Finding Optimal Tangent Points for Reducing Distortions of Hard-label Attacks

Chen Ma 1  Xiangyu Guo 2  Li Chen 1,∗  Jun-Hai Yong 1  Yisen Wang 3,4
1 School of Software, BNRist, Tsinghua University, Beijing, China
2 Department of Computer Science and Engineering, University at Buffalo, Buffalo NY, USA
3 Key Lab. of Machine Perception, School of Artificial Intelligence, Peking University, Beijing, China
4 Institute for Artificial Intelligence, Peking University, Beijing, China
machenstar@163.com, xiangyug@buffalo.edu
{chenlee, yongjh}@tsinghua.edu.cn, yisen.wang@pku.edu.cn

Abstract

One major problem in black-box adversarial attacks is the high query complexity in the hard-label attack setting, where only the top-1 predicted label is available. In this paper, we propose a novel geometric-based approach called Tangent Attack (TA), which identifies an optimal tangent point of a virtual hemisphere located on the decision boundary to reduce the distortion of the attack. Assuming the decision boundary is locally flat, we theoretically prove that the minimum $\ell_2$ distortion can be obtained by reaching the decision boundary along the tangent line passing through such tangent point in each iteration. To improve the robustness of our method, we further propose a generalized method which replaces the hemisphere with a semi-ellipsoid to adapt to curved decision boundaries. Our approach is free of pre-training. Extensive experiments conducted on the ImageNet and CIFAR-10 datasets demonstrate that our approach can consume only a small number of queries to achieve the low-magnitude distortion. The implementation source code is released online at https://github.com/machanic/TangentAttack.

1 Introduction

Adversarial attacks cause deep neural networks (DNNs) to make incorrect predictions by slightly perturbing benign images during the test time. They can be divided into two main categories on the basis of the amount of information exposed by the target model, namely white-box and black-box attacks. Many white-box attacks [6, 30, 33] have been proposed, and they can compute the gradients w.r.t. the target model’s input images to generate adversarial examples with the first-order optimization techniques. In contrast, black-box attacks are more practical because they craft adversarial examples without requiring the target model’s gradients.

Over the past years, the community has made considerable efforts in developing black-box attacks, and the proposed methods can be divided into transfer- and query-based attacks. Transfer-based attacks [26, 40, 41] generate adversarial examples by using a white-box attack method against a surrogate model to fool the target model. Although there is no need to query the target model in these attacks, the attack success rate can not be guaranteed, especially in the case of targeted attacks. To achieve satisfactory attack success rates, the query-based attacks use elaborate queries to obtain the feedback of the target model for crafting adversarial examples. In the score-based setting, the query-based attacks [2, 12, 22, 29] estimate approximate gradients by querying the predicted scores of the target model at multiple points. However, in most real-world scenarios, the score-based setting

∗Corresponding author.
Figure 1: Simplified two-dimensional illustration of our motivation. $H$ is the decision boundary, and $x_{t-1}$ is the current adversarial example mapped onto the decision boundary at the $(t-1)$-th iteration. HSJA updates $x_{t-1}$ along the gradient direction to reach $g$ and then maps it to $H$ at $x_g$ along the line through $x$ and $g$. However, the optimal update should be the tangent point $k$ because it can be mapped onto $H$ at $x_t$ that has the shortest distance to the original image $x$.

is not applicable because the public service returns only the top-1 predicted label (i.e., the hard label) rather than the predicted score. In this case, since the feedback information is limited and the objective function is discontinuous, the attack requires solving a high-dimensional combinatorial optimization problem, which is often challenging.

To reformulate the attack as a real-valued continuous optimization problem, OPT [10], Sign-OPT [11], and RayS [8] focus on minimizing an objective function $g(\theta)$, which is defined as the distance from the original image to the nearest adversarial example along the direction $\theta$. However, when attacking complex models, it may be difficult to find a suitable direction $\theta$ along which adversarial examples exist.

Boundary Attack (BA) [4], HopSkipJumpAttack (HSJA) [7], QEBA [25], qFool [27], and Policy Driven Attack (PDA) [43] eliminate the search of the direction $\theta$. Instead, they start from a large adversarial perturbation and then reduce its distortion while staying in the adversarial region. Because the output labels of the target model flip only near the decision boundary, these attacks restrict their explorations to the regions near the decision boundary. However, they do not thoroughly investigate the geometric properties of the decision boundary to accelerate the attack. For example, PDA uses a reinforcement learning framework to train a policy network to predict search directions, which are not geometrically optimal. In addition, the prediction accuracy of the policy network decreases significantly in the later stages of the iterations, resulting in worse performance of these iterations.

HSJA and qFool simply use the gradient $u$ estimated at the decision boundary as the direction of each update, while ignoring a geometrically critical issue, i.e., $u$ is not the optimal direction to be followed (Fig. 1). We could explore better search directions at each attack iteration.

To find the optimal search direction for minimizing the distortions of attack, we propose a new geometric-based approach whose motivation is illustrated in Fig. 1. We construct a virtual semicircle $B$ centered at $x_{t-1}$ to indicate all possible locations that $x_{t-1}$ can reach along different directions, and the radius of $B$ limits the range of updates for successful attacks. It is easy to observe that moving along the tangent line can reach the nearest location of the decision boundary to the original image $x$, thereby producing the adversarial example with the minimum distortion. In real attack scenarios, the image data reside in a high-dimensional space, and the semicircle becomes a hemisphere. In this case, the benefit of using tangent points still exists, and we provide the detailed description and the formal proof in Section 3 and appendix.

To summarize, the main contributions of this study are as follows.

1. We cast the problem of minimizing the distortion in hard-label attacks into a geometric problem. We discover that the minimum distortion can be obtained by searching the optimal tangent point of a virtual hemisphere around the adversarial example at each iteration.

2. We propose a novel geometric-based method to obtain a closed-form solution of the optimal tangent point. We provide an intuitive explanation of our approach, as well as a formal proof of its correctness.

3. To improve robustness, we further propose a generalized method that replaces the hemisphere with a semi-ellipsoid to adapt to the target models with curved decision boundaries.

4. Extensive experiments conducted on the CIFAR-10 [24] and ImageNet [14] datasets demonstrate the effectiveness of our approach.
2 Related Work

Query-based black-box attacks can be divided into score- and decision-based attack (a.k.a. the hard-label attack). Score-based attacks \([1, 2, 3, 9, 12, 22, 29, 32]\) use the predicted probability score to craft adversarial examples, which is not always available in most real-world systems. Hard-label attacks are more useful, but obviously more challenging, because only the top-1 predicted label can be obtained. The hard-label attacks usually fall into three categories.

The first category starts from the original image \(x_0\) and attempts to find a optimal direction \(\theta\) to reach the adversarial example. OPT \([10]\) searches an optimal \(\theta\) to minimize the distance from \(x_0\) to the nearest adversarial example. Sign-OPT \([11]\) improves the query efficiency of OPT by using a single query to estimate the sign of the directional derivative. RayS \([8]\) eliminates the gradient estimation and proposes a fast check step to efficiently find the direction \(\theta\). However, RayS is only applicable to the untargeted attack under the \(\ell_\infty\) norm because it is difficult to find a suitable direction to reach the region of the target class in a targeted attack, especially in the case of a large number of classes.

The second category starts from a large perturbation or an image of the target class, and then reduces its distortion while staying in adversarial region, thereby gradually making it closer to the original image. BA \([4]\) and NES \([1]\) are two representative methods, but they have high query complexity. Biased BA \([5]\) reinterprets BA as a biased sampling framework and incorporates different biases to improve the query efficiency. HSJA \([7]\) utilizes the gradient estimation and the binary search to outperform BA. HSJA can be used as the baseline of hard-label attacks. QEB \([25]\) improves HSJA by using dimension reduction techniques. SurFree \([31]\) and GeoDA \([34]\) improve the performance of HSJA by exploiting geometric properties of DNNs. However, the geometric features of DNNs have not been fully explored, and they do not support the targeted attack. PDA \([43]\) uses a reinforcement learning framework to train a policy network to predict promising directions. However, PDA requires a high-cost pre-training, which is not always available in all tasks.

The third category uses a random sampling technique to improve query efficiency. Customized Adversarial Boundary (CAB) attack \([37]\) uses the current noise to select the sensitive area of images and customize sampling distribution. Evolutionary \([16]\) improves the query efficiency by reducing the dimension of the search space with the stochastic coordinate selection.

The issue of RayS and GeoDA is that they only support untargeted attacks. Because moving the original image far enough in any direction can always make it escape the non-adversarial region in untargeted attacks. However, in targeted attacks, it is difficult to find a suitable direction along which the target class’s region exists. In contrast, our approach supports all types of attacks, and we exploit the geometric characteristics of the decision boundaries of DNNs to boost the attack.

3 The Proposed Approach

3.1 The Goal of Hard-label Attacks

Given a target model \(f : \mathbb{R}^d \rightarrow \mathbb{R}^k\) and a benign image \(x \in [0, 1]^d\) that is correctly classified using \(f\), the goal of the adversary is to slightly perturb \(x\) into \(x_{\text{adv}}\), such that \(f(x_{\text{adv}})\) outputs incorrect prediction. In the hard-label attack, the adversary can only observe the top-1 predicted label of \(f\), denoted as \(\hat{y} = \arg \max_i f(x_{\text{adv}})_i\). We define an indicator function \(\phi(\cdot)\) of a successful attack:

\[
\phi(x_{\text{adv}}) := \begin{cases} 
1 & \text{if } \hat{y} = y_{\text{adv}} \text{ in the targeted attack,} \\
0 & \text{otherwise,}
\end{cases} \quad (1)
\]

where \(y \in \mathbb{R}\) is the true label of \(x\) and \(y_{\text{adv}} \in \mathbb{R}\) is a predefined target class label. In this study, we focus on generating an adversarial example \(x_{\text{adv}}\) that satisfies \(\phi(x_{\text{adv}}) = 1\), such that the distortion \(d(x_{\text{adv}}, x) := ||x_{\text{adv}} - x||_p\) is minimized. This goal can be formulated as the optimization problem:

\[
\min_{x_{\text{adv}}} d(x_{\text{adv}}, x) \quad \text{s.t.} \quad \phi(x_{\text{adv}}) = 1. \quad (2)
\]

3.2 Motivation

Most recent attacks belong to the second category (Section 2) follow a common procedure: it starts from an adversarial image yet not close enough to the benign one, then it iteratively searches for a
closer adversarial image. Let us take a typical attack HSJA [7] for example (Fig. 1). First, the attack initializes the adversarial example by using an image of the target class (in a targeted attack) or a noisy version of the original image (in an untargeted attack). Next, it performs the binary search to map the initial sample onto the decision boundary \( H \), denoted as \( x_{t-1} \). Then, the algorithm iteratively performs three steps to update \( x_{t-1} \): estimating gradient \( u \) at \( H \) by sampling many probes around \( x_{t-1} \); jumping to the point \( g \) along the direction \( u \) with the step size determined by the geometric progression; and mapping \( g \) onto \( H \) at \( x_g \) by performing the binary search along the line passing through \( g \) and the original image \( x \). However, geometrically speaking, the estimated gradient \( u \) is not the optimal search direction, and we can find a better boundary point \( x_t \) by connecting \( x \) to a certain point on the semicircle \( B \) and then taking the intersection point on \( H \). Fig. 1 shows that the tangent point \( k \) is the optimum update because it leads to the minimum distortion of the attack.

### 3.3 Definition of Optimal Tangent Points

There is some experimental evidence that the decision boundaries of DNNs are smooth surfaces with the low curvature [17, 34]. Based on this observation, we approximate the local decision boundary with a hyperplane \( H \). Fig. 3 illustrates our problem in three-dimensional space. We will derive our algorithm from \( \mathbb{R}^3 \) and show that it can be directly extended to higher-dimensional spaces. Lastly, we address the case of curved decision boundaries.

As described in Section 3.2, \( x_{t-1} \) denotes the boundary sample that already lies on the decision boundary. We create a virtual hemisphere \( B \) centered at \( x_{t-1} \) with a radius \( R \), which is an estimation of the safe region where we can search for adversarial examples. In the targeted attack, \( B \) represents the current estimation of the target class region around \( x_{t-1} \). Ideally, \( B \) is small enough to be fully contained in the target class’s region, i.e., \( \phi(x') = 1 \) for \( \forall x' \in B \). Then, we need to find a point on \( B \) which would produce the minimum distortion when mapped to \( H \). In two-dimensional case, this point is the tangent point. However, in \( n \)-dimensional space where \( n \geq 3 \), there are infinitely many tangent lines of \( B \) passing through \( x \) which create infinitely many tangent points on \( B \), shown as the red points in Figs. 2 and 3. Still, we will show that exactly one tangent point can lead to the minimum distortion when mapping it onto \( H \) along the tangent line.

Formally, let \( k \) be any point on the surface of \( B \), \( u \) be the approximate gradient of \( H \) estimated at \( x_{t-1} \), and \( x_t \) be the intersection of \( H \) and the line passing through \( x \) and \( k \), we have the following theorem.

**Theorem 1.** Let \( H, u, k, x, x_{t-1} \) be defined above, then the distance \( \|x - x_t\| \) is minimized if \( k \) is the optimal solution of the following constrained optimization problem:

\[
\arg \max_k \quad \langle k - x_{t-1}, u \rangle \tag{3}
\]

subject to \( \|k - x_{t-1}\| = R \) \( \tag{4} \) and \( \langle k - x_{t-1}, u \rangle \geq 0 \). \( \tag{5} \)

In particular, the optimal \( k \) is in the plane spanned by \( u \) and \( x - x_{t-1} \).

The objective function of Eq. (3) is to maximize the projection of the vector \( k - x_{t-1} \) onto \( u \), which is equivalent to finding \( k \) that is farthest away from \( H \). The first constraint ensures that \( k \) is a tangent point. The second constraint indicates \( k \) is on the surface of \( B \). The last constraint states that \( k \) cannot appear on the same side of \( H \) as \( x \), which is always satisfied if \( \|\Pi_H(x - x_{t-1})\| \geq R \), where the notation \( \Pi_H : \mathbb{R}^n \rightarrow H \) denotes the orthogonal projection from \( \mathbb{R}^n \) onto the hyperplane \( H \). If there is no feasible solution, then our algorithm (Algorithm 1) reduces the radius \( R \) to guarantee that the last constraint is always satisfied. The formal proof of Theorem 1 is presented in the appendix. Here,
we refer the readers to Fig. 2 for an intuitive geometrical explanation. Eq. (3) is computationally expensive to solve in the high-dimensional space. Fortunately, we show there actually exists a closed-form solution.

### 3.4 Closed-Form Solution of the Optimal Tangent Point

The main intuition of the derivation is illustrated in Fig. 3, which shows an example in $\mathbb{R}^3$. For ease of presentation, we move $x_{t-1}$ to 0. The known variables are $x$, $x_{t-1}$, the unit normal vector $u$ of hyperplane $H$, and the radius $R$. We need to solve for the unknown $k$. Let $V = \text{span}(\{x, u\})$ be the plane spanned by $x$ and $u$. According to Theorem 1, we know $k \in V$. Let us denote the angle between $x$ and $H$ as $\alpha$, the angle between $x$ and $k$ as $\beta$, and the angle between $k$ and $H$ as $\gamma$. Then, $\beta = \alpha + \gamma$ because all three points $x$, $k$, and $x_{t-1}$ are on the same plane $V$. Because the angle between $x$ and $u$ is $\pi/2 + \alpha$, we have

$$
\langle x, u \rangle = ||x|| \cdot ||u|| \cdot \cos \left( \frac{\pi}{2} + \alpha \right) = ||x|| \cdot ||u|| \cdot (-\sin \alpha).
$$

(7)

Then we have $\sin \alpha = -\frac{\langle x, u \rangle}{||x|| \cdot ||u||}$ and $\cos \alpha = \sqrt{1 - \sin^2 \alpha} = \sqrt{\frac{||x||^2 \cdot ||u||^2 - \langle x, u \rangle^2}{||x||^2 \cdot ||u||^2}}$. By the constraint (4), $x - k$ is orthogonal to $k$. Thus, we have $\cos \beta = R / ||x||$ and $\sin \beta = \sqrt{1 - \cos^2 \beta} = \sqrt{||x||^2 - R^2 / ||x||}$. Then, $\sin \gamma$ and $\cos \gamma$ can be derived as functions of $\alpha$ and $\beta$ from basic facts of trigonometric functions:

$$
\sin \gamma = \sin(\beta - \alpha) = \sin \beta \cos \alpha - \cos \beta \sin \alpha,
$$

$$
\cos \gamma = \cos(\beta - \alpha) = \cos \beta \cos \alpha + \sin \beta \sin \alpha.
$$

(8)

Now, let $k' \in H$ be the orthogonal projection of $k$ onto the plane $H$. The distance between $k$ and $k'$ is denoted as $h$ (Fig. 3). Then $h = R \cdot \sin \gamma = R \cdot (\sin \beta \cos \alpha - \cos \beta \sin \alpha)$.

To derive $k'$, let us denote $x'$ as the orthogonal projection of $x$ onto $z$ axis. So we have $x' = \langle x, u \rangle \cdot (-u) / ||u||^2 = \langle x, u \rangle \cdot u / ||u||^2$. Then, because $k'$ and $x - x'$ are on the same direction, we have

$$
k' = \frac{x - x'}{||x - x'||} \cdot ||k'|| = \frac{x - \langle x, u \rangle \cdot u / ||u||^2}{||x - \langle x, u \rangle \cdot u / ||u||^2||} \cdot R \cdot \cos \gamma.
$$

(9)

Now, $||k'|| = R \cdot \cos \gamma$ and $x' = \langle x, u \rangle \cdot u / ||u||^2$ are plugged into Eq. (9), and $k'$ is obtained as

$$
k' = \frac{x - \langle x, u \rangle \cdot u / ||u||^2}{||x - \langle x, u \rangle \cdot u / ||u||^2||} \cdot R \cdot \cos \gamma.
$$

(10)

Therefore, $k$ can be derived as

$$
k = k' + h \cdot u = \frac{x - \langle x, u \rangle \cdot u / ||u||^2}{||x - \langle x, u \rangle \cdot u / ||u||^2||} \cdot R \cdot \cos \gamma + h \cdot u.
$$

(11)
When the local decision boundary is not flat enough, the boundary point obtained via the tangent line may not be the optimal, as shown in Fig. 4a. A simple solution is to simply continue halving the radius $R$ of the hemisphere: as long as $R$ becomes small enough, the local flatness can always be obtained. However, too small a radius will reduce the convergence rate of our algorithm, because the distance between $x_t$ and $x_{t-1}$ is proportional to $R$. Therefore, when the classification decision boundary is a curved surface, the attack algorithm should change the shape of hemisphere rather than simply reducing $R$. Based on this idea, we propose the Generalized Tangent Attack (G-TA).

First, although the shape of the decision boundary can be very complex in a high-dimensional space, the important thing for our algorithm is only the situation in the two-dimensional plane spanned by $x$ and $u$. In particular, if the decision boundary is “downward” curved (as opposed to the example in Fig. 4a), then searching along the tangent line is still a better approach than HSJA’s solution. Thus, the only “bad case” we have to deal with is when the decision boundary is “upward” curved and has a large curvature, as shown in Fig. 4a.

According to Theorem 1, we only need to focus on the plane $V$ spanned by $x$ and $u$, as shown in Fig. 4b. Now, $x$ and $v := (x - \langle x, u \rangle \cdot u) / \|u\|^2$ form an orthogonal basis of the plane $V$, then $x$ can be identified with coordinates $(x_0, z_0)$, i.e., $x = x_0v + z_0u$. Let $\theta$ denote the angle between the vector $x$ and the vector $-u$, i.e., $\theta = \arccos \left(\frac{\langle x, -u \rangle}{\|x\| \cdot \|u\|}\right)$. Then $(x_0, z_0) = (\|x\| \cdot \sin \theta, -\|x\| \cdot \cos \theta)$. Consider the projection of the ellipsoid on $V$ (which is an ellipse), we denote $L$ as its radius along the direction of $u$, and $S$ as its radius along the direction of $v$. Because the optimal tangent point $k$ lies in the plane $V$, $k$ can also be identified as $k = x_kv + z_kv\_u$, and we only need to solve for the unknown $(x_k, z_k)$.

The ellipse is characterized by the equation $x^2 / S^2 + z^2 / L^2 = 1$, thus the tangent point $k$ satisfies $x_k^2 / S^2 + z_k^2 / L^2 = 1$. Now we view $z$ as a function of $x$, and take the derivative w.r.t. $x$ at both sides of the equation to get the following formula:

$$\frac{2x_k}{S^2} + \frac{2z_k}{L^2} \cdot \frac{dz}{dx} \bigg|_{x=x_k} = 0. \quad (12)$$

Thus, we have the slope of tangent line at $k$ be $\frac{dz}{dx} \bigg|_{x=x_k} = -\frac{x_kL^2}{z_kS^2}$. Therefore, the tangent line can be written as $z - z_k = -\frac{x_kL^2}{z_kS^2}(x - x_k)$. Since the tangent line passes through $x$, we know
$$z_0 - z_k = -\frac{z_k L^2}{z_0 S^2} (x_0 - x_k).$$ In summary, we can obtain the following system of equations:

$$\begin{align*}
L^2 x_k^2 + S^2 z_k^2 - x_k x_0 L^2 - z_0 z_k S^2 &= 0, \\
\frac{x_k^2}{S^2} + \frac{z_k^2}{S^2} &= 1.
\end{align*}$$

(13)

In Eq. (13), the known variables are $L$, $S$, $x_0$ and $z_0$, and the unknown variables that we need to solve for are $x_k$ and $z_k$. In general, there are two solutions for Eq. (13), i.e., $k$ and $k'$ depicted in Fig. 4b. Apparently, the solution of the optimal tangent point should satisfy $z_k > 0$, so the one of two solutions in which $z_k > 0$ should be picked:

$$\begin{align*}
x_k &= \frac{S^2 \left( L^2 - z_0 - \frac{L^2 S^2 z_0 + L^2 x_0 \sqrt{-L^2 S^2 + L^2 x_0^2 + S^2 z_0^2}}{L^2 x_0^2 + S^2 z_0^2} \right)}{L^2 \cdot x_0}, \\
z_k &= \frac{L^2 S^2 z_0 + L^2 x_0 \sqrt{-L^2 S^2 + L^2 x_0^2 + S^2 z_0^2}}{L^2 x_0^2 + S^2 z_0^2}.
\end{align*}$$

(14)

Finally, the optimal tangent point $k$ is obtained via $k = |x_k| \cdot \frac{x - (\langle x, u \rangle / \|u\|^2) u}{x - (\langle x, u \rangle / \|u\|^2) u} + z_k \cdot u$. In the implementation, the value of $L$ is determined in the same way as the radius $R$ in TA (the hemisphere version). So we fix $L = R$, and use a hyperparameter $r = L/S$ to control the value of $S$. Imagine that in the case of $\mathbb{R}^3$, the semi-ellipsoid becomes “slender” by setting $r > 1$, thereby adapting to the decision boundary of curved surface while preserving a relatively large step size.

### 3.6 The Complete Algorithm

TA (the hemisphere version) and G-TA (the semi-ellipsoid version) can be combined into one algorithm process, which is shown in Algorithm 1. It first performs a binary search to map the initial sample $x_0$ to the decision boundary. Note that the binary search step always maps any $x_{adv}$ to the adversarial side of $H$ that satisfies $\phi(x_{adv}) = 1$, hence the attack success rate is always 100%. Then, it performs a for loop of $T$ iterations to find the adversarial example that is close to $x$. In the first iteration, we sample $B_0$ probes around the boundary sample to estimate the gradient, which is increased to $B_0 \sqrt{T}$ at the $t$-th iteration. This is because the error of gradient estimation in the later iterations has a greater impact on the attack performance, so using more samples can reduce the estimation error. Then, a while loop is performed to determine a reasonable radius $R$ by repeatedly halving the radius until the tangent point $k$ is in the adversarial region. Finally, Algorithm 1 uses the binary search method to map $k$ back to the classification decision boundary to end this iteration.

**Algorithm 1** Tangent Attack

**Input:** benign image $x$, attack success indicator function $\phi(\cdot)$ defined in Eq. (1), initial batch size $B_0$, iteration $T$, mode $m \in \{\text{semi-ellipsoid, hemisphere}\}$, radius ratio $r$.

**Initialize** $x_0$ that satisfies $\phi(x_0) = 1$; $x_0 \leftarrow \text{BinarySearch}(x_0, x, \phi)$; $d_0 = ||x_0 - x||$; $\triangledown$ boundary search

**for** $t$ in $1, \ldots, T$ **do**

**Sample** $B_t \leftarrow B_0 \sqrt{T}$ random vectors to estimate the gradient $u$;

**Initialize** $R \leftarrow d_{t-1} \sqrt{T}$; $\triangledown$ the initial radius

**while** true **do**

**Compute** the optimal tangent point $k$ based on Eq. (11) **if** $m = \text{hemisphere}$ **else** Eq. (14);

$R \leftarrow \frac{R}{r}$; $\triangledown$ search the radius, and we set $L = R, S = \frac{L}{r}$ if $m = \text{semi-ellipsoid}$

**if** $\phi(k) = 1$ **then**

break;

end if

end while

$k \leftarrow \text{Clip}(k, 0, 1)$; $\triangledown$ boundary search

$x_t \leftarrow \text{BinarySearch}(k, x, \phi)$;

$d_t = ||x_t - x||$;
4 Experiment

4.1 Experimental Setting

Datasets. TA and G-TA are evaluated on two datasets, namely CIFAR-10 and ImageNet with the image resolutions of $32 \times 32 \times 3$ and $299 \times 299 \times 3$, respectively. We randomly select 1,000 correctly classified images from their validation sets for experiments.

Method Setting. The initial batch size $B_0$ is set to 100, which means the algorithm samples 100 probes for estimating a gradient at the first iteration. The threshold $\gamma$ that controls the termination of the binary search is set to 1.0 in the CIFAR-10 dataset and 1.1 in the ImageNet dataset. The radius ratio $r$ is set to 1.5 in the CIFAR-10 dataset and 1.1 in the ImageNet dataset. Besides, we also set $r$ to 1.5 when attacking defense models. In targeted attacks, the target class label is set to $y_{adv} = (y + 1) \mod C$, where $y$ is the true label, and $C$ is the number of classes.

Compared Methods. The advantage of our method is that it supports all types of attacks, including both untargeted and targeted attacks under $\ell_2$ norm and $\ell_\infty$ norm constraints. Therefore, for complete and fair comparisons, we select the compared methods that support both untargeted and targeted attacks with state-of-the-art performance, including Boundary Attack (BA) [4], Sign-OPT [11], SVM-OPT [11], and HopSkipJumpAttack (HSJA) [7]. HSJA is adopted as the baseline, whose hyperparameters are set to be the same with ours (e.g., the same initial batch size $B_0$ and threshold $\gamma$). In addition, QEBA [25] is a HSJA-based method which has three variants: QEBA-I, QEBA-S and QEBA-F. We select QEBA-S in the additional experiment to verify whether the proposed method can improve attack performance of other HSJA-based method. For the consistency of experiments, we translate the implementations of Sign-OPT, SVM-OPT and HSJA from the official NumPy version into the PyTorch version by replacing each NumPy function with the corresponding PyTorch function. Thus, the two versions behave exactly the same. In the targeted attack, we randomly select an image from the target class of the validation set to be the initial sample of HSJA, BA, TA and G-TA. For fair comparison, we set the initial direction $\theta_0$ of Sign-OPT and SVM-OPT to the direction of a randomly selected image of the target class. The detailed settings are presented in the appendix.

Target Models. In the CIFAR-10 dataset, we select four target models implemented using the PyTorch framework: (1) a 272-layer PyramidNet+ShakeDrop network (PyramidNet-272) [19, 42] trained using AutoAugment [13]; (2) a model obtained through a neural architecture search called GDAS [15]; (3) a wide residual network with 28 layers and 10 times width expansion (WRN-28) [44]; and (4) a wide residual network with 40 layers (WRN-40) [44]. In the ImageNet dataset, we select four target models from an off-the-shelf library containing pre-trained weights: (1) Inception-v3 [39], (2) Inception-v4 [38], (3) ResNet-101 [20], and (4) SENet-154 [21].

Evaluation Metric. Following previous studies [43], we report the mean $\ell_2$ distortions as $\frac{1}{|X|} \sum_{x \in X} \|x_{adv} - x\|$ under different query budgets, where $X$ is the test set.

4.2 Comparisons with State-of-the-Art Methods

Results of Attacks against Undefended Models. Tables 1 and 2 show the experimental results on the ImageNet and CIFAR-10 datasets. We derive two conclusions based on the results:

1. We found that the experiments of CIFAR-10 requires a larger radius ratio $r$ than that of ImageNet to achieve the satisfactory performance of G-TA. We speculate that the reason is that the target models of ImageNet have relatively flat decision boundaries.

2. TA is more effective in targeted attacks, while the G-TA performs better in untargeted ones. This is because the adversarial region of the target class is narrower and more scattered, making the local classification decision boundary smoother, so that Theorem 1 holds and TA performs better.

In addition, one unique benefit of our approach is that it can be used as a performance enhanced plug-in when combining it with other HSJA-based approaches (e.g., QEBA-S). Specifically, the method is to change the jump directions of boundary samples of QEBA-S to the directions of the optimal tangent points. We present that “QEBA-S + TA” can further improve the performance of QEBA-S, as shown in Table 3.

---

Pre-trained weights: https://github.com/machanic/SimulatorAttack
Pre-trained weights: https://github.com/Cadene/pretrained-models.pytorch
Table 1: Mean $\ell_2$ distortions of different query budgets on the ImageNet dataset, where $r = 1.1$.

| Target Model | Method | Targeted Attack | Untargeted Attack |
|--------------|--------|-----------------|------------------|
|              | @300  | @1K  | @2K  | @5K  | @8K  | @10K |
| ResNet-101   | SVM-OPT [11] | 79.314 | 75.547 | 64.124 | 53.917 | 40.725 |
|              | Sign-OPT [11] | 79.272 | 75.520 | 64.167 | 53.964 | 40.772 |
|              | HSJA [7] | 79.208 | 75.419 | 64.137 | 53.917 | 40.725 |
|              | TA     | 79.166 | 75.399 | 64.110 | 53.917 | 40.725 |
|              | G-TA   | 79.591 | 55.224 | 44.047 | 25.164 | 15.442 |

Table 2: Mean $\ell_2$ distortions with different query budgets on the CIFAR-10 dataset, where $r = 1.5$.

| Target Model | Method | Targeted Attack | Untargeted Attack |
|--------------|--------|-----------------|------------------|
|              | @300  | @1K  | @2K  | @5K  | @8K  | @10K |
| PyramidNet-272 | SVM-OPT [11] | 8.651 | 6.873 | 4.837 | 3.333 |
|               | HSJA [7] | 7.917 | 6.432 | 4.288 | 2.076 |
|               | TA     | 7.943 | 6.426 | 4.277 | 2.076 |
|               | G-TA   | 7.816 | 6.427 | 4.277 | 2.076 |

Table 3: Experimental results of the combined method of QEBA-S and TA.

| Target Model | Method | Targeted Attack |
|--------------|--------|-----------------|
|              | @300  | @1K  | @2K  | @5K  | @8K  | @10K |

Results of Attacks against Defense Models. We conduct the experiments of untargeted attacks on defense models. Fig. 5 shows the experimental results on the CIFAR-10 dataset. We select 4 defense models: (1) a model obtained through adversarial training, abbreviated as AT [30]; (2) an
In this paper, we propose a new geometric-based method for query-efficient hard-label black-box attacks. Our method relies on the observation that the minimum $\ell_2$ distortion can be obtained by searching a boundary point along a tangent line of a virtual hemisphere. We offer a closed-form solution for computing the optimal tangent point and provide a formal proof of its correctness. We further propose a generalized method that replaces the hemisphere with a semi-ellipsoid to adapt to the target models with curved decision boundaries. Lastly, we evaluate our approach through extensive experiments and show its superior performance compared with baseline methods.

### 4.3 Comprehensive Understanding of Tangent Attack

In the ablation studies, we conduct the experiments of the targeted attacks on the ImageNet dataset to understand our approach in depth, and the target model is ResNet-50. The results are shown in Fig. 6.

**Initialization.** Our algorithm starts with an image $\tilde{x}_0$ selected from the target class, and we study three selection strategies: (1) a randomly selected image, (2) the image with the shortest distance to the original image, and (3) the image with the longest distance to the original image. Fig. 6a shows that the strategy of (2) achieves the best performance.

**Radius Ratio.** Fig. 6b shows that the performance of G-TA is not sensitive to the radius ratio $r$.

**Jump Direction.** Fig. 6c shows the effects of different jump directions. RandomHSJA is a variant of HSJA which adopts a random vector $r$ that satisfies $\langle r, u \rangle > 0$ as the jump direction. Fig. 6c verifies the benefit of jumping to the optimal tangent point.

**Initial Batch Size.** In general, Fig. 6d shows that a smaller $B_0$ performs better since it saves queries. But $B_0 = 5$ performs worse than $B_0 = 30$ because it uses too few samples for gradient estimation.

### 5 Conclusion

In this paper, we propose a new geometric-based method for query-efficient hard-label black-box attacks. Our method relies on the observation that the minimum $\ell_2$ distortion can be obtained by searching a boundary point along a tangent line of a virtual hemisphere. We offer a closed-form solution for computing the optimal tangent point and provide a formal proof of its correctness. We further propose a generalized method that replaces the hemisphere with a semi-ellipsoid to adapt to the target models with curved decision boundaries. Lastly, we evaluate our approach through extensive experiments and show its superior performance compared with baseline methods.
Acknowledgments

This research is partially supported by the National Key R&D Program of China (2019YFB1405703) and TC190A4DA/3, the National Natural Science Foundation of China (Grant Nos. 62021002, 61972221). Yisen Wang is partially supported by the National Natural Science Foundation of China under Grant 62006153, and Project 2020BD006 supported by PKU-Baidu Fund.

References

[1] Ilyas Andrew, Engstrom Logan, Athalye Anish, and Lin Jessy. Black-box adversarial attacks with limited queries and information. In International Conference on Machine Learning, pages 2142–2151, 2018.

[2] Yang Bai, Yuyuan Zeng, Yong Jiang, Yisen Wang, Shu-Tao Xia, and Weiwei Guo. Improving query efficiency of black-box adversarial attack. In European Conference on Computer Vision, pages 101–116, 2020.

[3] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In European Conference on Computer Vision, pages 158–174, 2018.

[4] 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, pages 1–12, 2018.

[5] T. Brunner, F. Diehl, M. T. Le, and A. Knoll. Guessing smart: biased sampling for efficient black-box adversarial attacks. In IEEE/CVF International Conference on Computer Vision, pages 4957–4965, 2019.

[6] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In IEEE Symposium on Security and Privacy, pages 39–57, 2017.

[7] Jianbo Chen, Jordan Michael I., and Wainwright Martin J. HopSkipJumpAttack: a query-efficient decision-based adversarial attack. In IEEE Symposium on Security and Privacy, pages 1277–1294, 2020.

[8] Jinghui Chen and Quanquan Gu. RayS: a ray searching method for hard-label adversarial attack. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 1739–1747, 2020.

[9] 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, pages 15–26, 2017.

[10] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, JinFeng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: an optimization-based approach. In International Conference on Learning Representations, pages 1–14, 2019.

[11] Minhao Cheng, Simranjit Singh, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-OPT: a query-efficient hard-label adversarial attack. In International Conference on Learning Representations, pages 1–16, 2020.

[12] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. In Advances in Neural Information Processing Systems, pages 10934–10944, 2019.

[13] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: learning augmentation strategies from data. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 113–123, 2019.

[14] 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, pages 248–255, 2009.

[15] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four GPU hours. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1761–1770, 2019.

[16] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7714–7722, 2019.

[17] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In Advances in Neural Information Processing Systems, page 1632–1640, 2016.
[18] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In International Conference on Learning Representations, pages 1–12, 2018.

[19] Dongyoon Han, Jiwan Kim, and Junmo Kim. Deep pyramidal residual networks. In IEEE Conference on Computer Vision and Pattern Recognition, pages 5927–5935, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7132–7141, 2018.

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 1–60, 2009.

[23] Jing Zhang, Yang Yuan, and Jiajun Liu. ComDefend: an efficient image compression model to defend adversarial examples. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6084–6092, 2019.

[24] Thibault Maho, Teddy Furon, and Erwan Le Merrer. SurFree: a fast surrogate-free black-box attack. In IEEE/CVF International Conference on Computer Vision, pages 4889–4897, 2019.

[25] Zhihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: DNN-oriented jpeg compression against adversarial examples. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 860–868, 2019.

[26] Chen Ma, Li Chen, and Jun-Hai Yong. Simulating unknown target models for query-efficient black-box attacks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7132–7141, 2018.

[27] 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, pages 2574–2582, 2016.

[28] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. GeoDA: a geometric framework for black-box adversarial attacks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8446–8455, 2020.

[29] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: a Python toolbox to benchmark the robustness of machine learning models. In Reliable Machine Learning in the Wild Workshop of International Conference on Machine Learning, pages 1–6, 2017.

[30] Jonas Rauber, Roland Zimmermann, Matthias Bethge, and Wieland Brendel. Foolbox Native: fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX. Journal of Open Source Software, 5(53):2607, 2020.

[31] Yucheng Shi, Yahong Han, and Qi Tian. Polishing decision-based adversarial noise with a customized sampling. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1030–1038, 2020.
[38] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*, page 4278–4284, 2017.

[39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zhigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[40] Xin Wang, Jie Ren, Shuyun Lin, Xiangming Zhu, Yisen Wang, and Quanshi Zhang. A unified approach to interpreting and boosting adversarial transferability. In *International Conference on Learning Representations*, pages 1–35, 2021.

[41] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: on the transferability of adversarial examples generated with ResNets. In *International Conference on Learning Representations*, pages 1–15, 2020.

[42] Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba, and Koichi Kise. Shakedrop regularization for deep residual learning. *IEEE Access*, 7:186126–186136, 2019.

[43] Ziang Yan, Yiwen Guo, Jian Liang, and Changshui Zhang. Policy-driven attack: learning to query for hard-label black-box adversarial examples. In *International Conference on Learning Representations*, pages 1–15, 2021.

[44] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, pages 87.1–87.12, 2016.

[45] Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems*, pages 1829–1839, 2019.

[46] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482, 2019.

[47] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, pages 1–29, 2021.
Appendix

A Potential Negative Societal Impacts

The adversarial attack is a major security concern in the real-world machine learning system, because the generated adversarial perturbation could be used for the malicious purpose. Our study relies only on the top-1 predicted label to craft the adversarial examples which is applicable to most real-world systems, making it more useful and practical. Although the experiments in this paper are about attacking the image classifier, this method can be used in other settings, such as the object detection, the recommender system, the facial recognition system, and the autonomous driving. In summary, this study could be used in harmful ways by malicious users.

In a broader perspective, the adversarial example is not restricted to malicious applications, and it can be used in the positive side, e.g., the generation of CAPTCHA and the privacy protection. In particular, the study of adversarial attacks can promote the defense techniques. In recent years, many proposed defenses are broken by the latest attacks, which stimulates the development of defenses.

Our results also point to the potential defense techniques against hard-label attacks. For example, the defense can prohibit queries near the decision boundary, then the approximate gradient cannot be estimated, making Tangent Attack ineffective. Another possible defense is to add random perturbations to the input image to prevent effective gradient estimation, or predict random classification labels for samples near the classification decision boundary.

B Proof of Theorem 1

B.1 Notations and Assumption

Before we formally prove Theorem 1, let us first define the notations that will be used in the proof. Let $x$ denote the original image, and w.l.o.g. we assume the boundary sample $x_{t-1} = 0$ be the origin of the coordinate axis. Let $B$ denote a $n$-dimensional ball centered at $x_{t-1}$ with the radius of $R$, and its surface is denoted as $S := \partial B$. Note that $B$ denotes a complete ball in this proof. However, $B$ denotes the hemisphere in the main text of the paper. Theorem 1 assumes that the classification decision boundary of the target model is the hyperplane $H$, which is defined by its unit normal vector $u$. Then, the hyperplane $H$ divides $\mathbb{R}^n$ into two half-spaces:

$$
H_{\geq 0} = \{v \in \mathbb{R}^n \mid \langle v, u \rangle \geq 0\},
$$

$$
H_{\leq 0} = \{v \in \mathbb{R}^n \mid \langle v, u \rangle \leq 0\}. \quad (15)
$$

In the attack, $H_{\geq 0}$ mainly contains the adversarial region, and $H_{\leq 0}$ represents the non-adversarial region. In Fig. 7, we visually represent the hyperplane $H$ and two half-spaces in $\mathbb{R}^3$.

Figure 7: Illustration of the entities defined in the proof, where $C$ is a convex cone whose boundary intersects with the circle formed by all the tangent points from $x$ to the ball $B$. 

14
Suppose \( x \in H_{\leq 0} \setminus B \) is a fixed point outside \( B \) such that \( \langle x, u \rangle < 0 \). Now, let us define the cosine function \( \cos(a, b) := \frac{(a, b)}{|a||b|} \) to represent the cosine of the angle between two vectors, then we can define the convex cone \( C \) with \( x_{t-1} \) as its vertex, as shown below:

\[
C := \left\{ v \in \mathbb{R}^n \mid \cos(v, x) \geq \frac{R}{||x||} \right\}.
\]  

(16)

Fig. 7 demonstrates the convex cone \( C \) in \( \mathbb{R}^3 \). For \( v \in S \cap C \) that satisfies \( \cos(v, x) = R / ||x|| \), the equation \( ||v - x||^2 = ||x||^2 - ||v||^2 \) holds, i.e., \( v \) is the tangent point of the tangent line from \( x \) to the surface of \( B \).

To make the feasible region of the optimization problem (3) in Theorem 1 nonempty, we need to make an assumption about the positional relationship between \( x \) and the ball \( B \). Let \( \Pi_H : \mathbb{R}^n \mapsto H \) denote the orthogonal projection from \( \mathbb{R}^n \) onto the hyperplane \( H \), we make the following assumption:

**Assumption B.1.** \( \Pi_H(x) \in C \)

Note that Assumption B.1 is not really an “assumption”: it essentially means that there is a tangent point on \( S \cap H_{\geq 0} \), which is in the adversarial region. Assumption B.1 means the feasible region of the optimization problem (3) is a nonempty set. By repeatedly reducing the radius \( R \), the algorithm guarantees that the optimal tangent point is in the adversarial region, thereby making Assumption B.1 always hold. In addition, according to Assumption B.1, \( ||\Pi_H(x)|| \geq R \) holds.

In Theorem 1, \( k \) is an arbitrary point on the surface of the hemisphere \( B \cap H_{\geq 0} \), so this proof mainly focuses on points in this region. In the following text, the hemisphere is denoted as \( B' := B \cap H_{\geq 0} \), and its surface is denoted as \( S' := S \cap H_{\geq 0} \) for brevity. Now, let us pick up any \( k \in S' \), and then the intersection point between the hyperplane \( H \) and the line passing through \( x \) and \( k \) is denoted as \( y_k \). Then, \( (y_k, \lambda) \) is the unique solution of the following equation system:

\[
y_k = \lambda k + (1 - \lambda)x,
\]

\[
\langle y_k, u \rangle = 0,
\]

\[
0 \leq \lambda \leq 1.
\]  

(17)

Because the position of \( k \) determines the distance between \( y_k \) and \( x \), we can define the function \( f(k) := ||y_k - x|| \) to represent the distance between \( x \) and \( y_k \).

**B.2 Proof**

To prove Theorem 1, we turn to prove the following lemma, which is equivalent to Theorem 1.

**Lemma 1.** Let \( S' \) be defined as above, then minimizing \( f \) over the feasible region \( S' \) is equivalent to finding the point \( k \) from the set \( S' \cap C \) that is farthest away from \( H \), i.e.,

\[
\arg \min_{k \in S'} f(k) = \arg \max_{k \in (S' \cap C)} \langle k, u \rangle.
\]  

(18)

In addition, we can replace \( S' \) with \( B' \) in the above equation, and the optimal solution of \( f(k) \) does not change. In other words, when the feasible region is \( B' \), the optimal solution can be always obtained at the surface of \( B' \). Thus, the following equation holds:

\[
\arg \min_{k \in S'} f(k) = \arg \min_{k \in B'} f(k) = \arg \max_{k \in (S' \cap C)} \langle k, u \rangle = \arg \max_{k \in (B' \cap C)} \langle k, u \rangle.
\]  

(19)

**Proof.** By simplifying the original problem to a two-dimensional plane, the proof of Lemma 1 will be readily apparent. Let \( V := \text{span}(\{x, u\}) \) be the plane spanned by \( x \) and \( u \). It is easy to observe \( B, C, B', S' \) are symmetrical about the plane \( V \). Next, we will show that for any \( k \in B' \), there must exist a point \( k^* \in S' \cap V \) such that \( f(k^*) \leq f(k) \). To find the \( k^* \) that satisfies the condition, we introduce the notation \( \Pi_V : \mathbb{R}^n \mapsto V \) to denote the projection from \( \mathbb{R}^n \) to \( V \).

Now, take any \( k \in B' \), and use \( k'' \) to denote the mirror point of \( k \) with respect to \( V \), as shown in Fig. 8. The projection point \( \Pi_V(k) \) is the midpoint of the line between \( k \) and \( k'' \), i.e., \( k'' = 2\Pi_V(k) - k \).

---

\(^4\)Note that \( k \) defined here may not be a tangent point on the ball.
The angle between \( \theta \) and \( u \) function is \( f \). Hence, to the definition of the convex cone \( C \) that \( \| y \| \) is fixed. Therefore, the only way to minimize \( f \) is to maximize \( \theta \). According to the definition of the convex cone \( C_0 \), \( u \) is outside \( C_0 \). Notice that \( x \in C_0 \) and \( k' \in \Pi_V(B') \), hence \( k' - x \) is in the convex cone \( \Pi_V(C_0) \). Therefore, based on the positions of the two vectors \( u \) and \( k' - x \) with respect to the cone \( \Pi_V(C_0) \), we conclude that \( \theta_1 \geq \theta_2 \). In such case, the distance function is \( f(k') = \| y_{k'} - x \| = |(x, u)| / \cos(\theta_1 - \theta_2) \), as shown in Fig. 9a. Because both \( x \) and \( u \) are fixed, the value of \( \theta_1 \) is fixed. Therefore, the only way to minimize \( f(k') \) is to maximize \( \theta_2 \). Among all possible choices of \( k' \) in \( \Pi_V(B') \), the \( k' \) that maximizes the angle \( \theta_2 \) appears on...
we can still conclude that
\[
\arg \min_x f(x) \geq \theta
\]
Therefore, in the first case, \(\arg \min \theta \leq \arg \max f(x)\). For fair comparison, we set \(\theta = R\) into the PyTorch version for the experiments. In the experiments of targeted attacks, we randomly select an image from the target class as the initial adversarial example. For fair comparison, we set

\[
\text{HSJA} \quad \text{Hyperparameters of HSJA} \quad \text{are listed in Table 4. We translate the implementation code into the PyTorch version for the experiments. In the experiments of targeted attacks, we randomly select an image from the target class as the initial adversarial example. For fair comparison, we set}
\]

\[
\text{C Experimental Settings}
\]

In this section, we provide the hyperparameter settings of the compared methods, \textit{i.e.}, HSJA [7], BA [4], Sign-OPT [11], and SVM-OPT [11].

\textbf{Experimental Equipment.} The experiments of all compared methods are conducted by using PyTorch 1.7.1 framework on a NVIDIA 1080Ti GPU.

\textbf{HSJA.} Hyperparameters of HSJA [7] are listed in Table 4. We translate the implementation code into the PyTorch version for the experiments. In the experiments of targeted attacks, we randomly select an image from the target class as the initial adversarial example. For fair comparison, we set
Table 4: The hyperparameters of HSJA.

| Dataset  | Hyperparameter                                      | Value                  |
|----------|-----------------------------------------------------|------------------------|
| CIFAR-10 | $\gamma$, threshold of the binary search            | 1.0                    |
|          | $B_0$, the initial batch size for gradient estimation| 100                    |
|          | $B_{\text{max}}$, the maximum batch size for gradient estimation | 10,000 geometric progression |
|          | the search method for step size                     |                        |
|          | number of iterations                                | 64                     |
| ImageNet | $\gamma$, threshold of the binary search            | 1,000.0                |
|          | $B_0$, the initial batch size for gradient estimation| 100                    |
|          | $B_{\text{max}}$, the maximum batch size for gradient estimation | 10,000 geometric progression |
|          | the search method for step size                     |                        |
|          | number of iterations                                | 64                     |

Table 5: The hyperparameters of BA.

| Hyperparameter                                      | Value                  |
|-----------------------------------------------------|------------------------|
| maximum number of trials per iteration              | 25                     |
| number of iterations                                | 1,200                  |
| spherical step size                                 | 0.01                   |
| source step size                                    | 0.01                   |
| step size adaptation multiplier                     | 1.5                    |
| disable automatic batch size tuning                 | False                  |
| generate candidates and random numbers without using multithreading | False |

Table 6: The hyperparameters of Sign-OPT.

| Hyperparameter                                      | Value                  |
|-----------------------------------------------------|------------------------|
| $k$, number of queries for estimating an approximate gradient | 200                    |
| $\alpha$, the update step size of the direction $\theta$ | 0.2                    |
| $\beta$, used for the gradient estimation of $\theta$ and determining the stopping threshold of binary search the number of iterations | 0.001 | 1,000 |
| the binary search’s stopping threshold of the CIFAR-10 dataset | $\frac{\beta}{\alpha}$ |
| the binary search’s stopping threshold of the ImageNet dataset | $\times 10^{-4}$ |

Table 7: The hyperparameters of SVM-OPT.

| Hyperparameter                                      | Value                  |
|-----------------------------------------------------|------------------------|
| $k$, number of queries for estimating gradients     | 100                    |
| $\alpha$, the step size of the gradient descent of $\theta$ | 0.2                    |
| $\beta$, used for the gradient estimation of $\theta$ and determining the stopping threshold of binary search the number of iterations | 0.001 | 1,000 |
| the binary search’s stopping threshold of the CIFAR-10 dataset | $\frac{\beta}{\alpha}$ |
| the binary search’s stopping threshold of the ImageNet dataset | $\times 10^{-4}$ |

the hyperparameters of TA and G-TA to be the same with HSJA, i.e., the same initial batch size $B_0$ and the same $\gamma$.

**BA.** Hyperparameters of BA [4] are listed in Table 5. In the experiments, we directly use the implementation of BA from Foolbox 2.0 [35, 36], and adopt a randomly selected image from the target class as the initialization in the targeted attack.

**Sign-OPT and SVM-OPT.** Hyperparameters of Sign-OPT [11] and SVM-OPT [11] are listed in Tables 6 and 7. We translate the implementation code into the PyTorch version for the experiments. In the experiments of targeted attacks, we set the initial direction $\theta_0$ of Sign-OPT and SVM-OPT to the direction of a randomly selected image of the target class.
## D Experimental Results

### D.1 Limitation of Tangent Attack

The proposed approach supports all types of attacks, including both untargeted and targeted attacks under the both $\ell_2$ and $\ell_\infty$ norm constraints. This is the strength of the proposed approach. However, in the $\ell_\infty$ norm attack, TA and G-TA obtain the similar performance to the baseline method HSJA. Because under the definition of the $\ell_\infty$ norm distance: $D_{\ell_\infty}(x, y) := \max_i(|x_i - y_i|), i \in \{1, \ldots, d\}$ ($d$ is the image dimension), the intersection of the tangent line and the decision boundary may not be the one with the shortest $\ell_\infty$ norm distance to the original image. Therefore, searching the boundary sample along the tangent line cannot always outperform HSJA in the $\ell_\infty$ norm attack.

Tables 8 and 9 demonstrate the experimental results of attacking against undefined models on the CIFAR-10 and ImageNet datasets.

**Table 8:** Mean $\ell_\infty$ distortions of different query budgets on the ImageNet dataset, where the radius ratio $r$ is set to 1.1 in G-TA. BA is not applicable to the $\ell_\infty$ norm attack, hence it is not listed.

| Target Model | Method       | Targeted Attack | Untargeted Attack |
|--------------|--------------|-----------------|-------------------|
|              | @300 @1K    | @2K @5K @8K @10K | @300 @1K @2K @5K @8K @10K |
| Sign-OPT [11]| 0.557 0.519 | 0.481 0.421 0.390 0.375 | 1.078 0.792 0.548 0.328 0.262 0.239 |
| SVM-OPT [11] | 0.558 0.512 | 0.476 0.423 0.397 0.385 | 1.079 0.763 0.526 0.336 0.280 0.260 |
| Inception-v3 | HSJA [7]    | 0.370 0.330 0.289 0.211 0.169 0.147 | 0.305 0.236 0.174 0.093 0.069 0.059 |
|              | TA          | 0.370 0.330 0.291 0.216 0.172 0.149 | 0.304 0.234 0.173 0.093 0.068 0.059 |
|              | G-TA        | 0.364 0.326 0.289 0.220 0.179 0.159 | 0.304 0.238 0.174 0.093 0.068 0.059 |
| Inception-v4 | HSJA [7]    | 0.357 0.324 0.287 0.215 0.175 0.152 | 0.336 0.257 0.185 0.091 0.060 0.048 |
|              | TA          | 0.354 0.328 0.294 0.221 0.182 0.161 | 0.337 0.264 0.196 0.103 0.073 0.062 |
|              | G-TA        | 0.354 0.324 0.290 0.220 0.182 0.162 | 0.337 0.264 0.196 0.104 0.073 0.061 |
| SENet-154    | HSJA [7]    | 0.347 0.288 0.249 0.176 0.139 0.119 | 0.253 0.195 0.141 0.071 0.048 0.039 |
|              | TA          | 0.346 0.289 0.251 0.181 0.142 0.123 | 0.253 0.196 0.141 0.071 0.047 0.038 |
|              | G-TA        | 0.344 0.288 0.252 0.179 0.142 0.123 | 0.253 0.196 0.141 0.070 0.047 0.038 |
| ResNet-101   | HSJA [7]    | 0.349 0.301 0.250 0.200 0.170 0.149 | 0.345 0.270 0.205 0.133 0.091 0.071 |
|              | TA          | 0.340 0.282 0.246 0.180 0.143 0.125 | 0.340 0.266 0.196 0.109 0.064 0.050 |
|              | G-TA        | 0.337 0.280 0.246 0.182 0.150 0.132 | 0.340 0.270 0.204 0.133 0.091 0.071 |

**Table 9:** Mean $\ell_\infty$ distortions of different query budgets on the CIFAR-10 dataset, where the radius ratio $r$ is set to 1.5 in G-TA. BA is not applicable to the $\ell_\infty$ norm attack, and thus it is not listed.

| Target Model | Method       | Targeted Attack | Untargeted Attack |
|--------------|--------------|-----------------|-------------------|
|              | @300 @1K    | @2K @5K @8K @10K | @300 @1K @2K @5K @8K @10K |
| Sign-OPT [11]| 0.395 0.318 | 0.237 0.134 0.096 0.082 | 0.284 0.189 0.115 0.059 0.047 0.043 |
| SVM-OPT [11] | 0.390 0.299 | 0.226 0.134 0.099 0.087 | 0.286 0.173 0.104 0.059 0.049 0.046 |
| PyramidNet-272| HSJA [7]    | 0.218 0.155 0.112 0.057 0.039 0.032 | 0.133 0.056 0.034 0.016 0.012 0.011 |
|              | TA          | 0.219 0.154 0.112 0.057 0.039 0.032 | 0.134 0.057 0.035 0.017 0.013 0.011 |
|              | G-TA        | 0.218 0.153 0.113 0.057 0.039 0.031 | 0.133 0.056 0.034 0.016 0.012 0.010 |
| GDAS         | HSJA [7]    | 0.398 0.352 0.266 0.153 0.107 0.089 | 0.305 0.269 0.231 0.183 0.167 0.160 |
|              | TA          | 0.214 0.151 0.115 0.062 0.041 0.032 | 0.099 0.030 0.020 0.012 0.010 0.009 |
|              | TA          | 0.186 0.107 0.070 0.031 0.021 0.018 | 0.049 0.029 0.020 0.011 0.009 0.008 |
| WRN-28       | HSJA [7]    | 0.194 0.111 0.072 0.032 0.022 0.019 | 0.284 0.208 0.125 0.051 0.042 0.039 |
|              | TA          | 0.195 0.112 0.073 0.032 0.022 0.019 | 0.082 0.029 0.020 0.012 0.010 0.009 |
| WRN-40       | HSJA [7]    | 0.397 0.305 0.220 0.120 0.085 0.073 | 0.284 0.208 0.125 0.051 0.042 0.039 |
|              | TA          | 0.381 0.293 0.220 0.126 0.092 0.079 | 0.273 0.190 0.120 0.057 0.045 0.041 |

The results of Tables 8 and 9 show that HSJA, TA and G-TA obtain the similar average $\ell_\infty$ distortions. Therefore, although the proposed approach is applicable to $\ell_\infty$ norm attack, the performance is similar to that of the baseline method HSJA.
D.2 Experimental Results of Attacks against Defense Models

We also conduct experiments by using $\ell_\infty$ norm attacks to break five defense models, and the experimental results are shown in Table 10. The conclusion drawn from this table is the same as that in Tables 8 and 9: TA and G-TA obtain the similar performance with HSJA in $\ell_\infty$ norm attacks.

Table 10: The experimental results of performing $\ell_\infty$ norm attacks against the defense models on the CIFAR-10 dataset, where the radius ratio $r$ is set to 1.5 in G-TA.

| Target Model       | Method          | Untargeted Attack @300 | @1K | @2K | @5K | @8K | @10K |
|--------------------|-----------------|-------------------------|-----|-----|-----|-----|------|
| AT [30]            | Sign-OPT [11]   | 0.731 0.519 0.395 0.288 0.255 0.243 |     |     |     |     |      |
|                    | SVM-OPT [11]    | 0.719 0.498 0.382 0.287 0.261 0.251 |     |     |     |     |      |
|                    | HSJA [7]        | 0.181 0.145 0.121 0.090 0.080 0.075 |     |     |     |     |      |
|                    | TA              | 0.184 0.147 0.121 0.090 0.079 0.075 |     |     |     |     |      |
|                    | G-TA            | 0.181 0.145 0.121 0.090 0.080 0.075 |     |     |     |     |      |
| TRADES [46]        | Sign-OPT [11]   | 0.748 0.562 0.419 0.304 0.269 0.257 |     |     |     |     |      |
|                    | SVM-OPT [11]    | 0.743 0.534 0.409 0.306 0.281 0.271 |     |     |     |     |      |
|                    | HSJA [7]        | 0.194 0.162 0.137 0.106 0.095 0.090 |     |     |     |     |      |
|                    | TA              | 0.195 0.163 0.138 0.107 0.095 0.090 |     |     |     |     |      |
|                    | G-TA            | 0.194 0.163 0.138 0.107 0.095 0.090 |     |     |     |     |      |
| JPEG [18]          | Sign-OPT [11]   | 0.301 0.292 0.281 0.262 0.250 0.243 |     |     |     |     |      |
|                    | SVM-OPT [11]    | 0.301 0.288 0.275 0.256 0.249 0.246 |     |     |     |     |      |
|                    | HSJA [7]        | 0.094 0.086 0.078 0.066 0.061 0.058 |     |     |     |     |      |
|                    | TA              | 0.093 0.087 0.080 0.067 0.061 0.058 |     |     |     |     |      |
|                    | G-TA            | 0.097 0.091 0.081 0.068 0.062 0.059 |     |     |     |     |      |
| Feature Distillation [28] | Sign-OPT [11]   | 0.344 0.330 0.317 0.290 0.273 0.255 |     |     |     |     |      |
|                    | SVM-OPT [11]    | 0.354 0.338 0.323 0.297 0.284 0.279 |     |     |     |     |      |
|                    | HSJA [7]        | 0.090 0.087 0.080 0.069 0.064 0.061 |     |     |     |     |      |
|                    | TA              | 0.089 0.086 0.079 0.070 0.063 0.060 |     |     |     |     |      |
|                    | G-TA            | 0.090 0.086 0.079 0.067 0.062 0.059 |     |     |     |     |      |
| Feature Scatter [45] | Sign-OPT [11]   | 0.561 0.580 0.246 0.135 0.110 0.101 |     |     |     |     |      |
|                    | SVM-OPT [11]    | 0.550 0.544 0.222 0.137 0.116 0.110 |     |     |     |     |      |
|                    | HSJA [7]        | 0.202 0.137 0.104 0.062 0.048 0.042 |     |     |     |     |      |
|                    | TA              | 0.202 0.137 0.104 0.062 0.048 0.042 |     |     |     |     |      |
|                    | G-TA            | 0.205 0.139 0.105 0.062 0.048 0.042 |     |     |     |     |      |

Next, we conduct experiments by using $\ell_2$ norm attack to break different defense models on the CIFAR-10 and ImageNet datasets. In the CIFAR-10 dataset, we select six types of defense models:

- Adversarial Training (AT) [30]: the most effective defense method, which uses adversarial examples as the training data to obtain the robust classifier.
- TRADES [46]: an improved AT that optimizes a regularized surrogate loss.
- JPEG [18]: a standard image compression algorithm based on the discrete cosine transform, which can remove the adversarial perturbations, thereby providing some degree of defense.
- Feature Distillation [28]: a defense method based on the improved JPEG image compression. Its defense mechanism is divided into two steps. Firstly, it filters out adversarial perturbations by using a semi-analytical method. Secondly, it restores the classification accuracy of benign images by using a DNN-oriented quantization process.
- Feature Scatter [45]: a feature scattering-based AT method, which is an unsupervised approach for generating adversarial examples during the training.
- ComDefend [23]: a defense model that consists of a compression CNN and a reconstruction CNN to transform the adversarial image into its clean version to defend against attacks.

In the ImageNet dataset, we directly use the publicly available AT models for experiments, all of which use the ResNet-50 networks as their backbones. The pre-trained weights can be downloaded from https://github.com/MadryLab/robustness. In the experiments, we set the radius ratio $r$ of G-TA to 1.5, and the experimental results are shown in Fig. 10. In untargeted attacks (Figs. 10a, 10b, 10c), the G-TA (the semi-ellipsoid version) outperforms the TA (the hemisphere version), and the baseline method HSJA outperforms TA and G-TA. We conjecture that it is because the classification decision boundaries of the AT models on the ImageNet dataset are extremely curved in untargeted attacks, resulting in the better performance of HSJA. In targeted attacks (Figs. 10d, 10e, 10f), both TA and G-TA outperform HSJA in the attacks of different AT models. These results indicate that TA and G-TA are more suitable for the targeted attack. Another interesting finding is that SVM-OPT
performs better in untargeted attacks while Sign-OPT performs better in targeted attacks. We will explore the reasons for these results in the future work.

Figs. 11 and 12 show the experimental results of untargeted and targeted attacks on the CIFAR-10 dataset, respectively. In the results of untargeted attacks (Fig. 11), G-TA outperforms HSJA and TA...
in the attacks of ComDefend, JPEG and Feature Distillation. When the target models are AT, Feature Scatter and TRADES, the performance of G-TA is similar to that of the baseline attack method HSJA.

In addition, in the experimental results of targeted attacks (Fig. 12), the performance of G-TA is similar to that of TA when attacking different defense models.

D.3 Distributions of Distortions across Different Adversarial Examples

So far, all the experimental results only show the average $\ell_2$ distortion of 1,000 adversarial examples. To check the distortion of each adversarial example in more detail, we extract the $\ell_2$ distortions of 20 samples from HSJA, TA and G-TA. These samples are selected from 1,000 images in the following way: from the 1st image to the 1,000th image, we select one image for every 50 images. Fig. 13 shows the distributions of $\ell_2$ distortions across 20 adversarial examples on the ImageNet dataset, where the 1st image’s “image number index” is 0. The results indicate that the $\ell_2$ distortions obtained by TA and G-TA are uniformly better than that of the baseline method HSJA. Thus, our approach can obtain better $\ell_2$ distortions on different adversarial examples, not just on specific samples.

D.4 Experimental Results of Median Distortions

In this section, we report the median $\ell_2$ distortions of different query budgets on the CIFAR-10 and ImageNet datasets. Tables 11 and 12 show the experimental results. We can draw the following conclusions based on the results.

(1) TA and G-TA perform better in attacking high-resolution images, i.e., the images of the ImageNet dataset. The median $\ell_2$ distortions of Table 11 are larger than that of Table 12, because the high-resolution images of the ImageNet dataset lead to larger $\ell_2$ distortions.

(2) TA is more effective in the targeted attacks. We speculate that it is because the adversarial region of the target class is narrower and more scattered in the targeted attack, resulting in a smoother decision boundary. Thus, TA is more suitable for targeted attacks.
Figure 13: Comparisons of $\ell_2$ distortions across 20 adversarial examples in targeted attacks of the ImageNet dataset.
### Table 11: Median $\ell_2$ distortions of different query budgets on the ImageNet dataset.

```
| Target Model Method | Targeted Attack | | | | |
|---------------------|----------------|---|---|---|---|
|                     | @300 | @1K | @2K | @5K | @8K | @10K |
| BA [4]              | 105.513 | 101.877 | 101.056 | 97.481 | 81.269 | 73.524 |
| SVM-OPT [11]        | 96.905 | 83.215 | 66.601 | 43.350 | 31.036 | 25.350 |
| Inception-v3        | 93.649 | 77.838 | 63.631 | 42.897 | 31.838 | 26.322 |
| HSIA [7]            | 106.341 | 92.114 | 79.225 | 47.469 | 30.624 | 23.838 |
| TA                  | 96.612 | 75.610 | 62.573 | 38.226 | 25.892 | 19.993 |
| G-TA                | 97.449 | 75.499 | 62.484 | 38.886 | 26.004 | 20.091 |
| Target Model Method | Untargeted Attack | | | | |
|                     | @300 | @1K | @2K | @5K | @8K | @10K |
| BA [4]              | - | - | - | - | - | - |
| SVM-OPT [11]        | 115.140 | 73.319 | 38.327 | 12.761 | 8.277 | 6.808 |
| Inception-v3        | 114.879 | 59.343 | 30.627 | 12.085 | 8.552 | 7.025 |
| HSIA [7]            | 105.702 | 53.880 | 32.684 | 12.360 | 7.829 | 6.227 |
| TA                  | 95.302 | 50.878 | 31.833 | 11.921 | 7.464 | 6.030 |
| G-TA                | 96.410 | 50.985 | 31.176 | 11.861 | 7.549 | 6.087 |
```

Table 12: Median $\ell_2$ distortions of different query budgets on the CIFAR-10 dataset.

```
| Target Model Method | Targeted Attack | | | | |
|---------------------|----------------|---|---|---|---|
|                     | @300 | @1K | @2K | @5K | @8K | @10K |
| BA [8]              | 8.240 | 7.711 | 7.697 | 6.013 | 3.938 | 3.068 |
| SVM-OPT [11]        | 7.900 | 6.050 | 3.796 | 1.441 | 0.762 | 0.459 |
| Inception-v3        | 8.780 | 7.432 | 4.199 | 1.651 | 0.894 | 0.565 |
| HSIA [7]            | 7.616 | 4.013 | 2.109 | 0.599 | 0.384 | 0.235 |
| TA                  | 7.650 | 3.874 | 2.071 | 0.599 | 0.380 | 0.218 |
| G-TA                | 7.452 | 3.980 | 2.110 | 0.602 | 0.387 | 0.234 |
| Target Model Method | Untargeted Attack | | | | |
|                     | @300 | @1K | @2K | @5K | @8K | @10K |
| BA [8]              | - | - | - | - | - | - |
| SVM-OPT [11]        | - | - | - | - | - | - |
| G-TA                | - | - | - | - | - | - |
```
