CGBA: Curvature-aware Geometric Black-box Attack

Md Farhamdur Reza, Ali Rahmati, Tianfu Wu, and Huaiyu Dai

Department of ECE, North Carolina State University

mreza2@ncsu.edu, arahmat@alumni.ncsu.edu, tianfu_wu@ncsu.edu , hdai@ncsu.edu

Abstract

Decision-based black-box attacks often necessitate a large number of queries to craft an adversarial example. Moreover, decision-based attacks based on querying boundary points in the estimated normal vector direction often suffer from inefficiency and convergence issues. In this paper, we propose a novel query-efficient curvature-aware geometric decision-based black-box attack (CGBA) that conducts boundary search along a semicircular path on a restricted 2D plane to ensure finding a boundary point successfully irrespective of the boundary curvature. While the proposed CGBA attack can work effectively for an arbitrary decision boundary, it is particularly efficient in exploiting the low curvature to craft high-quality adversarial examples, which is widely seen and experimentally verified in commonly used classifiers under non-targeted attacks. In contrast, the decision boundaries often exhibit higher curvature under targeted attacks. Thus, we develop a new query-efficient variant, CGBA-H, that is adapted for the targeted attack. In addition, we further design an algorithm to obtain a better initial boundary point at the expense of some extra queries, which considerably enhances the performance of the targeted attack. Extensive experiments are conducted to evaluate the performance of our proposed methods against some well-known classifiers on the ImageNet and CIFAR10 datasets, demonstrating the superiority of CGBA and CGBA-H over state-of-the-art non-targeted and targeted attacks, respectively. The source code is available at https://github.com/Farhamdur/CGBA.

1. Introduction

Adversarial attacks are broadly classified into two types: white-box and black-box attacks. In the white-box attack setting [12, 23, 3], the adversary possesses the full knowledge of the target classifier and its weights. However, it is often impractical to avail information about the target classifier in real-world scenarios. Therefore, the black-box setting—transfer-based, score-based, and decision-based—is the practical setting for adversarial attacks with limited knowledge of the classifier. The transfer-based adversarial attacks [25, 29] use a surrogate model to generate adversarial examples though it does not guarantee a high attack success rate. Score-based attacks [6, 15] query the target classifier for the prediction probabilities of all the classes in order to estimate the gradient in each step and lessen perturbation. However, this attacking strategy may not be feasible because, in many real-world applications, classifiers only return the top-1 classification label in response to a query. Thus, the decision-based adversarial attack is the most practical adversarial attack as it allows the adversary to craft an adversarial example by only querying the top-1 classification label from the target classifier.

Most state-of-the-art decision-based attacks, such as HSJA [5], qFool [19], GeoDA [26], QEBA [17] and TA [20], are based on finding the normal vector at a point on the decision boundary and iteratively search for new boundary points with reduced perturbation. Among these attacks, HSJA [5], qFool (targeted) [19], QEBA [17], and TA [20] employ the estimated normal vector direction to obtain a point inside the adversarial region, and then apply binary search between the obtained adversarial point and the source to get a new boundary point. The aforementioned approaches, however, do not explicitly take into account the geometry of the boundary when coining adversarial examples. qFool (non-targeted) [19] and GeoDA [26], on the other hand, approximate the boundary as a hyperplane and find a new boundary point by conducting a binary search along the direction of the estimated normal vector (BSNV). As further discussed below, while BSNV is effective for low-curvature boundaries where the linear approximation is sufficiently accurate, its effectiveness deteriorates as curvature and nonlinearity increase. For a boundary with high curvature, BSNV may not even hit the boundary due to the narrow adversarial region. SurFree [22] also considers a hyperplane boundary and conducts boundary search along a semicircular path, but it does not utilize the information of the normal vector and instead estimates the attack direction through random trials.

A careful examination of the above normal vector based attacks reveals the following limitations. The estimation of the normal vector may be inaccurate due to the limited query budget and the non-linearity of the boundary. Thus, the expected reduction in perturbation may not occur when searching along the direction of the estimated normal vector. Moreover, if the adversarial region is narrow enough, the search process does not converge towards perturbation reduction due to the inability to find the adversarial region in the search direction. Fundamentally, these limitations are related to the one-dimensional (1-D) search nature dictated by the estimated normal vector. The state-of-the-art (SOTA) non-targeted attack SurFree, on the other hand, queries an adversarial point along a semicircular path but does not use the critical normal vector information to estimate the attack direction. Motivated by the above observation, we propose a new curvature-aware geometric black-box attack (CGBA) in this work to further improve the attack efficiency. Particularly, rather than conducting a boundary point search towards the estimated normal direction or along a semicircular path in some random direction, CGBA conducts the boundary point search along a semicircular path (BSSP) in a restricted 2-D plane spanned by two vectors: the direction towards a boundary point from the source (i.e., $\hat{\boldsymbol{v}}_t$ in Figure 1) and the estimated normal direction on that boundary point (i.e., $\hat{\boldsymbol{\eta}}_t$ in Figure 1). As further illustrated in Section 4 and Appendix D, the proposed CGBA overcomes the limitations of 1D boundary search and is a query-efficient approach for low curvature boundaries. However, it gradually loses query efficiency as the curvature of the boundary increases. Thus, we modify CGBA to CGBA-H which follows the same restricted 2D semicircular path but more swiftly adapts to the high curvature of the decision boundary. Our main contributions are summarized as follows:

- We propose CGBA, a novel iterative decision-based black-box attack that conducts boundary search along a semicircular path on a restricted 2-D plane and effectively overcomes the limitations of existing 1-D search based on estimated normal vectors at the decision boundary.
- The proposed CGBA attack can effectively exploit the decision boundary's low curvature for non-targeted attacks. When the decision boundary assumes a high curvature, we develop a new variant, CGBA-H, which achieves better performance for targeted attacks.
- Moreover, we introduce an algorithm to choose a better initial boundary point and demonstrate that this initialization method leads to significant performance improvement for the targeted attack.
- Experimental results on ImageNet and CIFAR10 reveal the efficacy of CGBA and CGBA-H for non-targeted and targeted attacks, respectively.

2. Related work

Decision-based black-box attack is the most challenging setting to obtain adversarial examples as the only information available to perform this type of attack is the target classifier's top-1 classification label. Some decision-based black-box attacks use a random search, while others are based on finding the gradient on the decision boundary. Boundary Attack [1] algorithm performs a random walk along the decision boundary to reduce the perturbation with query though it still incurs a large number of queries. To speed up the performance of [1], Biased Boundary Attack [2] proposes three priors to reduce the search space, and it is shown that the Perlin bias introduces the most favorable effect. OPT [7] and Rays [4] are decision-based attacks that randomly search for optimal directions to reduce the perturbation. However, Rays is only applicable for non-targeted attacks [20], and its performance is shown for ℓ_∞-norm. Sign-OPT [8] improves the query efficiency of OPT [7] by computing the sign of the directional derivatives to estimate the gradient. In [10], an evolutionary attack method is proposed in which random samples are drawn from a normal distribution with customized co-variance in reduced search space. AHA [18], on the other hand, utilizes the mean of the historical queries to generate random samples from a normal distribution. Triangle Attack (TriA) [28] is based on the geometric relationship between benign samples, current and future adversarial examples, forming a triangle in a subspace at each iteration. SurFree [22], a surrogate-free algorithm, claims that bypassing the query cost of normal vector estimation would improve query efficiency. However, we refute this claim by conducting the boundary search in a restricted 2D plane guided by the normal vector and achieving better performance. Moreover, SurFree only supports non-targeted attacks [20] as opposed to our methods addressing both non-targeted and targeted attacks.

Several existing attacks rely on estimating the normal vector on the boundary point. HSJA [5] proposes query-efficient methods by estimating the normal vector on the decision boundary to obtain a boundary point with reduced perturbation. qFool [19] and GeoDA [26] are based on the observation that the curvature of the decision boundary is small around adversarial examples. To improve the performance using the normal vector estimation, GeoDA [26], which is applicable for non-targeted attacks, proposes a method to distribute the query optimally to iterations given a query budget. QEBA [17] is built on top of HSJA [5] with a dimension-reduced subspace to generate queries for estimating the normal vector direction. QEBA proposes spatial, frequency, and intrinsic component subspaces to better estimate the normal vector. TA [20] demonstrates a new method for minimizing the ℓ_2-norm of perturbation by obtaining the tangent of a virtual hemisphere.

3. Problem definition

Let a pre-trained L-class classifier be modeled as $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^L$. For a given input image $\boldsymbol{x} \in [0,1]^n$, $f \in \mathbb{R}^L$ is the confidence score of the classifier. In a decision-based black-box attack, the classifier only returns the top-1 classification label of f. The output of the classifier f for a given query $\boldsymbol{x}$ in the decision-based attack can be expressed as $\hat{y}(\boldsymbol{x}) = \arg\max_j [f(\boldsymbol{x})]_j$, where $[f]_j$ is the prediction probability of j-th class, $1 \leq j \leq L$.

For a correctly classified source image $\boldsymbol{x}_s$ by the classifier f, the goal is to find a direction $\hat{\boldsymbol{\zeta}}$ so that $\boldsymbol{x}_s$ can be moved towards that direction to get an adversarial image with minimum perturbation. If a query $\boldsymbol{x}_q = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}})$ is in the adversarial region, the classifier f returns an incorrect prediction due to the added perturbation, where $\boldsymbol{d}(\hat{\boldsymbol{\zeta}})$ denotes the perturbation added to $\boldsymbol{x}_s$. The optimal direction to get an adversarial image can be formulated as:

$$\hat{\boldsymbol{\zeta}}^* = \arg\min_{\hat{\boldsymbol{\zeta}} \in \mathbb{R}^n} \|\boldsymbol{d}(\hat{\boldsymbol{\zeta}})\|_2, \quad \text{s.t. } \phi(\boldsymbol{x}_q) = 1, \tag{1}$$

where $\|\boldsymbol{d}(\hat{\boldsymbol{\zeta}})\|_2 = d$ is the ℓ_2-norm of perturbation added in the direction $\hat{\boldsymbol{\zeta}}$, and $\phi(.)$ denotes an indicator function to determine whether the query is correctly classified or misclassified. For a non-targeted attack:

$$\phi(\boldsymbol{x}_q) = \begin{cases} 1, & \text{if } \hat{y}(\boldsymbol{x}_q) \neq \hat{y}(\boldsymbol{x}_s) \\ -1, & \text{otherwise} \end{cases}, \tag{2}$$

and for a targeted attack with an intended classification label l_t:

$$\phi(\boldsymbol{x}_q) = \begin{cases} 1, & \text{if } \hat{y}(\boldsymbol{x}_q) = l_t \\ -1, & \text{otherwise} \end{cases}. \tag{3}$$

The optimal direction $\hat{\boldsymbol{\zeta}}^*$ results in minimum perturbation $\boldsymbol{d}(\hat{\boldsymbol{\zeta}}^*)$ to obtain the desired adversarial image $\boldsymbol{x}^*_{adv} = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}^*)$. This paper proposes novel methods for obtaining adversarial images for non-targeted and targeted attacks.

4. Proposed methods

Geometric-based attacks like qFool(non-targeted) [19], GeoDA [26], and SurFree [22] approximate the decision boundary as a hyperplane. However, SurFree doesn't use normal vector information, while qFool and GeoDA lose effectiveness with sufficiently curved boundaries. In contrast, CGBA conducts a boundary search along a semicircular path guided by the estimated normal vector, which works effectively for arbitrary decision boundaries, and demonstrates significant improvement on low to medium curvatures. Moreover, CGBA is modified to CGBA-H to further adapt to the high curvature setting. Our methods are iterative and estimate the boundary point's normal vector in each iteration, which is one key component accounting for its success.

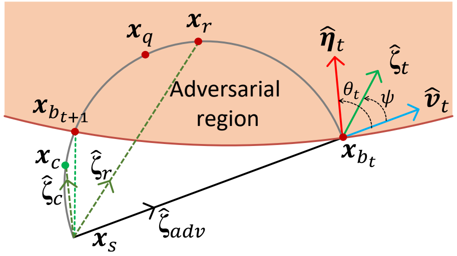


Figure 1: The geometry of CGBA.

Normal vector approximation on decision boundary. Let us assume $\boldsymbol{x}_{b_t}$ as a boundary point at the t-th iteration. To estimate the normal vector on the decision boundary, we generate N_t number of random samples $\{\boldsymbol{z}_i\}_{i=1}^{N_t}$ from a Gaussian distribution $\boldsymbol{z}_i \sim \mathcal{N}(0, \sigma^{\boldsymbol{2}})$. Then, for each of the samples, we query the classifier with $\boldsymbol{x}_{b_t} + \boldsymbol{z}_i$, $i \in \{1, ..., N_t\}$ to obtain the hard-label prediction of the classifier. Using these queries and their corresponding predictions, the normal unit vector on the boundary at t-th iteration can be approximated as [26]:

$$\hat{\boldsymbol{\eta}}_t = \frac{\sum_{i=1}^{N_t} \phi(\boldsymbol{x}_{b_t} + \boldsymbol{z}_i)\boldsymbol{z}_i}{\| \sum_{i=1}^{N_t} \phi(\boldsymbol{x}_{b_t} + \boldsymbol{z}_i)\boldsymbol{z}_i\|_2}. \tag{4}$$

4.1. CGBA

Denote $\hat{\boldsymbol{v}}_t = (\boldsymbol{x}_{b_t} - \boldsymbol{x}_s)/\|\boldsymbol{x}_{b_t} - \boldsymbol{x}_s\|_2$ as the direction of a boundary point $\boldsymbol{x}_{b_t}$ from a source $\boldsymbol{x}_s$, and $\hat{\boldsymbol{\eta}}_t$ is the estimated normal direction on $\boldsymbol{x}_{b_t}$ at t-th iteration. The key idea of this method is to conduct a boundary search to obtain a better subsequent boundary point $\boldsymbol{x}_{b_{t+1}}$ on a semicircular path in a 2-D plane spanned by $(\hat{\boldsymbol{\eta}}_t, \hat{\boldsymbol{v}}_t)$ on the side of $\hat{\boldsymbol{\eta}}_t$, where the semicircular path is formed between $\boldsymbol{x}_s$ and $\boldsymbol{x}_{b_t}$ centered at $(\boldsymbol{x}_{b_t} + \boldsymbol{x}_s)/2$, as shown in Figure 1. The search direction to perform a query in the plane spanned by $(\hat{\boldsymbol{\eta}}_t, \hat{\boldsymbol{v}}_t)$ can be obtained as:

$$\hat{\boldsymbol{\zeta}}_t(m) = \frac{\hat{\boldsymbol{\eta}}_t + m\hat{\boldsymbol{v}}_t}{\|\hat{\boldsymbol{\eta}}_t + m\hat{\boldsymbol{v}}_t\|_2}, \tag{5}$$

where m is a multiplication factor that controls the search direction. For a search direction $\hat{\boldsymbol{\zeta}}_t$, we can calculate the perturbation $\boldsymbol{d}(\hat{\boldsymbol{\zeta}}_t)$ to obtain the query on the semicircular path as:

$$\boldsymbol{d}(\hat{\boldsymbol{\zeta}}_t) = \|\boldsymbol{x}_{b_t} - \boldsymbol{x}_s\|_2 (\hat{\boldsymbol{\zeta}}_t \cdot \hat{\boldsymbol{v}}_t)\hat{\boldsymbol{\zeta}}_t, \tag{6}$$

where $\|\boldsymbol{x}_{b_t} - \boldsymbol{x}_s\|_2 (\hat{\boldsymbol{\zeta}}_t \cdot \hat{\boldsymbol{v}}_t) = \|\boldsymbol{x}_{b_t} - \boldsymbol{x}_s\|_2 \cos\psi$ denotes the added ℓ_2-norm of perturbation in the direction $\hat{\boldsymbol{\zeta}}_t$ with ψ as the angular difference between $\hat{\boldsymbol{\zeta}}_t$ and $\hat{\boldsymbol{v}}_t$. By varying $\hat{\boldsymbol{\zeta}}_t$, we query the target model for $\boldsymbol{x}_q = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_t)$ to conduct boundary search on the semicircular path.

The proposed CGBA first finds a non-adversarial point and an adversarial point on the semicircle, and then progressively reduces the range between the adversarial and

Algorithm 1: CGBA

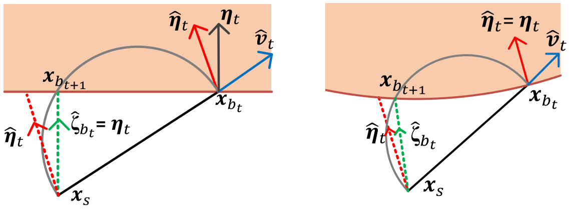
```
Inputs: Source image $\boldsymbol{x}_s$, indicator function $\phi(.)$, a
  random direction $\boldsymbol{\Theta}$, queries to estimate initial normal
  vector $N_0$, iteration $T$.
Output: Adversarial example $\boldsymbol{x}_{adv}$.
$r \leftarrow \min\{r > 0 : \phi(\boldsymbol{x}_s + r * \frac{\boldsymbol{\Theta}}{\|\boldsymbol{\Theta}\|_2}) = 1\}$
$\boldsymbol{x}_{b_1} = \boldsymbol{x}_s + r * \frac{\boldsymbol{\Theta}_1}{\|\boldsymbol{\Theta}_1\|_2}$
for $t = 1 : T$ do
    Generate $N_t = N_0\sqrt{t}$ samples, $\boldsymbol{z_i} \sim \mathcal{N}(0, \sigma^{\boldsymbol{2}})$
    Estimate $\hat{\boldsymbol{\eta}}_t$ using $\boldsymbol{z_i}$ at $\boldsymbol{x}_{b_t}$ by $N_t$ queries.
    $\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{x}_{b_t} - \boldsymbol{x}_s}{\|\boldsymbol{x}_{b_t} - \boldsymbol{x}_s\|_2}$,  $\theta_t = \cos^{-1}(\hat{\boldsymbol{\eta}}_t \cdot \hat{\boldsymbol{v}}_t)$,  $i = 1$
    while True do
        $m_i = \sin\theta_t \cot(90^0 - \frac{90^0}{2^i}) - \cos\theta_t$
        $\hat{\boldsymbol{\zeta}}_t = (\hat{\boldsymbol{\eta}}_t + m_i\hat{\boldsymbol{v}}_t)/\|\hat{\boldsymbol{\eta}}_t + m_i\hat{\boldsymbol{v}}_t\|_2$
        $\boldsymbol{x}_q = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_t)$,   $i = i + 1$
        if $\phi(\boldsymbol{x}_q) = -1$ then
            break

    $\boldsymbol{x}_{b_{t+1}} \leftarrow BSSP(\boldsymbol{x}_s, \boldsymbol{x}_q, \boldsymbol{x}_{b_t}, \phi)$      /* to find
     the boundary point on semicircular
     path */
$\boldsymbol{x}_{adv} = \boldsymbol{x}_{b_{t+1}}$
```

non-adversarial points to obtain the boundary point $\boldsymbol{x}_{b_{t+1}}$, inspired by the binary search. If ψ_i is the search angle of $\hat{\boldsymbol{\zeta}}_t$ w.r.t. $\hat{\boldsymbol{v}}_t$, the multiplication factor m_i to to attain the search angle ψ_i can be calculated as:

$$m_i = \sin\theta_t \cot\psi_i - \cos\theta_t; \quad \forall i \in \mathbb{Z}^+, \tag{7}$$

where $\theta_t = \cos^{-1}(\hat{\boldsymbol{v}}_t \cdot \hat{\boldsymbol{\eta}}_t)$ and $\psi_i = (90^0 - \frac{90^0}{2^i}), \forall i \in \mathbb{Z}^+$. With the increase of i, the search angle ψ_i is also increased. Thus, for a particular value of m_i, we obtain a perturbation $\boldsymbol{d}(\hat{\boldsymbol{\zeta}}_t(m_i))$ and the corresponding point $\boldsymbol{x}_c = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_t(m_i))$ on the semicircle such that $\phi(\boldsymbol{x}_c) = -1$. Then the boundary search between the non-adversarial point $\boldsymbol{x}_c$ and the adversarial point $\boldsymbol{x}_{b_t}$ is conducted by using BSSP to find $x_{b_{t+1}}$ along a semicircular path. At the start of the BSSP at t-th iteration, consider $\hat{\boldsymbol{\zeta}}_{adv}$ and $\hat{\boldsymbol{\zeta}}_c$ are the directions of $\boldsymbol{x}_{b_t}$ and $\boldsymbol{x}_c$ from $\boldsymbol{x}_s$, respectively, and $\hat{\boldsymbol{\zeta}}_r = (\hat{\boldsymbol{\zeta}}_{adv} + \hat{\boldsymbol{\zeta}}_c)/\|\hat{\boldsymbol{\zeta}}_{adv} + \hat{\boldsymbol{\zeta}}_c\|_2$ is the resultant direction of $\hat{\boldsymbol{\zeta}}_{adv}$ and $\hat{\boldsymbol{\zeta}}_c$, as shown in Figure 1. The BSSP reduces the range of search direction from $[\hat{\boldsymbol{\zeta}}_{adv}, \hat{\boldsymbol{\zeta}}_c]$ to $[\hat{\boldsymbol{\zeta}}_r, \hat{\boldsymbol{\zeta}}_c]$ for $\phi(\boldsymbol{x}_r) = 1$ as $\boldsymbol{x}_{b_{t+1}}$ lies between the directions $\hat{\boldsymbol{\zeta}}_r$ and $\hat{\boldsymbol{\zeta}}_c$, while the range will be reduced to $[\hat{\boldsymbol{\zeta}}_{adv}, \hat{\boldsymbol{\zeta}}_r]$ for $\phi(\boldsymbol{x}_r) = -1$, where $\boldsymbol{x}_r = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_r)$. This process of reducing the range of the search direction is continued until obtaining the boundary point $\boldsymbol{x}_{b_{t+1}}$ with a certain accuracy. One important characteristic of BSSP is that it ensures $\boldsymbol{x}_{b_{t+1}}$ with a reduced perturbation since for any query $\boldsymbol{x}_q$ on the semicircular path, $\|\boldsymbol{x}_q - \boldsymbol{x}_s\|_2 \leq \|\boldsymbol{x}_{b_t} - \boldsymbol{x}_s\|_2$.

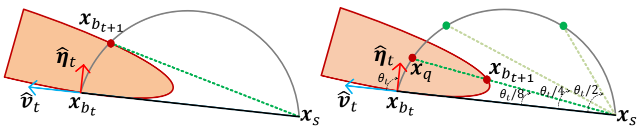
To demonstrate the efficacy of the BSSP algorithm, we consider two scenarios of boundary: linear and low curvature boundaries. Let us consider $\hat{\boldsymbol{\zeta}}_{b_t}$ as the direction of $\boldsymbol{x}_{b_{t+1}}$ obtained by using the BSSP method. Thus, $\boldsymbol{x}_{b_{t+1}}$ can be calculated as: $\boldsymbol{x}_{b_{t+1}} = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_{b_t})$. First of all, in the case of a linear boundary, the direction $\hat{\boldsymbol{\zeta}}_{b_t}$ makes a right angle with the boundary as any inscribed angle in a semicircle makes a right angle, as shown in Figure 2a. If $\boldsymbol{\eta}_t$ represents the true normal vector on the boundary, then the direction of $\boldsymbol{\eta}_t$ coincides with $\hat{\boldsymbol{\zeta}}_{b_t}$. Thus, it should be enough to push $\boldsymbol{x}_s$ towards $\boldsymbol{\eta}_t$ using the BSNV method to get the optimal perturbation as it is done in GeoDA[26] and qFool (non-targeted)[19]. However, $\boldsymbol{\eta}_t$ is an unknown parameter that can be approximated using the normal vector estimation process. There is a high chance that the estimated normal vector direction $\hat{\boldsymbol{\eta}}_t$ deviates from $\boldsymbol{\eta}_t$. From Figure 2a, if there is a deviation between $\boldsymbol{\eta}_t$ and $\hat{\boldsymbol{\eta}}_t$, pushing $\boldsymbol{x}_s$ towards $\hat{\boldsymbol{\eta}}_t$ will not result in the optimal $\boldsymbol{x}_{b_{t+1}}$. In contrast, querying on the semicircular path in the plane spanned by $(\hat{\boldsymbol{v}}_t, \hat{\boldsymbol{\eta}}_t)$ finds optimal $\boldsymbol{x}_{b_{t+1}}$ in that plane. Secondly, if we consider a low curvature decision boundary, as shown in Figure 2b, BSSP finds a boundary point with smaller perturbation than using binary search towards $\hat{\boldsymbol{\eta}}_t$ even $\hat{\boldsymbol{\eta}}_t$ is same as the $\boldsymbol{\eta}_t$.


$\hat{\boldsymbol{\eta}}_t$
$\boldsymbol{\eta}_t$
$\hat{\boldsymbol{v}}_t$
$\boldsymbol{x}_{b_{t+1}}$
$\boldsymbol{x}_{b_t}$
$\hat{\boldsymbol{\eta}}_t$
$\hat{\boldsymbol{\zeta}}_{b_t} = \boldsymbol{\eta}_t$
$\boldsymbol{x}_s$
$\hat{\boldsymbol{\eta}}_t = \boldsymbol{\eta}_t$
$\hat{\boldsymbol{v}}_t$
$\boldsymbol{x}_{b_{t+1}}$
$\boldsymbol{x}_{b_t}$
$\hat{\boldsymbol{\eta}}_t$
$\hat{\boldsymbol{\zeta}}_{b_t}$
$\boldsymbol{x}_s$

(a) (b)

Figure 2: (a) Linear and (b) low curvature boundaries.

Considering the above scenarios, the BSSP finds a better boundary point than the BSNV, which in turn makes the proposed CGBA effective. Experimental and theoretical evidence supporting that BSSP is more effective than BSNV are provided in Appendix D. The pseudocode of CGBA for the non-targeted attack is given in Algorithm 1 which can be easily converted to the targeted attack. The pseudocode of the BSSP algorithm is given in Appendix A.



(a) (b)

Figure 3: Boundary with high curvature.

4.2. CGBA-H

Adversarial attacks using CGBA can be an effective approach for the decision boundary with low curvature. However, this approach becomes less effective if the curvature of the boundary is too high. From Figure 3a, it can be realized that obtaining the boundary point $\boldsymbol{x}_{b_{t+1}}$ using the BSSP may result in a boundary point that is away from the optimal solution. To avoid this situation, we propose a more effective approach to get a better boundary point in each iteration for the boundary with high curvature. If $\theta_t = \cos^{-1}(\hat{\boldsymbol{\eta}}_t \cdot \hat{\boldsymbol{v}}_t)$ is the angle between $\hat{\boldsymbol{\eta}}_t$ and $\hat{\boldsymbol{v}}_t$, then the multiplying factor to estimate the direction to query can be calculated as:

$$m_i = \sin\theta_t \cot\left\{\frac{\theta_t}{2^i}\right\} - \cos\theta_t; \quad \forall i \in \mathbb{Z}^+, \tag{8}$$

where the value of i ensures $\cos^{-1}(\hat{\boldsymbol{v}}_t \cdot \hat{\boldsymbol{\zeta}}_t(m_i)) = \theta_t/2^i;\ \forall i \in \mathbb{Z}^+$. So, with the increase of i, the angular difference between $\hat{\boldsymbol{v}}_t$ and estimated $\hat{\boldsymbol{\zeta}}_t(m_i)$ can be reduced. Thus, in each iteration, the proposed CGBA-H finds a multiplication factor m_i and the corresponding $\boldsymbol{x}_q = \boldsymbol{x}_s + \boldsymbol{d}\hat{\boldsymbol{\zeta}}_t(m_i)$ on the semicircular trajectory such that $\phi(\boldsymbol{x}_q) = 1$, as shown in Figure 3b. Then, by conducting the binary search between $\boldsymbol{x}_s$ and $\boldsymbol{x}_q$, a better boundary point $\boldsymbol{x}_{b_{t+1}}$ can be obtained than CGBA. The pseudocode of CGBA-H for the targeted attack is given in Algorithm 2.

Algorithm 2: CGBA-H

```
Inputs: Source image $\boldsymbol{x}_s$, a random image $\boldsymbol{x}_t$ of target
   class $l_t$, indicator function $\phi(.)$, queries to find initial
   normal vector $N_0$, iteration $T$.
Output: Adversarial example $\boldsymbol{x}_{adv}$.
$\boldsymbol{x}_{b_1} \leftarrow BinarySearch(\boldsymbol{x}_s, \boldsymbol{x}_t, \phi)$      /* to find
   initial boundary point */
for $t = 1 : T$ do
    Generate $N_t = N_0\sqrt{t}$ samples, $\mathbf{z_i} \sim \mathcal{N}(0, \boldsymbol{\sigma^2})$
    Estimate $\hat{\boldsymbol{\eta}}_t$ using $\mathbf{z_i}$ at $\boldsymbol{x}_{b_t}$ by $N_t$ queries.
    $\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{x}_{b_t}-\boldsymbol{x}_s}{\|\boldsymbol{x}_{b_t}-\boldsymbol{x}_s\|_2},\quad \theta_t = \cos^{-1}(\hat{\boldsymbol{\eta}}_t \cdot \hat{\boldsymbol{v}}_t),\quad i = 1$
    while True do
        $m_i = \sin\theta_t \cot\left(\frac{\theta_t}{2^i}\right) - \cos\theta_t$
        $\hat{\boldsymbol{\zeta}}_t = (\hat{\boldsymbol{\eta}}_t + m_i\hat{\boldsymbol{v}}_t)/\|\hat{\boldsymbol{\eta}}_t + m_i\hat{\boldsymbol{v}}_t\|_2$
        $\boldsymbol{x}_q = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_t),\quad i = i + 1$
        if $\phi(\boldsymbol{x}_q) = 1$ then
            break
    $\boldsymbol{x}_{b_{t+1}} \leftarrow BinarySearch(\boldsymbol{x}_s, \boldsymbol{x}_q, \phi)$      /* to
       find boundary point */
$\boldsymbol{x}_{adv} = \boldsymbol{x}_{b_{t+1}}$
```

4.3. Initialization

The initial boundary point $\boldsymbol{x}_{b_1}$ may have a significant impact on the performance of an adversarial attack. In the existing normal vector-based targeted attack, a binary search in the direction of a randomly chosen image of the target class from the source image $\boldsymbol{x}_s$ is conducted to obtain initial boundary point $\boldsymbol{x}_{b_1}$. Rather than finding $\boldsymbol{x}_{b_1}$ in the direction of a randomly chosen target sample form $\boldsymbol{x}_s$, a set of K random directions $\{\boldsymbol{\Theta}_k\}_{k=1}^{K}$ towards the adversarial region by using K-number of samples of the target class can be used to find the direction that provides a boundary point with reduced perturbation for the targeted attack. Experimental results reveal that just using a few samples of the target class to obtain $\boldsymbol{x}_{b_1}$ can significantly improve the performance of a decision-based adversarial attack. The pseudocode of the Initialization method to obtain the first boundary point is given in Appendix A.

5. Experiments

In this section, we perform a comprehensive set of experiments and compare the results with state-of-the-art algorithms to demonstrate the effectiveness of our proposed methods for non-targeted and targeted attacks. Moreover, we show how initialization affects the performance of CGBA and CGBA-H.

5.1. Experimental setting

Datasets and target models. We evaluate the performance of CGBA and CGBA-H using ImageNet [9] and CIFAR-10 [16] datasets. The performance of the proposed attacks on the ImageNet dataset is evaluated using pretrained ResNet50 [13], VGG16 [27], ResNet101 [13] and ViT [11] classifiers. The first three pretrained classifiers can be found in the PyTorch, and ViT is obtained from the PyTorch Image Models[1]. For each target model, we randomly select 1000 images for the non-targeted attack and 1000 pairs of images for the targeted attack from the ILSVRC2012's validation set [9] that are correctly classified by the target model. The images are resized to 3 $\times 224 \times 224$ as an input to the classifiers. For the CIFAR-10 dataset, we consider PreActResNet-18 [14] and a wide residual network with 40 layers (WRN40) [30] as target classifiers. We train both classifiers for 200 epochs with an image resolution of $3 \times 32 \times 32$. The proposed attacks on CIFAR10 are also evaluated using a randomly chosen 1000 correctly classified images for the non-targeted attack and 1000 pairs of correctly classified images for the targeted attack.

Baselines and hyper-parameter setting. We compare the performance of CGBA and CGBA-H with the existing state-of-the-art non-targeted and targeted attacks. We choose HSJA [5], GeoDA [26], generalized TA [20], TriA [28], SurFree [22] and AHA [18] as baselines to compare. Among the baselines, HSJA and TA are available for both non-targeted and targeted attacks. However, GeoDA, TriA and SurFree are only given for the non-targeted at-

[1] https://github.com/rwightman/pytorch-image-models

	Attack	Non-targeted							Targeted						
	Queries	1000	2500	5000	7500	10000	15000	20000	1000	2500	5000	7500	10000	15000	20000
ResNet50	HSJA [5]	13.42	6.46	3.76	2.93	2.49	2.04	1.79	64.27	51.54	34.58	24.51	17.68	11.15	7.99
	GeoDA [26]	8.41	4.72	3.54	2.93	2.71	2.39	2.20	-	-	-	-	-	-	-
	TA [20]	13.98	6.36	3.77	2.97	2.46	1.97	1.70	63.09	46.55	31.94	23.05	16.95	10.91	7.87
	TriA [28]	6.26	5.58	5.39	5.15	5.03	4.84	4.73	-	-	-	-	-	-	-
	SurFree [22]	8.44	4.42	2.65	1.96	1.58	1.17	0.97	-	-	-	-	-	-	-
	AHA [18]	-	-	-	-	-	-	-	56.55	37.91	23.04	15.48	11.46	8.76	8.23
	CGBA	6.03	**2.55**	**1.44**	**1.05**	**0.86**	**0.68**	**0.59**	78.99	63.60	41.71	26.04	17.26	8.72	5.38
	CGBA-H	**5.78**	2.67	1.51	1.11	0.91	0.73	0.62	**56.01**	**36.86**	**21.83**	**13.71**	**9.47**	**5.63**	**4.03**
VGG16	HSJA[5]	8.58	4.11	2.54	2.06	1.75	1.44	1.29	65.10	47.31	30.61	21.72	15.58	9.72	7.13
	GeoDA [26]	5.74	3.41	2.49	2.11	1.98	1.67	1.63	-	-	-	-	-	-	-
	TA [20]	8.44	4.09	2.57	2.07	1.77	1.45	1.30	61.97	44.42	28.62	19.41	15.03	9.70	7.13
	TriA [28]	7.42	5.54	4.95	4.63	4.41	4.30	4.20	-	-	-	-	-	-	-
	SurFree [22]	6.01	3.18	1.96	1.52	1.24	0.97	0.82	-	-	-	-	-	-	-
	AHA [18]	-	-	-	-	-	-	-	55.40	36.46	21.37	14.26	10.91	8.73	8.34
	CGBA	3.99	**1.86**	**1.08**	**0.82**	**0.69**	**0.57**	**0.50**	80.13	67.09	44.93	27.67	16.33	7.69	4.91
	CGBA-H	**3.93**	1.94	1.17	0.89	0.75	0.61	0.54	**52.82**	**33.29**	**18.09**	**11.22**	**7.79**	**4.92**	**3.67**
ResNet101	HSJA [5]	16.12	7.59	4.17	3.26	2.66	2.07	1.77	68.80	55.78	38.48	28.24	20.68	12.99	9.45
	GeoDA [26]	8.85	4.99	3.83	3.04	2.79	2.38	2.22	-	-	-	-	-	-	-
	TA [20]	16.75	7.95	4.34	3.13	2.64	2.01	1.80	62.59	46.97	33.06	23.62	18.31	12.11	8.96
	TriA [28]	7.83	6.35	5.89	5.56	5.23	5.03	4.87	-	-	-	-	-	-	-
	SurFree [22]	10.47	5.62	3.12	2.16	1.79	1.35	1.11	-	-	-	-	-	-	-
	AHA [18]	-	-	-	-	-	-	-	56.47	39.67	23.78	16.02	12.61	9.52	8.94
	CGBA	7.89	3.38	1.84	**1.25**	**1.02**	**0.77**	**0.66**	73.17	60.38	39.85	25.47	17.48	9.26	6.13
	CGBA-H	**7.19**	**3.32**	**1.79**	1.26	1.03	0.78	0.67	**55.69**	**37.28**	**21.59**	**14.32**	**10.77**	**6.55**	**4.52**
ViT	HSJA[5]	26.41	10.33	5.87	4.61	3.86	3.16	2.74	61.84	42.54	27.07	19.39	15.05	10.71	8.34
	GeoDA [26]	15.39	8.05	5.84	4.73	4.25	3.66	3.38	-	-	-	-	-	-	-
	TA [20]	28.14	10.85	6.30	4.69	4.00	3.25	2.82	49.82	34.77	23.04	17.48	14.01	10.60	8.55
	TriA [28]	8.86	7.06	6.24	6.04	6.04	5.85	5.65	-	-	-	-	-	-	-
	SurFree [22]	14.96	6.90	4.11	3.10	2.53	1.95	1.61	-	-	-	-	-	-	-
	AHA [18]	-	-	-	-	-	-	-	43.06	28.29	17.54	12.59	9.58	6.77	5.74
	CGBA	10.62	3.53	**1.83**	**1.32**	**1.10**	**0.89**	**0.78**	60.03	42.35	24.46	14.90	10.06	6.36	5.48
	CGBA-H	**8.35**	**3.41**	1.86	1.38	1.15	0.92	0.83	**42.52**	**27.26**	**16.15**	**11.42**	**8.84**	**6.14**	**4.83**

Table 1: Median ℓ_2-norm of perturbation for different query budgets against ResNet50, VGG16, ResNet101 and ViT on ImageNet dataset.

tack [20], while AHA is available for the targeted attack. We consider GeoDA, SurFree and AHA for dimension-reduced subspace as these algorithms are given for this setting. For an image with a dimension of $3 \times 224 \times 224$, the reduced dimension by a factor f is given as $3 \times \frac{224}{f} \times \frac{224}{f}$. GeoDA and SurFree use dimension-reduced frequency subspace by reducing the dimension with a factor $f = 5.17$ and $f = 2$ to obtain coefficients of DCT transform, respectively, as their default setting. In contrast, AHA reduces the dimension in the spatial subspace by a factor $f = 4$ as their best setting. For our proposed attacks, we reduce the dimension by $f = 4$ in frequency subspace. We also set queries to estimate the initial normal vector as $N_0 = 30$ and the standard deviation for generating random samples from the Gaussian distribution as $\sigma = 0.0002$ to estimate the normal vector.

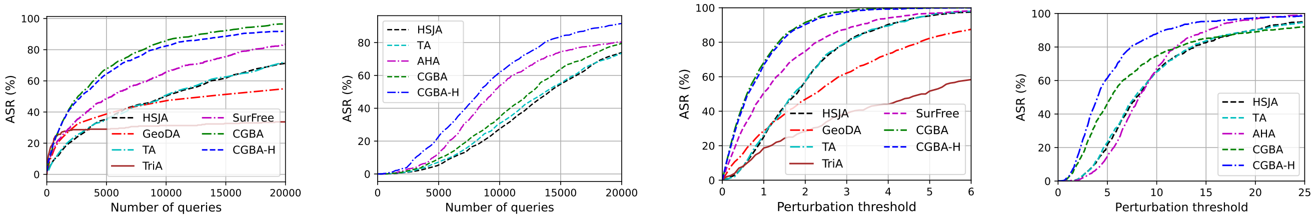
We use three metrics—median ℓ_2-norm of perturbation, attack success rate (ASR), and area under the curve (AUC)—to evaluate the performance of CGBA and CGBA-H with SOTA black-box attacks. The median of the ℓ_2-norm of perturbation for a given query budget using an attack determines the effectiveness of the attack. An attack with better capability to reduce the ℓ_2-norm of perturbation on a set of test images is deemed as a more effective attack. In addition, another popular metric, ASR, is used to determine the success rate of an adversarial attack for a given query budget and perturbation threshold. An attack is considered successful if the obtained perturbation for a particular query budget falls below the perturbation threshold. Moreover, AUC—the area under the curve of the median ℓ_2-norm of perturbation versus queries—demonstrates the convergence toward minimum perturbation of an attack with the number of queries. The lower the value of an attack's AUC, the faster the attack converges to the minimum perturbation.

5.2. Experimental results

Table 1 presents the median ℓ_2-norm of perturbation for different query budgets obtained by various baselines and our proposed algorithms for both non-targeted and targeted attacks, evaluated against ResNet50, VGG16, ResNet101 and ViT models using the ImageNet dataset. For further information, the corresponding curves for all classifiers can be found in Appendix B. Additionally, Appendix C contains the experimental results on CIFAR10.

	Methods	HSJA [5]	GeoDA [26]	TA [20]	TriA [28]	SurFree [22]	AHA [18]	CGBA	CGBA-H
ResNet50	Non-targeted	86531	129362	86756	107080	54575	-	37195	**36460**
	Targeted	520616	-	486946	-	-	383985	558611	**341965**
VGG16	Non-targeted	59049	54545	58987	98051	41270	-	**27275**	27604
	Targeted	471494	-	451480	-	-	370967	566433	**304521**
ResNet101	Non-targeted	96710	79915	97221	115203	66382	-	46229	**43250**
	Targeted	568382	-	504363	-	-	399841	541873	**353937**
ViT	Non-targeted	152219	130957	156360	129956	94329	-	63148	**55372**
	Targeted	447887	-	394218	-	-	298171	368239	**283216**

Table 2: AUC comparison against ResNet50, VGG16, ResNet101 and ViT for a query budget of 20000 on ImageNet.

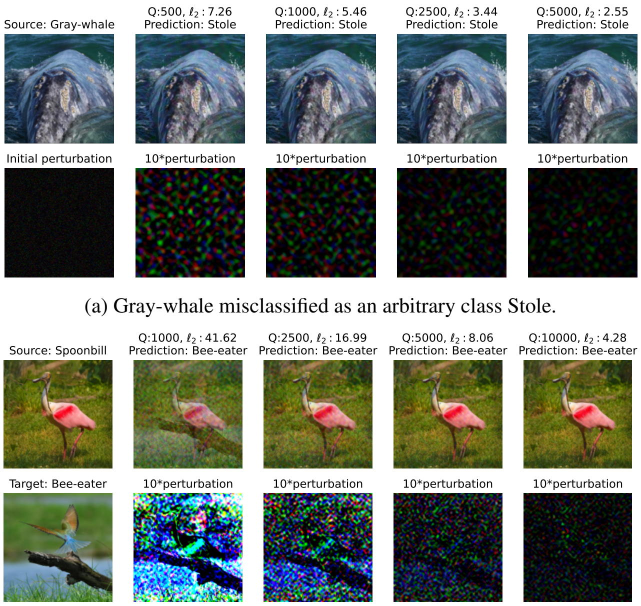


(a) Non-targeted attack (b) Targeted attack (c) Non-targeted attack (d) Targeted attack

Figure 4: ASR versus queries (a-b), and ASR versus perturbation thresholds (c-d) against ResNet50 on ImageNet.

In the case of non-targeted attacks, we observe that CGBA and CGBA-H outperform all the baselines. In most cases, with a sufficient query budget, CGBA achieves the best performance. When the query budget is limited, CGBA-H offers better performance; intuitively a lower-quality boundary point is obtained with a limited budget and the boundary appears more curved from the viewpoint of the source image. For targeted attacks, CGBA-H achieves the best performance across the board, and the gap with the baselines increases with the increase of query budget. It is interesting to note that even CGBA outperforms the baselines of targeted attacks with a sufficiently large query budget; intuitively the boundary appears much flatter from the viewpoint of the source image in this case with high-quality boundary points. The above observations conform to our insights and verify the effectiveness of our proposed methods.

Figure 4 demonstrates the ASR comparison of the proposed methods with the baselines against ResNet50, and Appendix B contains corresponding curves for the other classifiers on ImageNet. The left two sub-figures show the impact of queries on ASR. To plot these figures, we consider a perturbation threshold of 2.5 for non-targeted attacks and 12 for targeted attacks. The right two sub-figures, on the other hand, show the variation of ASR with different threshold values for a query budget of 20,000. These figures further demonstrate the superiority of CGBA and CGBA-H for non-targeted attacks and targeted attacks, respectively. Similar observations can be made from Table 2 on AUC comparison. Furthermore, we obtained perturbed images by using non-targeted CGBA and targeted CGBA-H for different query budgets, as shown in Figure 5a and 5b. In Figure 5, Q denotes a query budget, and ℓ_2 denotes the amount of perturbation corresponding to that query. We depict amplified obtained perturbation to observe how the perturbation diminishes with the increase of query budgets starting from an arbitrary random noise for the non-targeted attack and starting from a target image for the targeted attack.

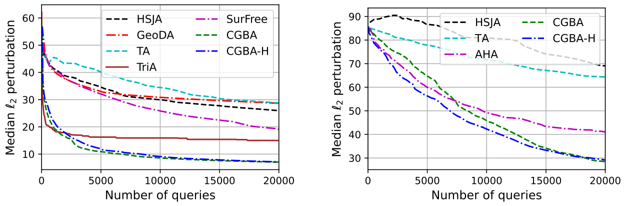


(a) Gray-whale misclassified as an arbitrary class Stole.

(b) Spoonbill misclassified as target class Bee-eater.

Figure 5: Adversarial examples for different query budgets.

Performance against adversarially-trained model. One of the most popular defense methods against adversarial at-



(a) Non-targeted attack (b) Targeted attack

Figure 6: Results against an adversarially-trained model.

		Non-targeted for different queries					Targeted for different queries			
f		1000	2500	5000	10000		1000	2500	5000	10000
4/3	SurFree	8.93	4.79	2.57	1.62	AHA	58.58	41.73	28.08	14.56
2		**8.44**	**4.42**	**2.65**	**1.58**		57.20	39.44	26.41	13.63
4		9.88	5.26	3.02	1.71		55.85	**37.36**	**23.14**	**11.44**
8		9.29	4.86	2.85	1.73		**53.59**	37.61	24.18	14.48
12		10.42	4.72	2.89	1.71		54.82	39.71	26.97	15.87
4/3	CGBA	10.49	4.09	2.07	1.19	CGBA-H	57.85	39.82	25.43	11.17
2		7.02	3.06	1.59	0.96		57.41	40.36	20.92	10.07
4		5.29	2.42	1.49	**0.88**		**55.32**	37.01	**21.36**	**9.57**
8		**4.72**	**2.35**	**1.43**	1.05		55.62	**36.77**	21.79	10.08
12		4.81	2.39	1.65	1.41		55.54	39.63	26.61	15.94

Table 3: Impact of dimension reduction on the performance.

tacks is adversarial training [21]. To evaluate the performance of the proposed attacks against the adversarially-trained model, we used a pre-trained ResNet50 model from the GitHub repository of MardyLab[2]. We randomly choose 200 samples for the non-targeted attack and 200 pairs of samples for the targeted attack. Figure 6a shows that our proposed methods perform much better than the SOTA baselines. One plausible explanation is that adversarial training makes the boundary flatter [24], and CGBA, guided by the normal vector, makes the best use of the flatness of the boundary. From Figure 6b, it is observed that the proposed methods are also effective in performing targeted attacks against the adversarially-trained model.

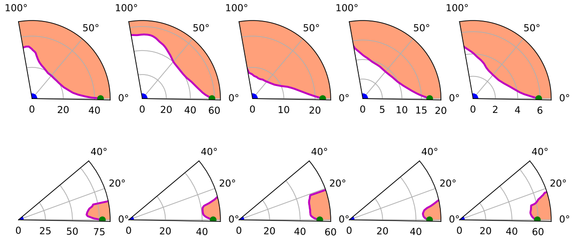
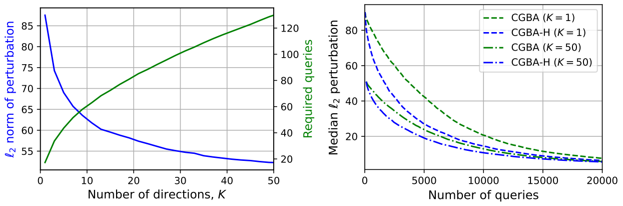


Figure 7: Boundary trajectory for non-targeted (top-row) and targeted (bottom-row) attacks.

Boundary trajectory. We demonstrate the difference in boundary trajectory between non-targeted and targeted attacks. We pick a set of five images and another set of five-pair of images randomly for non-targeted and targeted attacks, respectively. As the proposed methods are based on finding the normal vector on the decision boundary, on observing the boundary trajectory for a single iteration, we consider 400 queries to estimate $\hat{\boldsymbol{\eta}}_1$. In Figure 7, the blue dotted point at the center and green dotted point in each of the sub-figures indicate the source $\boldsymbol{x}_s$ and the initial boundary point $\boldsymbol{x}_{b_1}$, respectively. For a particular image, the direction of the green point from the blue point is denoted by $\hat{\boldsymbol{v}}_1$, and we consider this direction as a reference direction with 0-degree. After obtaining $\hat{\boldsymbol{\eta}}_1$ and $\hat{\boldsymbol{v}}_1$, we conduct a search for boundary points by gradually increasing the search direction towards $\hat{\boldsymbol{\eta}}_1$ in the plane spanned by $(\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{\eta}}_1)$. Figure 7 displays the boundary in the 2-D plane using a reddish curved line in which the shaded region indicates the adversarial region. We notice that the non-targeted attack has a low curvature boundary, as opposed to the targeted attack, which has a high curvature boundary and a narrow adversarial region. Because of this difference in the boundary trajectory, CGBA outperforms CGBA-H for non-targeted attacks while CGBA-H outperforms CGBA for targeted attacks.

Impact of dimension reduction. In Table 3, we compare the performance of CGBA and SOTA SurFree for the non-targeted attack, and the performance of CGBA-H and SOTA AHA for the targeted attack for different dimension reduction factor f. We randomly picked 100 images to perform the comparison of both attacks. For the non-targeted attack, SurFree offers the best performance for $f = 2$. However, with the further increase of f, it does not show any performance improvement in dimension-reduced frequency subspace. In contrast, CGBA offers the best performance for $f = 8$ with smaller query budgets and for $f = 4$ with larger ones. In all cases, CGBA significantly outperforms SurFree in dimension-reduced frequency subspace. For the targeted attack, it is observed that the performance of AHA and CGBA-H is comparable for a limited query budget, but notably improved performance is obtained for CGBA-H with a sufficient query budget.



(a) (b)

Figure 8: (a) Impact of the number of random direction K to obtain $\boldsymbol{x}_{b_1}$; (b) Performance comparison between initialization with $K = 1$ and $K = 50$.

Impact of initialization. In the above experiments, the same random initialization was used for all methods for a

[2] https://github.com/MadryLab/robustness

fair comparison. In this part, we discuss the impact of initialization on the performance of proposed methods on targeted attacks as mentioned in 4.3. Figure 8a depicts the amount of perturbation and corresponding required queries to find $\boldsymbol{x}_{b_1}$ with different numbers of random directions by using K samples of the target class. From this figure, a significant reduction in perturbation is observed with the increase of K. While with the random initialization, $K = 1$, the obtained perturbation is around 85 by spending about 20 queries, a reduction in perturbation of more than 30 is obtained with $K = 50$ by a small additional query cost of around 110. Figure 8b compares the performance of CGBA and CGBA-H with two different initialization: $K = 1$ and $K = 50$. Because a better initial boundary point is obtained by $K = 50$ (with additional query cost properly counted), both CGBA and CGBA-H converge faster towards optimal perturbation than initialization with $K = 1$. It's worth noting that the proposed initialization method can also be used to boost the baselines' performance.

6. Conclusion

In this work, we have proposed two novel decision-based black-box attacks: CGBA and CGBA-H, which use a semicircular trajectory in a restricted 2D plane to ensure finding a new boundary point with reduced perturbation regardless of the boundary's curvature. While CGBA outperforms the SOTA non-targeted attacks by effectively utilizing the low curvature of the decision boundary, CGBA-H is adapted to the high curvature of the decision boundary, resulting in better performance for targeted attacks. Furthermore, we have introduced an initialization algorithm that can be used to find a better initial boundary point to further boost the performance for decision-based targeted attacks. We have conducted extensive experiments to verify the effectiveness of the proposed attacks.

Acknowledgements

This work was supported in part by the US National Science Foundation under grants CNS-1824518 and ECCS-2203214. T. Wu is partly supported by ARO Grant W911NF1810295, NSF IIS-1909644, ARO Grant W911NF2210010, NSF IIS-1822477, NSF CMMI-2024688 and NSF IUSE-2013451. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ARO, NSF, DHHS or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright annotation thereon. The authors are grateful for the constructive comments by anonymous reviewers and area chairs.

References

[1] 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.

[2] Thomas Brunner, Frederik Diehl, Michael Truong Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4958–4966, 2019.

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[4] 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*, pages 1739–1747, 2020.

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

[6] 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*, pages 15–26, 2017.

[7] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, Jin-Feng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *International Conference on Learning Representations*, 2018.

[8] Minhao Cheng, Simranjit Singh, Patrick H Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*, 2020.

[9] 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*, pages 248–255, 2009.

[10] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7714–7722, 2019.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

[13] 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*, pages 770–778, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European*

conference on computer vision, pages 630–645. Springer, 2016.

[15] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018.

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[17] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1221–1230, 2020.

[18] Jie Li, Rongrong Ji, Peixian Chen, Baochang Zhang, Xiaopeng Hong, Ruixin Zhang, Shaoxin Li, Jilin Li, Feiyue Huang, and Yongjian Wu. Aha! adaptive history-driven attack for decision-based black-box models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16168–16177, 2021.

[19] Yujia Liu, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. A geometry-inspired decision-based attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4890–4898, 2019.

[20] Chen Ma, Xiangyu Guo, Li Chen, Jun-Hai Yong, and Yisen Wang. Finding optimal tangent points for reducing distortions of hard-label attacks. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] 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.

[22] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Surfree: a fast surrogate-free black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10430–10439, 2021.

[23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9078–9086, 2019.

[25] 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.

[26] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: a geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8446–8455, 2020.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Xiaosen Wang, Zeliang Zhang, Kangheng Tong, Dihong Gong, Kun He, Zhifeng Li, and Wei Liu. Triangle attack: A query-efficient decision-based adversarial attack. In *ECCV*, 2022.

[29] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *International Conference on Learning Representation*, 2020.

[30] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.