to reduce the query cost. Guo *et al.* [18] proposed the SimBA-DCT that adds or subtracts random vectors iteratively from a set of orthonormal vectors to craft adversarial examples. Considerable query cost is reduced by the aforementioned methods, which is however still far from satisfactory.

3. The Proposed Method

The large solution space for black-box and the inefficient optimization retain as two key bottlenecks for existing black-box attack methods. To solve these two issues, we first reduce the solution space from its original dimension with a low-frequency constrained sensing matrix, as detailed in Sec. 3.2. Then, to further reduce the query cost, we propose a novel weighted random walk optimization based on the sensing matrix, as described in Sec. 3.3.

3.1. Preliminaries

Given a deep neural network classifier $f : [0,1]^d \rightarrow \mathbb{R}^K$ that maps the input image x of d dimensions into the confidence scores of K classes, we define $F(x) = \arg\max_k f(x)_k$ as the function that outputs the predicted class. The goal of adversarial attack against classification is to find a perturbation $\delta \in \mathbb{R}^d$ that satisfies:

$$F\big(\Pi_{Img}(x+\delta)\big) \neq F(x), \text{ s.t. } \|\delta\|_p < \epsilon, \tag{1}$$

where $\Pi_{Img}(\cdot) = clip(\cdot, 0, 1)$[2] is a projection function that projects the input into the image space, *i.e.*, $[0,1]^d$, and ϵ is a hyper-parameter to make the perturbation invisible via restricting the l_p-norm. To achieve the goal, we adopt the widely used objective function termed C&W loss [8]:

$$\min_{\|\delta\|_p<\epsilon} L(\delta) = [f(x+\delta)_t - \max_{j\neq t}\big(f(x+\delta)_j\big)]^+, \tag{2}$$

where $[\cdot]^+$ denotes the $\max(\cdot, 0)$ function, t is the label of the clean input. For iterative optimization methods, to guarantee the constraint $\|\delta\|_p < \epsilon$, another projection function is needed after each update of δ. For instance, for l_2-norm, a projection function $\Pi_2(\delta, \epsilon) = \delta * min(1, \epsilon/\|\delta\|_2)$ should be applied in each step.

3.2. Low-Frequency Projection Matrix

3.2.1 Perspective from Compressed Sensing

Recent works have discovered that adversarial perturbations are biased towards the low frequency information [17, 34]. Suppose there exists an optimal low-frequency perturbation δ^* that is sparse in the frequency domain for Eq. (1). Thus, $\Psi\delta^*$ should be a sparse vector, where $\Psi \in \mathbb{R}^{d\times d}$ is the transform matrix of DCT that maps a vector from the time/spatial domain to the frequency domain and satisfies $\Psi\Psi^T = \Psi^T\Psi = I_{d\times d}$, where $I_{d\times d}$ is the indentify matrix.. According to the compressed sensing theory [5, 14], we can recover the sparse vector with a measurement matrix $\Phi \in \mathbb{R}^{m\times d}(m \ll d)$ and the corresponding measurement vector $z \in \mathbb{R}^m$ by:

$$\begin{aligned} &\min \|\Psi\delta^*\|_2, \\ &\text{s.t. } z = A\delta^* = \Phi\Psi\delta^*, \\ &\quad F(x+\delta^*) \neq F(x), \end{aligned} \tag{3}$$

where $A = \Phi\Psi \in \mathbb{R}^{m\times d}$ is the sensing matrix.

The measurement matrix Φ can be further simplified as $\Phi = [\Phi_m, \mathbf{0}], (\Phi_m \in \mathbb{R}^{m\times m}, \mathbf{0} \in \mathbb{R}^{m\times(d-m)})$, to suppress high frequency, considering that δ^* is biased to low frequency. Note that orthogonal matrices do not change the norm of a vector after transforming it, which also guarantees the restricted isometry property [4, 6] required by the compressed sensing theory. Therefore, we directly set Φ_m as an orthogonal matrix, and recover the perturbation δ^* with simple matrix multiplication as:

$$\begin{aligned} z &= \Phi\Psi\delta^*, \\ \Phi^T z &\approx \Psi\delta^*, \\ \Psi^T\Phi^T z &= A^T z \approx \delta^*. \end{aligned} \tag{4}$$

[2]For simplicity, we will omit it in what follows.

Finally, Eq. (3) can be rewritten as:

$$\begin{aligned} &\min \|z\|_2, \\ &\text{s.t. } F(x + A^T z) \neq F(x). \end{aligned} \tag{5}$$

As a result, we only need to perform the optimization in the m-dimensional space instead of the d-dimensional one ($m \ll d$), which results in a smaller solution space and the higher optimization efficiency.

3.2.2 Perspective from Low Frequency

From another perspective, we discover that the measurement vector z optimized in Eq. (5) has its physical meaning. The optimal perturbation vector δ^* can be linearly represented by the discrete cosine basis as below:

$$\delta^* = \sum_j \alpha_j \omega_j, \tag{6}$$

where w_j is a discrete cosine basis vector, and a_j is the corresponding coefficient. Note that w_j contains the specific frequency information, we can also view Eq. (6) as decomposing δ^* into the sum of different frequency vectors and the α_j is the corresponding amplitude. Since the vector ω_j can be easily crafted via applying inverse DCT on one of the standard basis vectors, we then rewrite Eq. (6) into the form of matrix multiplication as below:

$$\delta^* = \Omega\alpha = \Psi^T Q\alpha, \tag{7}$$

where Ω is a matrix formed by the frequency vectors ω_j as its columns, Ψ^T is the transform matrix of inverse DCT as mentioned before, $Q \in \mathbb{R}^{d\times m}$ is a submatrix subsampled from the standard basis $I_{d\times d}$ for low frequency, and α is the amplitude vector.

Comparing Eq. (7) with Eq. (4), it is inspiring to find that:

$$\begin{cases} \delta^* = \Psi^T Q\alpha, \\ \delta^* \approx \Psi^T \Phi^T z, \end{cases} \Rightarrow \Psi^T Q\alpha \approx \Psi^T \Phi^T z.$$

Since Q is orthogonal, it suggests a simple and efficient way to construct the sensing matrix A by applying inverse DCT to the standard basis[3], and the measurement vector z is just the amplitude α in Eq. (7).

3.3. Probability-Driven Optimization

As discussed in Sec. 3.2.2, the measurement vector z can be viewed as the amplitude. Therefore, the change of z in each iteration can be simplified by a triplet $\{-\rho, 0, \rho\}$, denoting decreasing the corresponding amplitude value by ρ, keeping it, and increasing it by ρ, respectively. Then the choice space of the iteration step is further restricted. Based on this setting, a random walk optimization is further adopted, which chooses steps randomly and moves when the step makes the loss descend.

To achieve better performance, instead of adopting the random steps, we make the best of information in the past by assuming that the directions of steps in the past are capable of guiding the choice of the current step to a certain degree. We rewrite the objective function in Eq. (2) as:

$$L(z, A) = [f(x + A^T z)_t - \max_{j\neq t} \big(f(x + A^T z)_j\big)]^+, \tag{8}$$

where an iterative optimization method like random walk can be applied. In particular, after defining Δz as the change of z in an iteration of random walk, a confusion matrix is calculated for each dimension Δz_j of Δz as:

	$-\rho$	0	ρ
# effective steps	$e_{-\rho}$	e_0	e_ρ
# ineffective steps	$i_{-\rho}$	i_0	i_ρ

where $e_{-\rho}$ means the number of times the loss function (*e.g.*, Eq. (2)) descends when $\Delta z_j = -\rho$, and $i_{-\rho}$ means the number of times the loss function keeps still or ascends when $\Delta z_j = -\rho$. We calculate the effective rate for every possible value with:

$$P(\text{effective}|\Delta z_j = v) = \frac{e_v}{e_v + i_v}, \text{ for } v \in \{-\rho, 0, \rho\}, \tag{9}$$

and sample Δz_j with a probability as:

$$P(\Delta z_j = v) = \frac{P(\text{effective}|\Delta z_j = v)}{\sum_u P(\text{effective}|\Delta z_j = u)}, \quad \text{for } v \text{ and } u \in \{-\rho, 0, \rho\}. \tag{10}$$

Therefore, when $\Delta z_j = v$, an effective query step increases the value of e_v along with $P(\text{effective}|\Delta z_j = v)$, which results in an increase in $P(\Delta z_j = v)$, and an ineffective one vice versa.

We prove that with T iterations, the norm of perturbation δ is bounded by:

$$\begin{aligned} \|\delta\|_2 &= \|A^T z\|_2 = tr\big((A^T z)^T A^T z\big) \\ &= tr(z^T A A^T z) = tr(z^T z) = \|z\|_2 \\ &= \|\sum_j \Delta z^j\|_2 \leq \|T \times \vec{\rho}\|_2 \\ &= \sqrt{m} \times T \times \rho, \end{aligned} \tag{11}$$

where Δz^j is the Δz of the j-th iteration and $\vec{\rho}$ is a vector with all elements of value ρ. Despite the above prove, we still utilize the projection function $\Pi_2(\cdot)$ to keep the norm constraint. For each iteration, we evaluate whether $\Pi_2(z + \Delta z)$ can successfully decrease the objective function and update the confusion matrices. We accept the Δz

[3]For 2D images, we utilize 2D IDCT, *i.e.*, utilize 1D IDCT twice.

Algorithm 1 Projection & Probability-Driven Black-Box Attack

Input: Input image x, maximum number of queries max_iter.
Output: Perturbation vector δ.

Initialize $z \leftarrow \mathbf{0} \in \mathbb{R}^m$, confusion matrices with all elements of 1, and $j \leftarrow 0$
Construct sensing matrix A via applying IDCT to the submatrix of $I_{d\times d}$
for $j < max\_iter$ **do**
  Generate Δz according to Eq. 10
  if $L\big(\Pi_2(z+\Delta z), A\big) < L(z, A)$ for L in Eq. (8) **then**
    $z \leftarrow \Pi_2(z + \Delta z)$
  end if
  if $L(z, A) <= 0$ **then**
    break.
  end if
  Update the confusion matrices accordingly
  $j \leftarrow j + 1$
end for
return $\delta = A^T z$

and update $z = \Pi_2(z + \Delta z)$ only if it succeeds. The above process repeats until we find an adversarial perturbation or meet the maximum number of iterations. We refer to this method as Projection & Probability-driven Black-box Attack (PPBA), and the detailed algorithm is provided in Alg. 1.

4. Experiments

4.1. Experimental Setups

Datasets and Victim Models. We evaluate the effectiveness of our proposed PPBA along with baselines on ImageNet [12]. For each evaluation, we sample $1,000$ images (one image per class) randomly from the validation set for evaluation. For the victim models, we choose the widely-used networks pre-trained on ImageNet, *i.e.*, ResNet50 [19], VGG-16 [35], and Inception V3 [37]. Considering the charge cost of Google API ($1.50 for $1,000$ queries), we randomly select 50 images to evaluate the results on the Google Cloud Vision API[4].

Evaluation Metric. Restricting the norm of the resulting perturbation, there are two aspects to evaluate black-box adversarial attacks: How often a feasible solution can be found, and how efficient the optimization method is. The attack success rate can quantitatively represent the first one. We define a successful attack for ImageNet as the one that changes the top-1 predicted label within the maximum queries. For the second one, the average number of queries (abbreviated to average queries) can give a rough sense. We report the average queries on both success samples and all samples. The average on success samples denotes how many queries are needed to successfully perturb an input, which is more useful. However, it is strongly connected to the success rate and thus gives a false sense for low success rate attack methods. We thereby report the average on all samples as a supplement. Considering that samples with a large number of queries have large impacts on the average value, we further depict the curve of the success rate versus the number of queries and calculate the area under the curve (AUC) for a better comparison.

Compared Methods and Settings. We mainly compare our proposed PPBA with NES [20], Bandits$_T$ [21] (Bandits with the time-dependent prior), Bandits$_{TD}$ [21] (Bandits with the time and data-dependent prior), and SimBA-DCT [18]. We evaluate the performance of the baselines with the source code released by the authors[5], and use the default parameters setting in their papers. We also perform the random walk with each step uniformly sampled from the triplet space to test the efficiency of our proposed probability-driven strategy. We name this kind of attack as *Projection & Random walk Black-box Attack* (PRBA). Following the settings in [21], we set the maximum l_2-norm for perturbations to 5, and the maximum l_∞-norm to 0.05. Since $10,000$ is a huge number in reality, we set the maximum number of queries to $2,000$ instead, and set ρ to 0.01.

4.2. On the Perturbation δ^* and Dimension m

Before evaluating the effectiveness of our proposed method, we first verify that the perturbation δ^* exists under the sensing matrix setting, and determine the value of dimension m experimentally. In this experiment, we utilize another 100 images randomly sampled from ImageNet for validation.

4.2.1 On the Existence of Perturbation δ^*

To verify that the existence of adversarial perturbation δ^* in the low-frequency space constrained by our sensing matrix, we first perform the white-box attack using the BIM method [23] with/without the sensing matrix A. The results are shown in Tab. 1. All attacks achieve a 100% success rate, which demonstrates that there indeed exist optimal perturbation with the sensing matrix constrained. Interestingly, we discover that perturbations found with the low-frequency constraint tend to have a much smaller average l_2-norm, which is consistent with the results in [17].

[4] https://cloud.google.com/vision/docs/drag-and-drop

[5] https://github.com/MadryLab/blackbox-bandits, https://github.com/cg563/simple-blackbox-attack

	Success Rate			Average L_2-Norm		
	R	V	I	R	V	I
BIM	100%	100%	100%	4.03	3.90	5.00
BIM+Sensing Matrix A	100%	100%	100%	2.06	1.70	1.86

Table 1. Results for BIM attack and BIM with our sensing matrix A under the white-box setting. R, V, and I denote ResNet50, VGG-16, and Inception V3, respectively. The 100% success rate verifies the existence of the adversarial perturbation.

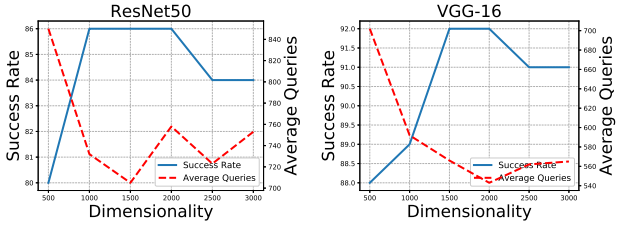


Figure 1. The effect of the dimensionality.

4.2.2 On the Choice of Dimension m

For determining the dimension of the measurement vectors, the Johnson–Lindenstrauss lemma [22] suggests that for a group of n points in $\mathbb{R}^d$, there exists a linear map $g : \mathbb{R}^d \to \mathbb{R}^m$, $m > 8\ln(n)/\epsilon^2$ that keeps the distance between these points:

$$(1-\epsilon)\|u-v\|^2 \leq \|g(u)-g(v)\|^2 \leq (1+\epsilon)\|u-v\|^2, \quad (12)$$

where u and v are two points, and $\epsilon \in (0,1)$ is a parameter controlling the quality of the projection. However, since it is hard to count the number of points in the low-frequency space, we leave the theoretical discovery of the value of m in our future work. Instead, we choose it experimentally in this paper. As depicted in Fig. 1, we evaluate the success rate and average queries by setting different values of m for ResNet50 and VGG-16. Intuitively, a moderate value is needed since a small dimension degenerates the representation ability and a large dimension enlarges the search space. Consistent with our intuition, the algorithm is hard to find a satisfying solution with a small dimension, which results in a poor success rate and more queries. With a large dimension and a large solution space, the algorithm needs more steps to find the optimal solution. As a result, we set m to $1,500$, $2,000$ and $4,000$ for ResNet50, VGG-16 and Inception V3, respectively.

4.3. On the Effect of the Sensing Matrix

As aforementioned, the sensing matrix we design can be integrated into existing methods. We evaluate the performance after plugging it into NES and Bandits$_T$. Note that the data-dependent prior [21] used in Bandits$_{TD}$ is also a kind of dimensionality reduction (as reduced from the original image space to a low-resolution space with the dimension of $50 \times 50 \times 3 = 7,500$), we compare the performance of our sensing matrix with it. Another projection matrix, *i.e.*, random Gaussian matrix, is also evaluated. The quantitative results conducted on $1,000$ randomly selected images can be found in Tab. 2. The results of the random Gaussian matrix show no positive influence on the performance with lower success rates and similar numbers of queries. On the contrary, the sensing matrix designed with the low-frequency constraint can at most improve the success rate by 27.2%, and reduce nearly 26% queries. Taking Bandits$_T$ as baseline, the data-dependent prior from Bandits$_{TD}$ is an effective method that can reduce 25% queries approximately with a 20.6% success rate improved. However, our sensing matrix is more effective that can reduce 34% queries with a 24.2% success rate improved. To explain, the solution space our sensing matrix maps to is much smaller than the one from the data-dependent prior, and the optimal perturbation exists in this space. Finally, it is worth noting that the low-frequency sensing matrix is plug-and-play, which can improve the performance of other methods efficiently.

Methods	ResNet50		VGG-16		Inception V3	
	ASR	Queries	ASR	Queries	ASR	Queries
NES	52.0%	1078/1521	60.7%	1013/1402	26.1%	1146/1777
NES+Gaussian	50.7%	1035/1511	59.0%	999/1410	25.1%	1112/1776
NES+Ours	**79.2%**	**896/1125**	**78.3%**	**873/1117**	**48.7%**	**958/1493**
Bandits$_T$	54.1%	719/1306	62.9%	679/1169	34.0%	866/1615
Bandits$_{TD}$	74.7%	621/970	78.6%	565/871	55.9%	701/1274
Bandits$_T$+Ours	**78.3%**	**552/867**	**79.5%**	**474/787**	**56.8%**	**668/1243**

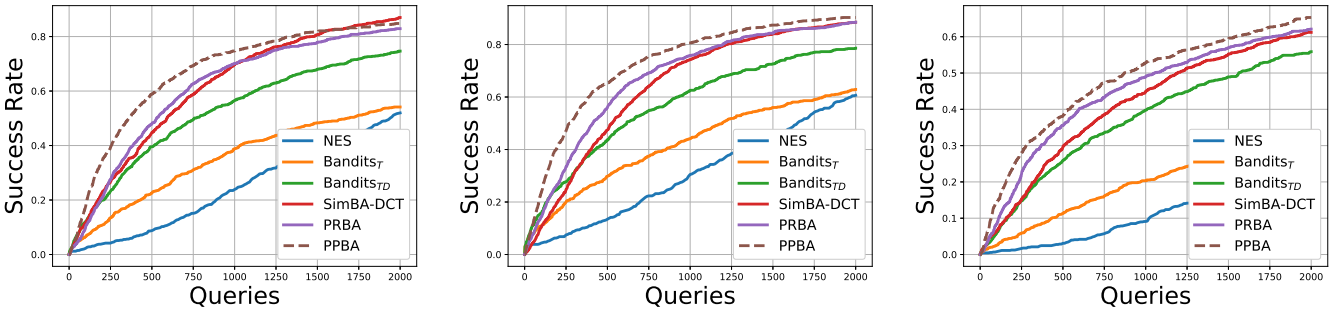
Table 2. Results of the sensing matrix. Gaussian means the random Gaussian matrix. ASR represents the attack success rate (higher is better). Queries denote the average queries, under which the left number is the amount on success samples and the right number is the amount on all samples (lower is better).

4.4. The Results of PPBA on ImageNet

We evaluate the performance of our proposed PPBA in Tab. 3 with the maximum l_2-norm of perturbation set to 5 as in [21]. Compared with NES and Bandits$_{TD}$, PRBA and PPBA both achieve a higher success rate with fewer queries. For example, PRBA improves 12.3% success rate and reduces 24% queries compared with Bandits$_{TD}$ for ResNet50, and PPBA achieves even better results. Compared with SimBA-DCT, PRBA and PPBA obtain competitive results except for the success rate on ResNet50. The average queries of PPBA are at most 24% fewer than SimBA-DCT, which makes PPBA standout. The PPBA method is better than PRBA with a 1.9% higher success rate and 15% fewer queries taking ResNet50 for example, which demonstrates the efficiency of the probability-driven strategy. To further investigate the relationship between the success rate and the number of queries, we plot the curves of success rate versus queries in Fig. 2. From these curves, we conclude that for more samples, PPBA finds feasible solutions within $2,000$ queries more quickly. The AUC has been calculated for a quantitative comparison in Tab. 3, which also indicates the superiority of our method.

Methods	ResNet50			VGG-16			Inception V3		
	ASR	Queries	AUC	ASR	Queries	AUC	ASR	Queries	AUC
NES	52.0%	1078/1521	481.5	60.7%	1013/1402	601.3	26.1%	1146/1777	224.3
Bandits$_T$	54.1%	719/1306	694.2	62.9%	679/1169	831.5	34.0%	866/1615	385.7
Bandits$_{TD}$	74.7%	621/970	1030.4	78.6%	565/871	1129.3	55.9%	701/1274	726.6
SimBA-DCT	**87.0%**	604/779	1214.3	88.5%	563/722	1271.5	61.2%	672/1181	812.6
PRBA	82.9%	540/790	1210.6	88.5%	489/663	1337.2	62.1%	580/1118	882.3
PPBA	84.8%	**430/668**	**1331.3**	**90.3%**	**392/548**	**1451.5**	**65.3%**	**546/1051**	**948.9**

Table 3. Results of l_2 attack for different methods.

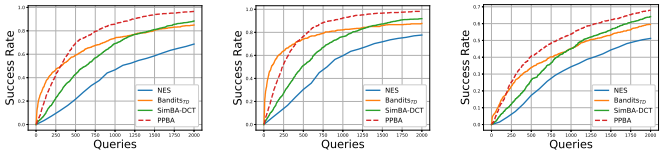


(a) ResNet50 (b) VGG-16 (c) Inception V3

Figure 2. Curves of the attack success rate versus the number of queries for l_2 attack.

Methods	ResNet50			VGG-16			Inception V3		
	ASR	Queries	AUC	ASR	Queries	AUC	ASR	Queries	AUC
NES	68.7%	867/1222	812.7	77.7%	745/1026	1013.2	51.2%	848/1411	606.7
Bandits$_{TD}$	84.9%	**409**/648	1352.7	87.7%	**238**/454	1526.6	59.9%	592/1162	836.9
SimBA-DCT	88.4%	646/797	1197.8	91.9%	556/667	1327.7	64.2%	747/1190	804.5
PPBA	**96.6%**	427/**481**	**1519.6**	**98.2%**	337/**367**	**1633.1**	**67.9%**	**566/1026**	**974.2**

Table 4. Results of l_∞ attack for different methods.

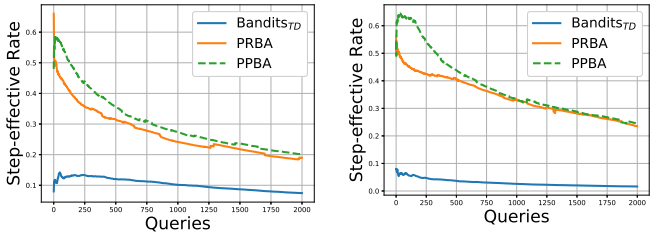


(a) ResNet50 (b) VGG-16 (c) Inception V3

Figure 3. Curves of the attack success rate versus the number of queries for l_∞ attack.

We evaluate the effectiveness of PPBA under the l_∞-norm as well. Following the setting in [21], the maximum l_∞-norm of perturbations is set to 0.05. The quantitative results are shown in Tab. 4, and the curves of the attack success rate versus the number of queries are depicted in Fig. 3. Similar to the results of l_2 attack, PPBA shows advantages over the baselines with at most 8.2% success rate improved and 26% queries reduced. We can see that PPBA is more effective and practical enough compared with the state-of-the-art methods.

To further investigate why PPBA is effective, we measure how often the optimization steps found by the algorithm can bring a descent on the objective function. Such a step is defined as an effective one, by which the step-effective rate is calculated. We select 50 images randomly for validation, and plot the curves of the step-effective rate versus the number of queries for PRBA and PPBA, along with Bandits$_{TD}$ in Fig. 4. As depicted in the figure, we find that PPBA achieves more than 20% step-effective rate throughout the optimization process, while those of Bandits$_{TD}$ are less than 20%. These results show that the restriction on the optimization step and the probability-driven strategy improve the sample efficiency, and help PPBA find feasible solutions quickly.



(a) ResNet50 (b) VGG-16

Figure 4. Curves of the step-effective rate versus the number of queries for PRBA, PPBA, and Bandits$_{TD}$.

4.5. On Attacking Google Cloud Vision

To demonstrate the effectiveness of our method against real-world online systems, we conduct attacks against the Google Cloud Vision API, which is an online service that offers powerful classification models. Attacking this system is significantly harder than attacking models pre-trained on ImageNet, since the model is trained on more classes and the exact classes are unclear, while only the top-k labels with their corresponding probabilities for input images can be obtained. We aim to attack the system by removing

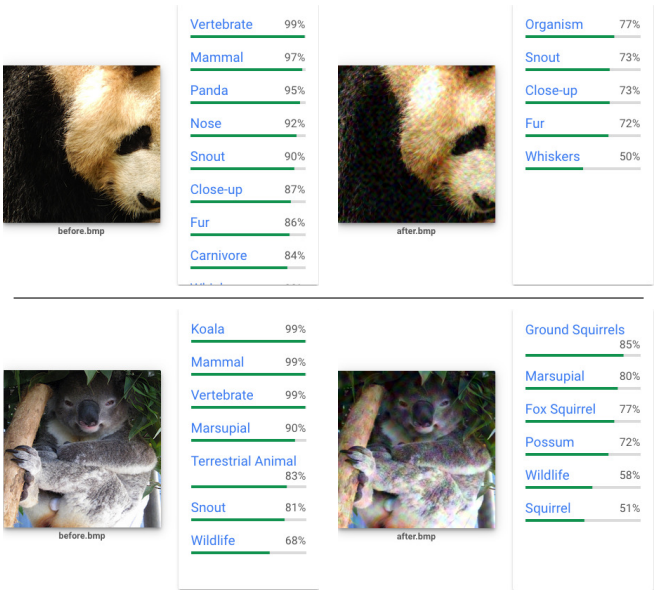


Figure 5. Examples of attacking the Google Cloud Vision API. The images on the left are the original ones and the images on the right are perturbed by PPBA to remove the top-3 labels. Taking the first row for example, the Google Cloud Vision API knows the left one is a *panda*, but the right one is not a *panda* anymore.

the top-3 labels presented in the original returned list. As in [18], we set the adversarial loss as the maximum of the original top-3 labels' returned probabilities, and minimize the loss with our PPBA. Fig. 5 shows two attack examples. The images on the left are the original ones and the images on the right are perturbed by PPBA. Taking the first row for example, the top-3 labels from the original returned list are related to *panda* with more than 95% probabilities. After perturbed by PPBA, the concepts related to *panda* disappear from the list, and the Google Cloud Vision API gives the labels related to the local content of the image.

Considering the cost the Google Cloud Vision API charges, we evaluate our method on 50 randomly selected images. We adopt a larger $\rho = 0.1$, and set the maximum l_∞-norm as $16/255$ ($16/255$ is widely used in recent attack competitions[6]). As a result, PPBA obtains an 84% success rate with 314 average queries on success samples under this setting, which demonstrates that PPBA is practical for real-world systems. More visual results are given in Fig. 6.

5. Conclusion

In this paper, we tackle the problem of the high query budget that black-box attacks suffer. We propose a novel projection & probability-driven attack, which mainly focuses on reducing the solution space and improving the optimization. Towards reducing the solution space, we propose to utilize a low-frequency constrained sensing matrix to reduce the dimensionality of the solution space, inspired by the compressed sensing theory and the low-frequency hypothesis. Based on the sensing matrix, we further propose a probability-driven optimization that makes the best use of all queries over the optimization process. We evaluate our proposed method on widely-used neural networks pre-trained on ImageNet, *i.e.*, ResNet50, VGG-16 and Inception V3, in which our method shows significantly higher attack performance with fewer queries compared with the state-of-the-art methods. Finally, we also attack the real-world system, *i.e.*, Google Cloud Vision API, with a success rate as high as 84%, which further demonstrates the practicality of our method. Last but not least, our work serves as an inspiration in designing more robust models. We leave it for our future work.

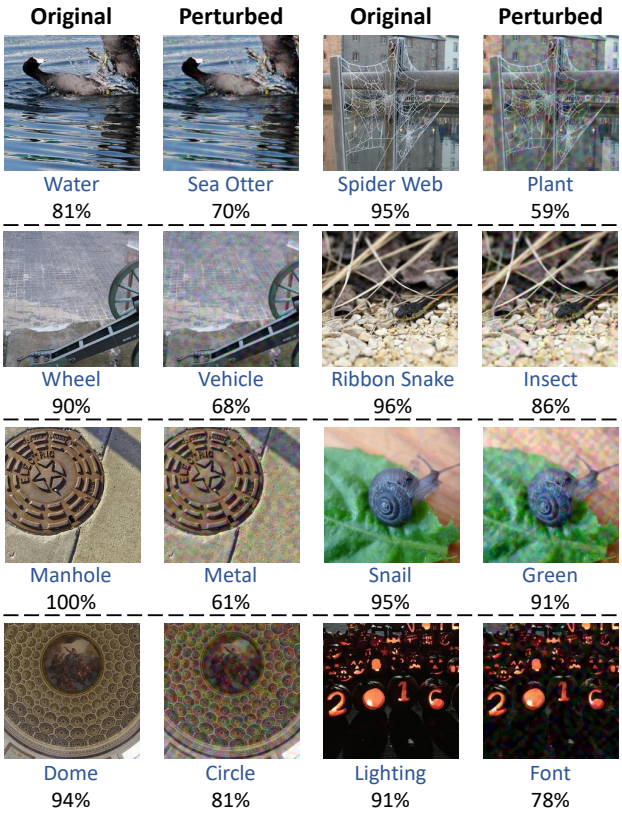


Figure 6. Visual examples for attacking the Google Cloud Vision API. In each pair, the left is the original image, and the right is the one perturbed by PPBA. The top-1 predictions with the probabilities are listed under the images.

Acknowledgements. This work is supported by the Nature Science Foundation of China (No.U1705262, No.61772443, No.61572410, No.61802324 and No.61702136), National Key R&D Program (No.2017YFC0113000, and No.2016YFB1001503), and Nature Science Foundation of Fujian Province, China (No. 2017J01125 and No. 2018J01106).

[6] https://www.kaggle.com/c/nips-2017-non-targeted-adversarial-attack, http://hof.geekpwn.org/caad/en/

References

[1] Nasir Ahmed, T. Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 1974.

[2] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Exploring the space of black-box attacks on deep neural networks. In *European Conference on Computer Vision*, 2019.

[3] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

[4] Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 2008.

[5] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 2006.

[6] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 2006.

[7] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *USENIX Security Symposium*, 2016.

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[9] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. *arXiv preprint arXiv:1904.02144*, 2019.

[10] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*, 2017.

[11] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Symposium on Computational Geometry*, 2004.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[13] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[14] David L Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 2006.

[15] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[17] Chuan Guo, Jared S Frank, and Kilian Q Weinberger. Low frequency adversarial perturbation. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

[18] Chuan Guo, Jacob R Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, 2019.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[20] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, 2018.

[21] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019.

[22] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 1984.

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations Workshop*, 2017.

[24] Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. Universal perturbation attack against image retrieval. In *International Conference on Computer Vision*, 2019.

[25] Pengcheng Li, Jinfeng Yi, and Lijun Zhang. Query-efficient black-box attack by active learning. In *IEEE International Conference on Data Mining*, 2018.

[26] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 1989.

[27] Hong Liu, Rongrong Ji, Jie Li, Baochang Zhang, Yue Gao, Yongjian Wu, and Feiyue Huang. Universal adversarial perturbation via prior driven uncertainty approximation. In *International Conference on Computer Vision*, 2019.

[28] Hong Liu, Rongrong Ji, Jingdong Wang, and Chunhua Shen. Ordinal constraint binary coding for approximate nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[30] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[31] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[32] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Asia Conference on Computer and Communications Security*, 2017.

[33] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[34] Yash Sharma, Gavin Weiguang Ding, and Marcus Brubaker. On the effectiveness of low frequency perturbations. In *International Joint Conference on Artificial Intelligence*, 2019.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[36] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[37] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[39] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Association for the Advancement of Artificial Intelligence*, 2019.

[40] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *International Conference on Computer Vision*, 2017.

[41] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[42] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *European Conference on Computer Vision*, 2018.