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
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 EECS, Peking University, Beijing, China
mac16@mails.tsinghua.edu.cn, 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 hyperparameters and 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 [32, 6, 29] 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 [25, 37, 58] 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, 28] 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 is not applicable because the public service returns only the top-1 predicted label (i.e., the hard

*Corresponding author.

HSJA updates $x_{t-1}$ along the gradient direction 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 $x$.

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 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 $x$.

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 [24], qFool [26], and Policy Driven Attack (PDA) [40] 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 subspace. 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 move to 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, 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 [23] 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 [1–3, 9, 12, 22, 28, 31] 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 constraint because it is difficult to find a suitable direction to reach the region of the target class.

The second category starts from a large perturbation or an image of the target class, and then reduces its distortion while staying in adversarial subspace, thereby gradually making it closer to the original image. BA [4] and NES [1] are two representative studies, but their query complexities are high. 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 binary search to achieve state-of-the-art query efficiency. HSJA can be used as the baseline of hard-label attacks. QEB [24] improves HSJA by using dimension reduction techniques. SurFree [30] and GeoDA [33] improve 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 [40] 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 [34] 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 benign example’s subspace 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:

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

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:

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

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 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 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 [7, 33]. 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 Fig. 2. 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 in \( B \), \( u \) 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 \( H \), \( u \), \( k \), \( x \), and \( x_{t-1} \) be defined above, then the distance \( \| x - x_t \| \) is minimized if \( k \) is the optimal solution of the following constrained optimization problem:

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

s.t.

\[
\langle k - x_{t-1}, x - k \rangle = 0, \quad (4)
\]

\[
\| k - x_{t-1} \| = R, \quad (5)
\]

\[
\langle k - x_{t-1}, u \rangle \geq 0. \quad (6)
\]

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 \( \Pi_H : \mathbb{R}^n \to H \) 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

\[
(x, u) = |x| \cdot |u| \cdot \cos(\frac{\pi}{2} + \alpha) = |x| \cdot |u| \cdot (\sin \alpha).
\]

(7)

Then we have $\sin \alpha = -\frac{(x, u)}{|x| \cdot |u|}$ and $\cos \alpha = \sqrt{1 - \sin^2 \alpha} = \sqrt{|x|^2 - (x, u)^2}$. By 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 $x' = (x, u) \cdot (\frac{-u}{||u||})^2 = (x, u) \cdot \frac{u}{||u||}$. Then, because $k'$ and $x - x'$ are on the same direction, we have

\[
k' = \frac{x - x'}{||x - x'||} \cdot ||k'||.
\]

(9)

Now, $||k'|| = R \cdot \cos \gamma$ and $x' = (x, u) \cdot \frac{u}{||u||}^2$ are plugged into Eq. (7), and $k'$ is obtained as

\[
k' = \frac{x - x'}{||x - x'||} \cdot R \cdot \cos \gamma.
\]

(10)

Therefore, $k$ can be derived as

\[
k = k' + h \cdot u = \frac{x - (x, u) \cdot \frac{u}{||u||}^2}{||x - (x, u) \cdot \frac{u}{||u||}||} \cdot R \cdot \cos \gamma + h \cdot u.
\]

(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 \( x_k \) 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, \( u \) and \( v := (x - (x, u) \cdot u / \|u\|^2) / \|x - (x, u) \cdot u / \|u\|^2\) form an orthogonal
basis of the plane \( V \), then \( x \) can be identified with coordinates \((x_0, z_0)\), i.e., \( x = x_0 v + z_0 u \).
Let \( \theta \) denote the angle between the vector \( x \) and the vector \(-u\), i.e., \( \theta = \arccos(\frac{(x, u)}{\|x\| \cdot \|u\|}) \). 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 \( k = x_k v + z_k u \),
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:

\[
\frac{2x_k}{S^2} + \frac{2z_k}{L^2} \cdot \frac{dz}{dx} \bigg|_{x=x_k} = 0. \tag{12}
\]

Thus, we have the slope of tangent line at \( k \) be \( \frac{dz}{dx} \bigg|_{x=x_k} = -\frac{x_k L^2}{z_k S^2} \). Therefore, the tangent line
can be written as \( z - z_k = -\frac{x_k L^2}{z_k S^2} (x - x_k) \). Since the tangent line passes through \( x \), we know
Algorithm 1

Tangent Attack

TA and G-TA can be combined into one algorithm process, which is shown in Algorithm 1. It first

version). So we fix

subspace. Finally, Algorithm 1 uses the binary search method to map

reasonable radius

while

using more samples can reduce the estimation error. Then, a

of gradient estimation in the later iterations has a greater impact on the attack performance, so

B

to estimate the gradient, which is increased to

example that is close to

x

attack success rate is always 100%. Then, it performs a

x

search step always maps any

˜

B

to the adversarial side of

x

Fig. 4b. Apparently, the solution of the optimal tangent point should satisfy

z

In Eq. (13), the known variables are

L
,S
,x
, and

z
, and the unknown variables that we need to solve for are

x
,k
, and

z
. 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
> 0, so the one of two solutions in which

z
> 0 should be picked:

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

Finally, the optimal tangent point

k
is obtained via

k = |x_k| \cdot \frac{x - (\langle x, u \rangle / \|u\|^2)}{x - (\langle x, u \rangle / \|u\|^2)} + 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 and G-TA 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 subspace. 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

\textbf{Input}: benign image

x,
attack success indicator function

\phi(\cdot)
defined in Eq. (1), initial batch size

B_0,
itration

T,
mode

m \in \{\text{semi-ellipsoid}, \text{hemisphere}\},
radius ratio

r.

Initialize

x_0
that satisfies

\phi(x_0) = 1;

\begin{align*}
x_0 &\leftarrow \text{BinarySearch}(x_0, x, \phi); \quad \triangleright \text{boundary search} \\
d_0 &\leftarrow ||x_0 - x||;
\end{align*}

for

i \in \{1, \ldots, T\}
do

\begin{align*}
    &\quad \text{Sample } B_t \leftarrow B_0 \sqrt{T} \text{ random vectors to estimate the gradient } u; \\
    &\quad \text{Initialize } R \leftarrow d_{t-1} / \sqrt{T}; \quad \triangleright \text{the initial radius} \\
    &\quad \text{while true do} \\
    &\quad \quad \text{Compute the optimal tangent point } k \text{ based on Eq. (11) if } m = \text{hemisphere else Eq. (14)}; \\
    &\quad \quad R \leftarrow \frac{R}{2}; \quad \triangleright \text{search the radius, and we set } L = R, S = \frac{L}{r} \text{ if } m = \text{semi-ellipsoid} \\
    &\quad \quad \text{if } \phi(k) = 1 \text{ then} \\
    &\quad \quad \quad \text{break;} \\
    &\quad \quad \text{end if} \\
    &\quad \text{end while} \\
    &\quad k \leftarrow \text{Clip}(k, 0, 1); \\
    &\quad x_t \leftarrow \text{BinarySearch}(k, x, \phi); \quad \triangleright \text{boundary search} \\
    &\quad d_t \leftarrow ||x_t - x||;
\end{align*}

end for
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 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 [24] 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 from the PyTorch-Classification package: (1) a 272-layer PyramidNet+ShakeDrop network (PyramidNet-272) [19, 39] 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) [41]; and (4) a wide residual network with 40 layers (WRN-40) [41]. In the ImageNet dataset, we select four target models from the pre-trained models package: (1) Inception-v3 [36], (2) Inception-v4 [35], (3) ResNet-101 [20], and (4) SENet-154 [21].

Evaluation Metric. Following previous studies [40], 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 subspace in the target attack 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.

---

3Pre-trained weights: https://github.com/Cadene/pretrained-models.pytorch.
Table 1: Mean $\ell_2$ distortions of different query budgets on the ImageNet dataset, where $r = 1.1$.

<table>
<thead>
<tr>
<th>Target Model</th>
<th>Method</th>
<th>@300</th>
<th>@1K</th>
<th>@2K</th>
<th>@5K</th>
<th>@8K</th>
<th>@10K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Targeted Attack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BA</td>
<td></td>
<td>111.798</td>
<td>108.044</td>
<td>106.283</td>
<td>102.715</td>
<td>86.931</td>
<td>78.326</td>
</tr>
<tr>
<td>Sign-OPT</td>
<td></td>
<td>103.937</td>
<td>87.706</td>
<td>71.291</td>
<td>46.744</td>
<td>34.640</td>
<td>29.414</td>
</tr>
<tr>
<td>SVM-OPT</td>
<td></td>
<td>101.630</td>
<td>82.950</td>
<td>67.965</td>
<td>42.275</td>
<td>35.694</td>
<td>31.106</td>
</tr>
<tr>
<td>Inception-v3</td>
<td></td>
<td>111.562</td>
<td>95.295</td>
<td>82.111</td>
<td>52.544</td>
<td>37.395</td>
<td>30.425</td>
</tr>
<tr>
<td>TA</td>
<td></td>
<td>103.781</td>
<td>80.327</td>
<td>66.708</td>
<td>42.121</td>
<td>30.846</td>
<td>25.566</td>
</tr>
<tr>
<td>G-TA</td>
<td></td>
<td>103.724</td>
<td>81.089</td>
<td>67.168</td>
<td>42.434</td>
<td>31.011</td>
<td>25.587</td>
</tr>
<tr>
<td></td>
<td>Untargeted Attack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>107.558</td>
<td>102.309</td>
<td>95.776</td>
<td>78.668</td>
<td>60.296</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Target Model</th>
<th>Method</th>
<th>@300</th>
<th>@1K</th>
<th>@2K</th>
<th>@5K</th>
<th>@8K</th>
<th>@10K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Targeted Attack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BA</td>
<td></td>
<td>81.090</td>
<td>77.725</td>
<td>76.112</td>
<td>71.981</td>
<td>53.933</td>
<td>47.952</td>
</tr>
<tr>
<td>Sign-OPT</td>
<td></td>
<td>75.722</td>
<td>62.876</td>
<td>49.191</td>
<td>30.155</td>
<td>21.333</td>
<td>17.672</td>
</tr>
<tr>
<td>SVM-OPT</td>
<td></td>
<td>74.658</td>
<td>58.677</td>
<td>46.827</td>
<td>30.264</td>
<td>22.461</td>
<td>19.186</td>
</tr>
<tr>
<td>Inception-v3</td>
<td></td>
<td>77.035</td>
<td>63.488</td>
<td>51.802</td>
<td>30.198</td>
<td>16.620</td>
<td>12.631</td>
</tr>
<tr>
<td>TA</td>
<td></td>
<td>70.739</td>
<td>55.256</td>
<td>43.604</td>
<td>24.961</td>
<td>16.756</td>
<td>13.876</td>
</tr>
<tr>
<td>G-TA</td>
<td></td>
<td>70.591</td>
<td>55.224</td>
<td>44.407</td>
<td>25.041</td>
<td>16.854</td>
<td>14.047</td>
</tr>
<tr>
<td></td>
<td>Untargeted Attack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>64.007</td>
<td>60.389</td>
<td>56.644</td>
<td>44.175</td>
<td>31.373</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Target Model</th>
<th>Method</th>
<th>@300</th>
<th>@1K</th>
<th>@2K</th>
<th>@5K</th>
<th>@8K</th>
<th>@10K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Targeted Attack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception-v3</td>
<td></td>
<td>100.295</td>
<td>79.604</td>
<td>63.621</td>
<td>35.194</td>
<td>22.773</td>
<td>18.414</td>
</tr>
<tr>
<td>Inception-v4</td>
<td></td>
<td>104.490</td>
<td>75.622</td>
<td>59.836</td>
<td>35.112</td>
<td>22.329</td>
<td>17.799</td>
</tr>
<tr>
<td>ResNet-154</td>
<td></td>
<td>101.845</td>
<td>73.347</td>
<td>62.357</td>
<td>35.375</td>
<td>22.367</td>
<td>18.376</td>
</tr>
<tr>
<td></td>
<td>Untargeted Attack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>35.227</td>
<td>22.773</td>
<td>18.414</td>
<td>17.799</td>
<td>18.376</td>
</tr>
</tbody>
</table>

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 [23]; (2) an
improved AT that optimizes a regularized surrogate loss, named as TRADES \cite{zhong2020fair}; (3) an image-transformation-based defense called JPEG \cite{zhang2017color}; and (4) a DNN-oriented compression defense called Feature Distillation \cite{he2019soft}. All defense models adopt ResNet-50 as the backbone. Previous studies \cite{wang2019query} have shown that AT and TRADES have the issue of robust overfitting, which leads to a significant increase in the curvature of the classification decision boundary. Fig. 5b shows that G-TA performs best in attacking TRADES. This superiority is also demonstrated in attacks on other defenses (e.g., Fig. 5d), proving the effectiveness of G-TA in attacking the models with curved decision boundaries.

### 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. 6b 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 to 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


