BO-DBA: QUERY-EFFICIENT DECISION-BASED ADVERSARIAL ATTACKS VIA BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Decision-based attacks (DBA), wherein attackers perturb inputs to spoof learning algorithms by observing solely the output labels, are a type of severe adversarial attacks against Deep Neural Networks (DNNs) that require minimal knowledge of attackers. Most existing DBA attacks rely on zeroth-order gradient estimation and require an excessive number (>20,000) of queries to converge. To better understand the attack, this paper presents an efficient DBA attack technique, namely BO-DBA, that greatly improves the query efficiency. We achieve this by introducing dimension reduction techniques and derivative-free optimization to the process of closest decision boundary search. In BO-DBA, we adopt the Gaussian process to model the distribution of decision boundary radius over a low-dimensional search space defined by perturbation generator functions. Bayesian Optimization is then leveraged to find the optimal direction. Experimental results on pre-trained ImageNet classifiers show that BO-DBA converges within 200 queries while the state-of-the-art DBA techniques using zeroth order optimization need over 15,000 queries to achieve the same level of perturbation distortion.

1 INTRODUCTION

Recent advances in computation and learning have made deep neural networks (DNNs) an important enabler for a plethora of applications. However, DNNs have also shown vulnerabilities to *adversarial examples* - a type of maliciously perturbed examples that are almost identical to original samples in human perception but can cause models to make incorrect decisions (Szegedy et al. (2013)). Such vulnerabilities can lead to severe and sometimes fatal consequences in many real-world DNN applications such as autonomous vehicles, financial services, and robotics. Therefore, it is critical to understand limitations of current learning algorithms and identify their vulnerabilities, which in turn helps to improve the robustness of learning.

According to the knowledge of attackers, adversarial attacks can be categorized into three primary types:*white-box attacks*, *score-based attacks*, and *decision-based attacks*. In the white-box setting (Goodfellow et al. (2014); Madry et al. (2017)), the attacker requires complete knowledge of the architecture and parameters of the target network. In score-based attacks (SBA), the attacker can only access the queried soft-label output (real-valued probability scores) of the target model. In decision-based attacks (DBA), also known as hard-label attacks, wherein only the hard label of a given input is available. Fig. 1 illustrates the accessible information of the target model for each of the three adversarial attacks.

Of the three attacks, DBA can lead to severe and ubiquitous threats to practical systems because of the minimal required knowledge of the victim model, and has attracted great interests recently. However, DBA is also the most challenging adversarial attack to design because of the relative insensitivity of model outputs to input perturbation - it is often difficult for the attacker to determine whether the change of perturbation is preferred or not when the target model's prediction does not change. To launch DBA attacks, an attacker shall discover the decision boundary where a slight change of perturbation will cause the model to yield different prediction labels. Following this idea, current research on DBA attacks formulates the problem of finding minimum adversarial perturbation (to make the attack stealthy) into the problem of finding the direction resulting in a minimum boundary radius. This approach is known as Cheng's formulation (Cheng et al. (2018)). Because of the smoothness of Cheng's formulation, most existing works (Chen et al. (2020); Cheng et al. (2018;

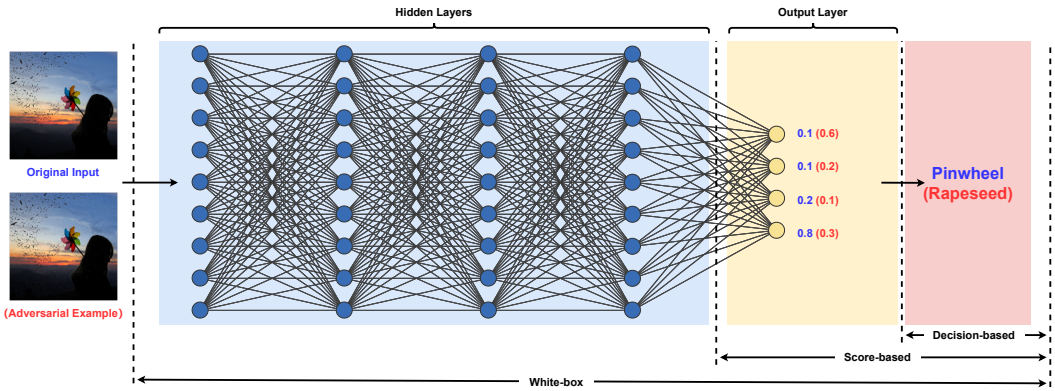

Figure 1: An illustration of the three types of adversarial attack. A white-box attack can access to the whole model; a score-based attack can access to the soft-label output layer; a decision-based attack can only access the predicted label. With an unnoticeable perturbation, "Pinwheel" is classified as "Rapeseed"

2019); Li et al. (2021)) solve the problem via zeroth-order optimization methods which require tens of thousands of queries to estimate the gradient in every optimization step.

In reality, cloud-based machine learning platforms often set a limit on the allowed number of queries within a certain period of time. For example, the Google cloud vision API currently allows 1,800 requests per minute. Therefore, improving query efficiency is critical for successful DBA attacks in practical systems. While current research is mostly focused on improving the efficiency of zeroth-order gradient estimation (Cheng et al. (2019); Chen et al. (2020); Li et al. (2021),Zhang et al. (2021)), the overall query complexity of DBA attacks remains impractically high. A recent work RayS (Chen & Gu (2020)) reduces the query complexity of (Cheng et al. (2018)) via hierarchical searching which is a derivative-free method. However, the searching space of RayS is the original data sample space which can be high dimensional and usually requires a large number of queries to find the optimal solution. Essentially, RayS is a straightforward search-based approach without exploitation of stochastic information like decision boundary radius distribution. These observations raise the following question - *can we solve Cheng's formulation in lower-dimensional sampling space via derivative-free optimization approach to achieve better query efficiency?*

In this paper, we answer this question affirmatively by proposing an efficient DBA attack, namely BO-DBA. We summarize our main contributions as follows:

- We propose a novel decision-based adversarial attack named BO-DBA, which for the first time solves Cheng's formulation via a derivative-free approach in low dimensional sampling subspace. We show that our method is more effective than previous decision-based attacks in the terms of query efficiency and attack success rate.
- Different from previous work, most of which rely on zeroth-order optimization methods that introduce tremendous query cost during gradient approximation, we use Gaussian Process to model an accurate probability distribution of decision boundary radius and find the optimal direction via Bayesian Optimization.
- Moreover, BO-DBA does not rely on the smoothness of Cheng's formulation as in previous zeroth order optimization methods with dimension reduction. Our method can support more complex or non-differentiable functions to achieve dimension reduction and allow the attacker to have better control over subspace selection.
- We demonstrate the superior efficiency of our algorithm over several state-of-the-art decision-based attacks through extensive experiments. For example, BO-DBA requires $75\times$ fewer queries than zeroth-order optimization methods and $5\times$ to $10\times$ fewer than RayS.

This rest of the paper is organized as follows: Section 2 reviews adversarial attacks and technical preliminaries. Section 3 describes our technical intuition and Section 4 elaborates the design of BO-DBA. Experimental results are provided in Section 5. We conclude the paper in Section 6.

2 RELATED WORK

2.1 BLACK-BOX ADVERSIAL ATTACKS

Black-box attacks are one type of adversarial attacks against learning systems where the attacker has no knowledge about the model and can only observe inputs and their corresponding output by querying the model. According to the types of model outputs, black-box attacks can be classified into **decision-based attacks** (DBA) and **score-based attacks** (SBA). SBA attackers can access the real-valued probability or the score of each output class while DBA attackers only can access the output labels which may not necessarily be real-valued.

SBA attacks assume the attacker can access the real-valued confidence scores such as class probabilities. Although there are some transfer based methods (Papernot et al. (2017; 2016)), effectiveness of these methods is not quite satisfactory because a carefully-designed substitute model or even access to part of the training data is required. On the other hand, optimization-based SBA aims to approximate gradient via zeroth-order optimization methods. ZOO (Chen et al. (2017)) adopts the zeroth-order gradient estimation to optimize the confidence score of perturbed inputs. One breakthrough of SBA design was made by Ru et al. (2019) and Shukla et al. (2019), which only need less than a thousand queries to fool the ImageNet classifier. Both attacks perform Bayesian optimization on some low dimensional inputs and then up-sample them to perturbations using traditional image up-sampling algorithm or deep generative models. Co et al. (2019) further improves the attack success rate by designing a perturbation generator that can produce natural details using procedural noises (Lagae et al. (2010)). Those Bayesian Optimization based methods model the distribution of soft-decision over the input space to locate the possible adversarial examples that satisfy the l_p norm constraint.

DBA attacks are detrimental to learning systems because the minimal requirements on the knowledge of attackers. There have been several DBA techniques in the literature. In Boundary Attack (Brendel et al. (2017)), a perturbed example is generated starting with a large perturbation sampled from a proposed distribution. It then iteratively reduces the distance of the perturbed example to the original input through a random walk along the decision boundary. Opt-Attack (Cheng et al. (2018)) first proposed Cheng's formulation which turns DBA problem into the problem of finding the optimal direction that leads to the shortest l_2 distance to decision boundary and optimized the new problem via zeroth-order optimization methods. Chen et al. (2020) and Cheng et al. (2019) furthermore improve Cheng et al. (2018)'s query efficiency via estimating the sign of gradient or optimizing the hyperparamters of optimizing procedural. Recently, Zhang et al. (2021) further reduce the query complexity of zeroth-order gradient estimation by projecting the input space into a low dimensional subspace as long as the projection does not violate the smoothness of Cheng's formulation. Meanwhile, Chen & Gu (2020) explores solving the Cheng's formulation via gradient free methods like hierarchical searching on input dimension.

2.2 BAYESIAN OPTIMIZATION

Bayesian optimization (BO) is a sequential optimization method particularly suitable for problems with low dimension and expensive query budgets Mockus (2012) such as black-box optimization. It contains two main components - a *probabilistic surrogate model*, usually a Gaussian Process (GP), for approximating the objective function, and an *acquisition function* that assign a value to each query that describes how optimal the query is.

Gaussian Process is a statistic surrogate that induces a posterior distribution over the objective functions Rasmussen (2003). In particular, a Guassian Process $\mathcal{GP}(\mu_0, \Sigma_0)$ can be described by a prior mean function μ_0 and positive-definite kernel or covariance function Σ_0. In this paper, we adapt the Matern $5/2$ Kernel Shahriari et al. (2015) as the covariance function, which is defined as:

$$\Sigma(x, x') = (1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2})exp(-\frac{\sqrt{5}r}{l})$$

where $r = x - x'$ and l is the length-scale parameter Snoek et al. (2012).

Acquisition Function in Bayesian optimization is a function that evaluates the utility of model querying at each point, given the surrogate model, to find the optimal candidate query point for

the next iteration Brochu et al. (2010). *Expected Improvement* (EI) and *Upper Confidence Bound* (UCB) are the two most popular acquisition functions that have been shown effective in real black-box optimization problems Shahriari et al. (2015). In black-box adversarial attacks, most studies Shukla et al. (2019); Co et al. (2019) adopted EI as the acquisition function because of its better convergence performance Shahriari et al. (2015); Snoek et al. (2012). In this paper, we also use EI as the acquisition function which is defined as:

$$EI_n(x) = \mathbb{E}_n[max(h(x) - h_n^*, 0)]$$

where h is the objective function and h_n^* is the best observed value so far. $\mathbb{E}_n[\cdot] = \mathbb{E}_n[\cdot|D_{1:n-1}]$ denotes the expectation taken over the posterior distribution given evaluations of h at $x_1, \cdots, x_{n-1}$.

3 TECHNICAL INTUITION

In this section, we introduce the technical intuition of BO-DBA attack. We first take a overview of the Cheng's formulation and analyze the limitation of previous proposed gradient-based methods. Then we describe the motivation of our design.

3.1 OVERVIEW OF CHENG'S FORMULATION

The classification model F takes images as inputs and outputs a $K-$dimensional vector which represents confidence scores over $K-$classes (we will take images as examples in the rest of this paper). In decision-based setting, we can define a Boolean-value objective function $h_b : [0,1]^d \rightarrow \{-1, 1\}$ as following:

$$h_b(\gamma) = \begin{cases} sign\{\max\limits_{i \neq y}[F_i(x+\gamma)] - F_y(x+\gamma)\} & \text{(Untargeted)} \\ sign\{F_k(x+\gamma) - \max\limits_{i \neq k}[F_i(x+\gamma)]\} & \text{(Targeted)} \end{cases} \tag{1}$$

where $x \in \mathbb{R}^d$ is the targeted data sample and $y \in \{1...K\}$ is its true label. $\gamma \in \mathbb{R}^d$ is the perturbation added to the input data. $k \in \{1...K\}$ represents the target label. Notes that the output of F is unavailable for decision-based attacker. So the objective function h_b can be consider as a black-box function:

$$h_b(\gamma) = \begin{cases} 1 & \text{Attack success} \\ -1 & \text{Attack failed} \end{cases} \tag{2}$$

Obviously, directly maximize h_b is very difficult because h_b is neither continuous nor differentiable. To overcome this problem, Cheng et al. (2018) reformulate the decision-based attack problem as:

$$\min_{\theta} g(\theta) \text{ where } g(\theta) = \arg\min_{\Delta > 0} \Big(h_b(x_0 + \Delta\theta) = 1 \Big) \tag{3}$$

In Cheng's formulation, $g(\theta)$ represents the decision boundary radius from input x along the ray direction $theta$. Then the DBA attack problem can be converted to find the ray direction with minimum decision boundary radius regarding the original example x. While most of the prior works focusing how to solve the formula by estimating the gradient through zeroth order optimization, the decision-only access makes solving (3) query inefficient. Specifically, the decision boundary radius $g(\theta)$ is typically estimated by a binary search procedure and approximation of the gradient of $g(\theta)$ via finite difference requires multiple rounds of computation of $g(\theta)$. RayS, on the other hand, adopts hierarchical searching to solve Cheng's formulation in a gradient-free fashion. However, straightforward searching will discard stochastic information that can be utilized in optimization methods like stochastic gradient or distribution of decision boundary radius. Moreover, RayS conducts the searching process on input space directly which will also introduce significant query complexity especially when the input dimension is large (e.g. color images).

In order to overcome the problems mentioned above, we propose BO-DBA attack which contains two critical design: (1) Bayesian Optimization is utilized to directly find the ray direction with the highest probability to generate the closest decision boundary and (2) a perturbation generator is adopted to reduce the dimension of the input search space.

3.2 TECHNICAL INTUITION

We first discuss how to reduce the input space by involving the perturbation generator in Cheng's formulation. We define perturbation generator as a function $S : \mathbb{R}^{d'} \to \mathbb{R}^d$ that takes low dimensional inputs $\delta' \in \mathbb{R}^{d'} (d' << d)$ and outputs an image-size perturbation $\delta \in \mathbb{R}^d$. Then we can formulate our objective function $g'(\delta')$ as:

$$\min_{\delta'} g'(\delta') \text{ where } g'(\delta') = \arg\min_{\Delta>0} \left(h_b(x_0 + \Delta \frac{S(\delta')}{|S(\delta')|}) = 1 \right) \tag{4}$$

In this formulation, we define the search direction in (3) using normalized perturbation generated by S: $\theta = \frac{S(\delta')}{|S(\delta')|}$. The value of g' can be evaluated via multiple decision-based queries which we will discuss in section 4. Note that, although prior work (Zhang et al. (2021)) has also adapted a projection matrix to reduce the searching space in decision-based attack, Zhang et al. (2021) still aims to solve problem (3) by approximating gradient $g'(\theta)$. This requires the projection matrix to preserve smoothness of the objective function. On the other hand, our method adopts derivative-free optimization which allows us to support more complex or non-differentiable perturbation generation functions to gain better control over searching space selection. For example, the Perlin noise generator will reduce the input space from all possible images into images of Perlin noise.

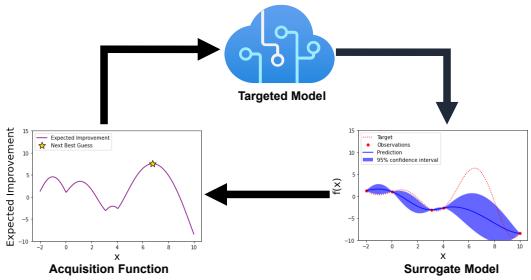

Figure 2: Workflow of Bayesian Optimization in BO-DBA

With objective function $g'(\delta')$ available, the optimization problem (4) is solvable using the Bayesian Optimization. Adopting the logic in Figure (2), the attacker will query the boundary radius $g'(\delta')$ on the searching direction θ generated by the low dimensional input δ'. The optimizer models the distribution of distances over the input space and acquires the next most possible optimal input for querying until an adversarial example near enough is found. In particular, for each iteration t, based on observation set $\{\delta'_i, g'(\delta'_i)\}_{i=1}^{t-1}$, we use Gaussian process to model the radius distribution of all possible directions. Then we use Acquisition Function to select δ'_t with the highest probability to generate the lowest radius (smallest perturbation) according to the statistic distribution. Then we query the model to compute $g'(\delta'_t)$ and add the result $\{\delta'_t, g'(\delta'_t)\}$ to the observation set.

Compared to the score-based methods that also adopt BO (Co et al. (2019); Ru et al. (2019)) (which formulates the SBA attack as a constrained optimization problem that maximizes the probability score of incorrect label), our optimization framework focuses on optimizing the boundary distance which can be evaluated via querying the decision-based model solely. Moreover, our algorithm does not rely on predefined distortion constraints like score-based BO, where the attacker needs to define the required boundary distance beforehand to trade the success rate for perturbation quality.

4 DECISION-BASED BAYESIAN OPTIMIZATION ATTACK

In this section, we describe an optimization framework for finding adversarial instances for a classification model F in detail. First we discuss how to compute $g'(\delta')$ up to certain accuracy using the Boolean-valued function h_b. Then we will solve the optimization problem via Bayesian Optimization and present our full algorithm.

Algorithm 1: Distance Evaluation Algorithm

input : Boolean-valued query function h_b of target model, original image x_0, low dimensional input δ', increase step size η, stopping tolerance ϵ, maximum distance Δ_{max}
output: $g'(\delta')$

$\theta \leftarrow \frac{S(\delta')}{S(\delta')}$; `// Compute the searching direction`
`// Fine-grained search`
if $h_b(x_0 + \eta\theta) = -1$ **then**
  $v_{low} \leftarrow x_0 + \eta\theta$, $v_{high} \leftarrow x_0 + 2\eta\theta$;
  while $h_b(v_{high}) = -1$ **do**
    $v_{low} \leftarrow v_{high}, v_{high} \leftarrow v_{high} + \eta\theta$;
    if $|v_{low}| \geq \Delta_{max}$ **then**
      return $g'(\delta') = \Delta_{max}$;
    end
  end
else
  $v_{low} \leftarrow 0$, $v_{high} \leftarrow x_0 + \eta\theta$;
end
`// Binary search between` $[v_{low}, v_{high}]$
while $|v_{high} - v_{low} > \epsilon|$ **do**
  $v_{mid} \leftarrow (v_{high} + v_{low})/2$;
  if $h_b(v_{mid}) = -1$ **then**
    $v_{high} \leftarrow v_{mid}$;
  else
    $v_{low} \leftarrow v_{mid}$;
  end
end
return $g'(\delta') = |v_{high}|$;

4.1 Distance Evaluation Algorithm

Algorithm 1 elaborates how to evaluate $g'(\delta')$ via queries on Boolean-value function h_b:

First, the attacker computes the search direction locally $\theta = \frac{S(\delta')}{|S(\delta')|}$. For a given low dimensional input δ', attacker first generates an image-size perturbation $S(\delta')$ via the perturbation generator S. Then normalize $S(\delta')$ into a unit vector $\frac{S(\delta')}{|S(\delta')|}$ to represent the search directionθ. It is easy to notice that for any given input δ', there is always a search direction θ that can be computed.

To evaluate the distance from input x_0 to the decision boundary along the direction θ, the attacker performs a fine-grain search and then a binary search. For simplicity, we assume the l_2 distance here, but the same procedure can also be applied to other distance measurements as long as vector operations are well defined in their respective spaces. In the fine-grained search phase, we cumulatively increase the search distance to query the points $\{x_0 + \eta\theta, x_0 + 2\eta\theta, \dots\}$ one by one until $h_b(x_0 + i\eta\theta) = 1$. Then we conduct a binary search between the interval $[x_0 + (i-1)\eta\theta, x_0 + i\eta\theta]$, within which the classification boundary is located. Note that, in practice the fine-grained search may exceed the numerical bounds defined by the image (or other type of samples). We can simply assign a maximum distance (e.g., the distance between all-black image and all-white image) for this searching direction. Unlike the gradient-based method that needs an accurate result to evaluate the gradient, Bayesian Optimization only needs statistical knowledge about each direction.

4.2 Bayesian Optimization

The detailed procedure of BO-DBA is presented in Algorithm 2. At beginning, we sample T_0 random low dimensional inputs δ' from the input space and evaluate the distance $g'(\delta')$ using Algorithm 1. Then we iteratively update the posterior distribution of the GP using available data D and query new δ' obtained by maximizing the acquisition function over the current posterior distribution of GP until a valid adversarial example within the desired distortion is found or the maximum number of

Algorithm 2: Bayesian Optimization for DBA

input : Targeted input x_0, Guassian process model GP, Acquisition function $\mathcal{A}$, Initialization sample size T_0, Maximum sample size T, Distance evaluation function $g'(\cdot)$, stopping tolerance ϵ, $D = \varnothing$.
output: Adversarial Examples x'
// Intialization
for $t = 0, 1, 2..., T_0 - 1$ **do**
  Generate input δ'_t randomly;
  $D \leftarrow D \cup (\delta'_t, g'(\delta'_t))$;
end
Update the GP on D;
// Optimization via GP and Acquisition function
while $t < T$ **do**
  $t \leftarrow t + 1$;
  $\delta'_t \leftarrow \arg\max_{\delta'} \mathcal{A}(\delta', D)$;
  if $|g'(\delta'_t)| > \epsilon$ **then**
    $D \leftarrow D \cup (\delta'_t, g'(\delta'_t))$ and update the GP;
  else
    $\theta = \frac{S(\delta'_t)}{|S(\delta'_t)|}$;
    return $x_0 + g'(\delta'_t)\theta$;
  end
end
// Return nearest adversarial example
$\theta = \frac{S(\delta'_*)}{|S(\delta'_*)|} \mid (\delta'_*, g(\delta'_*)) \in D$ **such that** $g(\delta'_*) \leq g(\delta') \quad \forall (\delta', g'(\delta')) \in D$;
return $x_0 + g'(\delta'_*)\theta$;

iteration is reached. Note that the query budget shall be larger than the number of iterations because we need multiple queries to evaluate the distance in Algorithm 1. The alternative stop condition of the optimization procedure is to set a maximum acceptable query budget.

5 Experiments

In this section, we carry out an experimental analysis of our BO-DBA attack. We first compare BO-DBA with other decision-based attack baselines on naturally trained models and models with run-time adversarial example detection (Xu et al. (2017)). Then, we examine how different types of perturbation generators affect attack success rate and perturbation quality. All experiments are carried out on a 2080 TI GPU, with code available online.[1]

5.1 Performance Evaluation

Baselines: We first compare our algorithm with opensouced state-of-the-art decision-based attacks: Opt-Attack (Cheng et al. (2018)), HSJA (Chen et al. (2020)), SignOPT (Cheng et al. (2019)) and RayS (Chen & Gu (2020)). For those attacks, we adopt the same hyperparamter settings in orignal papers.

Data and Models: We use two distinct DCN architectures pre-trained on ImageNet (Deng et al. (2009)): ResNet-50 (He et al. (2016)) and Inception V3 (Szegedy et al. (2016)). ResNet-50 takes input images with dimensions $224 \times 224 \times 3$ while Inception V3 take images with dimensions $299 \times 299 \times 3$. In addition to defenseless, we also carried out experiments on a classifier that has the run-time adversarial sample detection function (Xu et al. (2017)). The function detects abnormal inputs by comparing a DNN model's prediction on the original input with that on squeezed inputs (by reducing the color bit depth of each pixel or spatial smoothing).

[1] https://github.com/zzs1324/BO-DBA.git

Metrics: To measure the efficiency, we use the average l_∞ distance between perturbed and original samples over a subset of test images. For each method, we restrict the maximum number of queries to 1000. As an alternative metric, we also evaluate the *attack success rate (ASR)*. An adversarial attack is considered a success if the distortion distance between generated adversarial examples and original image does not exceed a given distance threshold. In this paper, we use the distance threshold ($l_\infty \leq \frac{16}{255}$) to define the ASR.

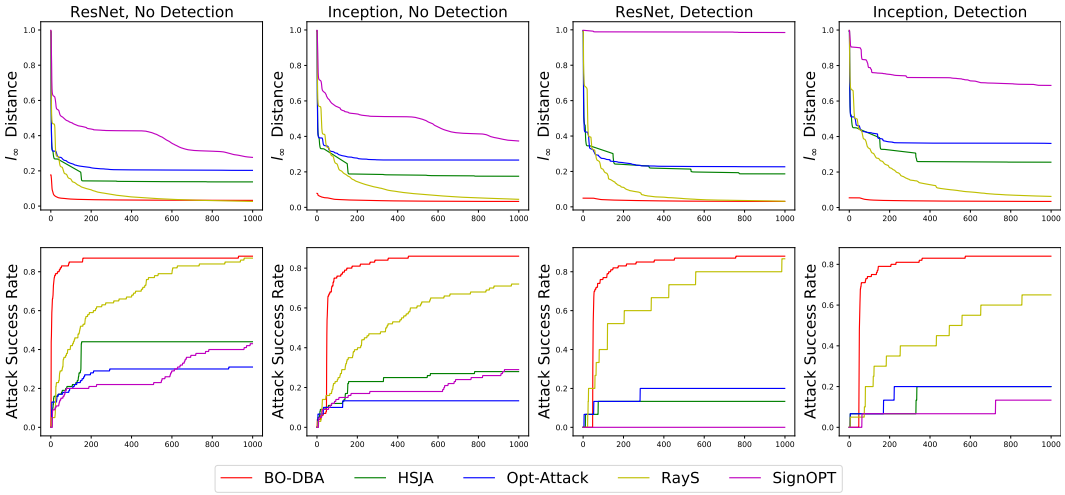

Figure 3: Average distance versus query budgets on ImageNet with ResNet, Inception, and ResNet with adversarial example detection, and Inception with adversarial example detection from left to right columns. 1st row: untargeted l_∞ distance. 2nd row: untargeted Attack Success Rate

Results: Figure 3 shows the average l_∞ distance and attack success rate against the query budgets. Column 1&2 compares the l_∞ distortion and attack success rate of our framework with four baseline DBA attacks on the defenseless classifier. We can see that BO-DBA consistently achieves a smaller distortion within 1000 queries than baseline methods. As a derivative-free method, BO-DBA can converge within 200 query budgets, while zeroth order optimization based techniques like OPT-Attack, HSJA and SignOPT[2] need over 15,000 queries to achieve the same level of perturbation distortion (Chen et al. (2020)). Although RayS adopts another derivative-free method, it is able to achieve similar perturbation distortion only after around 1000 queries. The obvious advantage of query efficiency of BO-DBA is mainly attributed to facts: 1) BO-DBA adopts the Bayesian Optimization to utilize the statistical information while RayS relies soley on a straightforward hierarchical searching; 2) BO-DBA reduces the searching space via perturbation generators, which results in a much higher convergence rate. Similar results can be seen in the Attack Success Rate as shown in row 2 of Figure 3. Column 3&4 of Figure 3 shows the l_∞ distortion and the attack success rate against the number of queries for all baseline methods on classifier equipped with adversarial example detection mechanism. Again we can see that BO-DBA achieves the highest overall attack success rate and best query efficiency as compared with the other four hard-label attack baselines.

5.2 Effect of Perturbation Generator

Baselines: We explore the influence of different perturbation generators on attack efficiency and perturbation quality when combined with our framework. In general, we can divide the tested perturbation generators into three types: **procedural noises generators**Co et al. (2019), **interpolation-based function**Ru et al. (2019) and **clustering-based function** Shukla et al. (2019). Procedural noises use the random function to generate an image with complex and intricate details which are widely used in computer graphicsLagae et al. (2010). For procedural noises, we consider Perlin and Gabor noise. Interpolation-based functions are widely used in image rescaling and we consider bilinear (BILI) and bicubic (BICU) interpolation. For the clustering-based function, we consider nearest-neighbor (NN) and clustering (CL).

[2]Note that the relatively weak performance of SignOPT is due to the fact that SignOPT is designed for l_2 norm attack while this experiment is under the l_∞ norm setting.

Generator	UAR	ASR	LPIPS	l_2	L_∞
Perlin	26.9%	87%	0.087	9.28	0.035
Gabor	29.4%	77%	0.135	15.70	0.842
BILI	4.7%	27%	0.158	43.44	0.279
BICU	5.9%	26%	0.155	42.14	0.266
CL	9.0%	67%	0.259	18.92	0.862
NN	11.9%	55%	0.115	24.94	0.891

Table 1: Perturbation Generators Evaluation

Metrics: In addition to the same evaluation matrix used in Section 5.1, we also measure the inherent properties of perturbation generators like University evasion rate (UAR)Co et al. (2019) which refers to perturbation's ability to apply across a dataset or to other models. Given a model f, a perturbation s, input $x \in X$ and a threshold ϵ, the UAR of s over X is:

$$\frac{|\{x \in X : \arg\max f(x+s) \neq \tau(x)|}{|X|}, |s|_\infty \leq \epsilon$$

Where $\tau(x)$ is the true label of x and we set $\epsilon = \frac{16}{255}$.

Results: Table 1 compares the perturbation quality and inherent properties of different perturbation generators. We can see that perturbation generators that belong to the same category have similar inherent properties. For example, procedural noise generators have a higher UAR than other generators. In terms of distortion quality, we found each noise generator has a distinct distortion quality in different distortion measurements. For example, NN generator has relatively high $l_2 \& l_\infty$ distortion but low LPIPS, which means this type of noise usually has less perceptional significance than value significance.

6 Conclusion

In this paper we introduce a new decision-based attack BO-DBA which leverages Bayesian optimization to find adversarial perturbations with high query efficiency. With the optimized perturbation generation process, BO-DBA converges much faster than the state-of-the-art DBA techniques. As compared to existing decision-based attack methods, BO-DBA is able to converge within 200 queries while the state-of-the-art DBA techniques need over 15,000 queries to achieve the same level of perturbation distortion.

References

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pp. 1277–1294. IEEE, 2020.

Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1739–1747, 2020.

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 *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26, 2017.

Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018.

Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*, 2019.

Kenneth T Co, Luis Muñoz-González, Sixte de Maupeou, and Emil C Lupu. Procedural noise adversarial examples for black-box attacks on deep convolutional networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 275–289, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

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

Ares Lagae, Sylvain Lefebvre, Rob Cook, Tony DeRose, George Drettakis, David S Ebert, John P Lewis, Ken Perlin, and Matthias Zwicker. A survey of procedural noise functions. In *Computer Graphics Forum*, volume 29, pp. 2579–2600. Wiley Online Library, 2010.

Huichen Li, Linyi Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Nonlinear projection based gradient estimation for query efficient blackbox attacks, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Jonas Mockus. *Bayesian approach to global optimization: theory and applications*, volume 37. Springer Science & Business Media, 2012.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

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

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pp. 63–71. Springer, 2003.

Binxin Ru, Adam Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. In *International Conference on Learning Representations*, 2019.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Satya Narayan Shukla, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Black-box adversarial attacks with bayesian optimization. *arXiv preprint arXiv:1909.13857*, 2019.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959, 2012.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Jiawei Zhang, Linyi Li, Huichen Li, Xiaolu Zhang, Shuang Yang, and Bo Li. Progressive-scale boundary blackbox attack via projective gradient estimation. *arXiv preprint arXiv:2106.06056*, 2021.