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

Anonymous Author(s)
Affiliation
Address
email

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 accessible. In this paper, we propose a novel geometric-based approach called Tangent Attack, which identifies an optimal tangent point of a virtual hemisphere around the current adversarial example to reduce the distortion of the attack. Assuming the decision boundary is locally flat, we theoretically prove that the minimal ℓ_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 that replaces the hemisphere with a semi-ellipsoid, so as to adapt to curved decision boundaries. Our approach is free of hyperparameters and pretraining. Extensive experiments conducted on ImageNet and CIFAR-10 datasets demonstrate that Tangent Attack can consume only a small number of queries to achieve the low-magnitude distortion.

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 to the target model, *i.e.,* white-box and black-box attacks. Many white-box attacks [32, 4, 29] have been proposed, and they can access the gradients w.r.t. the target model's inputs to generate adversarial examples with first-order optimization techniques. By 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, 13] generate adversarial examples using a surrogate model and then transfer them to the target model. Although there is no need to query the target model in these attacks, the attack success rate cannot be guaranteed, especially in the case of targeted attacks, where the target model and the surrogate model are usually very different. 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 [28, 23, 11] 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.,* hard label) rather than the predicted score. In this case, since the feedback information is limited and the objective function is discrete and discontinuous, the attack requires solving a high-dimensional combinatorial optimization problem, which is often challenging.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

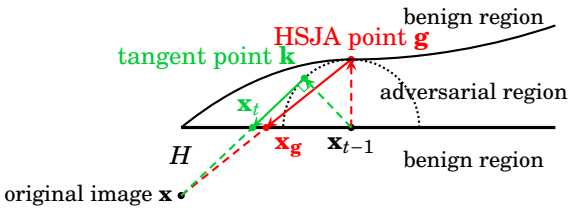


Figure 1: Simplified two-dimensional illustration of our motivation. H is the decision boundary, and $\mathbf{x}_{t-1}$ is the current adversarial example mapped onto the decision boundary at the $(t-1)$-th iteration. HSJA first updates $\mathbf{x}_{t-1}$ along the gradient direction and then maps it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

To reformulate the attack as a real-valued continuous optimization problem, OPT [8], Sign-OPT [10], and RayS [5] 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 θ. However, when attacking complex models, it may become very difficult to find a suitable direction θ along which adversarial examples exist.

Boundary Attack (BA) [2], HopSkipJumpAttack (HSJA) [6], QEBA [25], GeoDA [33], and Policy Driven Attack (PDA) [38] eliminate the search of the direction θ. Instead, they start from a large adversarial perturbation and then reduce its distortion while staying adversarial. 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, although Rahmati *et al.* claimed that GeoDA is grounded on the geometric properties of DNNs, it assumes that the decision boundary of DNNs is flat in the vicinity of data samples. GeoDA heavily relies on this assumption to estimate the gradient. Thus it cannot be applied to a complex target model with curved decision boundaries. Besides, GeoDA computes the next adversarial example by shifting the original image along the gradient estimated from the previous adversarial sample located at the boundary. Such algorithmic process is not applicable to the targeted attack. HSJA simply uses the gradient $\mathbf{u}$ estimated at the boundary as the direction of each movement, while ignoring a geometrically critical issue, *i.e.,* $\mathbf{u}$ is not the optimal direction to be followed (Fig. 1). We could explore even 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 $\mathbf{x}_{t-1}$ to indicate all possible locations wherein $\mathbf{x}_{t-1}$ can be moved 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 $\mathbf{x}$, thereby producing the adversarial example with the minimum distortion. In real attack scenarios, the image data resides in a high-dimensional space, and the semicircle becomes a semisphere. In this case, the benefit of tangent point 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, and we discover that the minimum distortion can be achieved by searching the optimal tangent point of a virtual hemisphere around the current boundary sample.
2. We propose a novel geometric-based method to obtain a closed-form solution of the optimal tangent point. We also provide a formal proof and an intuitive explanation of our approach.
3. To improve robustness, we further propose a generalized method that replaces the hemisphere with a semi-ellipsoid to adapt to target models with curved decision boundaries.
4. Extensive experiments conducted on 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, hard-label attacks). Score-based attacks [11, 23, 31, 28, 7, 22, 1] 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 accessed. The hard-label attacks usually fall into three categories.

The first category starts from the original image $\mathbf{x}_0$ and attempts to find a optimal direction θ to reach the adversarial example. OPT [8] searches an optimal θ to minimize the distance from $\mathbf{x}_0$ to the nearest adversarial example. Sign-OPT [9] improves the query efficiency of OPT by estimating the sign of the directional derivative by using a single query. RayS [5] proposes a fast check step to efficiently find the direction θ. However, RayS cannot support targeted attacks because it is difficult to find a ray direction pointing to 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 region, thereby gradually making it closer to the original image. BA [2] and NES [22] are two representative studies, but their query complexities are high. Biased BA [3] reinterprets BA as a biased sampling framework and incorporates different biases to improve the query efficiency. HSJA [6] utilizes the gradient estimation and binary search to achieve state-of-the-art query efficiency. HSJA can be considered the baseline of hard-label attacks. QEBA [25] 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 are still not fully explored, and they do not support the targeted attack. PDA [38] uses a reinforcement learning framework to train a policy network to predict promising directions. However, PDA requires a high-cost pretraining and training data collection, which are not always available in attack 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 modeling the local geometry and reducing the dimension of the search space with the stochastic coordinate selection. In contrast, we exploited the geometric characteristics of the decision boundary of DNNs. Our approach identifies that the optimal tangent point is crucial to minimize the distortion of the attack, thereby reducing the query complexity significantly.

3 The Proposed Tangent Attack

3.1 Adversary Setup

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

$$\phi(\mathbf{x}_{\text{adv}}) := \begin{cases} 1 & \text{if } \hat{y} = y_{\text{adv}} \text{ in the targeted attack,} \\ & \quad \text{or } \hat{y} \neq y \text{ in the untargeted attack} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $y \in \mathbb{R}$ is the true label of $\mathbf{x}$ and $y_{\text{adv}} \in \mathbb{R}$ is a predefined target class label. In this study, we focus on generating an adversarial example $\mathbf{x_{adv}}$ with $\phi(\mathbf{x}_{\text{adv}}) = 1$, such that the distortion $d(\mathbf{x_{adv}}, \mathbf{x}) = ||\mathbf{x}_{\text{adv}} - \mathbf{x}||_2$ is minimized. This goal can be formulated as the optimization problem:

$$\min_{\mathbf{x}_{\text{adv}}} d(\mathbf{x_{adv}}, \mathbf{x}) \quad \text{s.t. } \phi(\mathbf{x}_{\text{adv}}) = 1 \tag{2}$$

3.2 Motivation

Most recent attacks belong to the second category (Section 2) that follows a common procedure: it starts from an adversarial image yet not close enough to the benign one, then iteratively searches for a closer adversarial image. Let us take a typical attack of HSJA for example (Fig.

1). First, the attack initializes the adversarial example 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 binary search to map the initial sample onto the decision boundary H, denote as $\mathbf{x}_{t-1}$. Then, the algorithm iteratively performs three steps to update $\mathbf{x}_{t-1}$: **estimating gradient $\mathbf{u}$ at H** by sampling many probes around $\mathbf{x}_{t-1}$; **jumping to the point $\mathbf{g}$** along the direction $\mathbf{u}$ with the step size determined by a geometric progression; and **mapping $\mathbf{g}$ onto H at $\mathbf{x}_{\mathbf{g}}$** by performing binary search along the line passing through $\mathbf{g}$ and the original image $\mathbf{x}$. However, geometrically speaking, the estimated gradient $\mathbf{u}$ is not the optimal search direction, and we can find a better boundary point $\mathbf{x}_t$ by connecting $\mathbf{x}$ to a certain point on the semicircle B, then take the intersection point on H. Fig 1 shows that the tangent point $\mathbf{k}$ is the optimum update because it can be mapped onto H at $\mathbf{x}_t$ that results in the minimum distortion.

3.3 Defining Optimal Tangent Points

There is some experimental evidence that the decision boundary of neural networks is a rather low-curvature-smooth surface [17, 33]. Based on this observation, we approximate the decision boundary locally with a flat hyperplane H. The three-dimensional space of our problem is illustrated in Fig. 3a. 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.

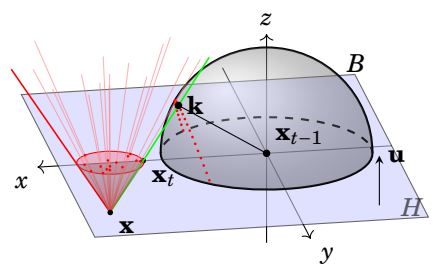


Figure 2: Geometrical explanation of Theorem 1. All points on H that are closer to $\mathbf{x}$ than $\mathbf{x}_t$ are within a red disk, which is the intersections of H and the red cone whose apex is $\mathbf{x}$. Clearly $\mathbf{k}$ is the only intersection point of the cone and the hemisphere B. Thus, of all the lines intersecting B, only the tangent line leads to the shortest distance from $\mathbf{x}$ to H.

Based on our motivation described in Section 3.2, $\mathbf{x}_{t-1}$ denotes the updated example that already lies on the decision boundary. We create a virtual hemisphere B centered at $\mathbf{x}_{t-1}$ 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 $\mathbf{x}_{t-1}$. Ideally, B is small enough to be fully contained in the target class region, *i.e.,* B is strictly contained in the adversarial region such that $\phi(\mathbf{x}') = 1$ for $\forall \mathbf{x}' \in B$. 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 $\mathbf{x}$ which creates infinitely many tangent points on B, shown as the red points in Fig. 3a. Still, we will show that only one tangent point can lead to the minimum distortion when mapping it onto H along the tangent line.

Formally, let $\mathbf{k}$ be any point in B, $\mathbf{u}$ be the approximate gradient of H estimated at $\mathbf{x}_{t-1}$, and $\mathbf{x}_t$ be the intersection of H and the line passing through $\mathbf{x}$ and $\mathbf{k}$, we have the following theorem.

Theorem 1 *Let H, $\mathbf{u}$, $\mathbf{k}$, $\mathbf{x}$, and $\mathbf{x}_{t-1}$ be defined above, then the distance $||\mathbf{x} - \mathbf{x}_t||_2$ is shortest if $\mathbf{k}$ is the optimal solution of the following constrained optimization problem.*

$$\arg\max_{\mathbf{k}} \quad \langle \mathbf{k} - \mathbf{x}_{t-1}, \mathbf{u} \rangle \tag{3}$$

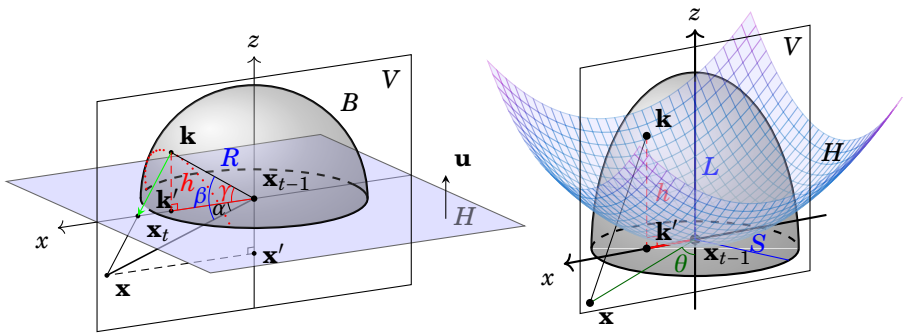
$$s.t. \quad \langle \mathbf{k} - \mathbf{x}_{t-1}, \mathbf{x} - \mathbf{k} \rangle = 0 \tag{4}$$

$$||\mathbf{k} - \mathbf{x}_{t-1}||_2 = R \tag{5}$$

$$\langle \mathbf{k} - \mathbf{x}_{t-1}, \mathbf{u} \rangle \geq 0 \tag{6}$$

In particular, the optimal $\mathbf{k}$ is in the plane spanned by $\mathbf{u}$ and $\mathbf{x} - \mathbf{x}_{t-1}$.

The objective function of Eq. (3) is to maximize the projection of the vector $\mathbf{k} - \mathbf{x}_{t-1}$ on the direction of $\mathbf{u}$, which is equivalent to finding $\mathbf{k}$ that is farthest away from H. The first constraint ensures that $\mathbf{k}$ being a tangent point. The second constraint indicates $\mathbf{k}$ is on the sphere of B. The last constraint states that $\mathbf{k}$ cannot appear on the same side of H as $\mathbf{x}$, which is always satisfied if $||\text{Proj}_H \mathbf{x} - \mathbf{x}_{t-1}||_2 \geq R$, where the notation $\text{Proj}_H \mathbf{x}$ indicates the *orthogonal* projection of $\mathbf{x}$ onto the hyperplane H. If there is no feasible solution, then our algorithm



(a) Tangent Attack (hemisphere) (b) Tangent Attack (semi-ellipsoid)

Figure 3: The 3D illustration of the optimal tangent point $\mathbf{k}$ for the flat decision boundary (Fig. 3a) and the curved one (Fig. 3b). In Fig. 3a, the tangent point $\mathbf{k}$ is on the surface of a virtual hemisphere B with a radius R centered at the adversarial example $\mathbf{x}_{t-1}$. $\mathbf{x}$ is the original image, $\mathbf{x}_t$ is the intersection point of the tangent line and the decision hyperplane H, $\mathbf{k}'$ and $\mathbf{x}'$ are the orthogonal projections of $\mathbf{k}$ and $\mathbf{x}$ on H and z axis, respectively. In Fig. 3b, the decision boundary H is a curved surface, B is replaced by a virtual semi-ellipsoid to increase the height of $\mathbf{k}$ for better performance.

(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 Optimal Tangent Points

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

$$\langle \mathbf{x}, \mathbf{u} \rangle = ||\mathbf{x}||_2 \cdot ||\mathbf{u}||_2 \cdot \cos(\frac{\pi}{2} + \alpha) = ||\mathbf{x}||_2 \cdot ||\mathbf{u}||_2 \cdot (-\sin\alpha) \tag{7}$$

Then we have $\sin\alpha = -\frac{\langle \mathbf{x}, \mathbf{u} \rangle}{||\mathbf{x}||_2 \cdot ||\mathbf{u}||_2}$ and $\cos\alpha = \sqrt{1 - \sin^2\alpha} = \frac{\sqrt{||\mathbf{x}||_2^2 - (\langle \mathbf{x}, \mathbf{u} \rangle)^2}}{||\mathbf{x}||_2}$. By constraint (4), $\mathbf{x} - \mathbf{k}$ is orthogonal to $\mathbf{k}$. Thus, we have $\cos\beta = \frac{R}{||\mathbf{x}||_2}$ and $\sin\beta = \sqrt{1 - \cos^2\beta} = \frac{\sqrt{||\mathbf{x}||_2^2 - R^2}}{||\mathbf{x}||_2}$. Then $\sin\gamma$ and $\cos\gamma$ can be derived as functions of α and β from basic facts of trigonometric functions.

$$\begin{aligned} \sin\gamma &= \sin(\beta - \alpha) = \sin\beta\cos\alpha - \cos\beta\sin\alpha \\ \cos\gamma &= \cos(\beta - \alpha) = \cos\beta\cos\alpha + \sin\beta\sin\alpha \end{aligned} \tag{8}$$

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

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

$$\frac{\mathbf{k}'}{||\mathbf{k}'||_2} = \frac{\mathbf{x} - \mathbf{x}'}{||\mathbf{x} - \mathbf{x}'||_2} \tag{9}$$

Now, $||\mathbf{k}'||_2 = R \cdot \cos\gamma$ and $\mathbf{x}' = \langle \mathbf{x}, \mathbf{u} \rangle \cdot \mathbf{u}$ are plugged into Eq. (9), and $\mathbf{k}'$ is obtained as

$$\mathbf{k}' = \frac{\mathbf{x} - \mathbf{x}'}{||\mathbf{x} - \mathbf{x}'||_2} \cdot ||\mathbf{k}'||_2 = \frac{\mathbf{x} - \langle \mathbf{x}, \mathbf{u} \rangle \cdot \mathbf{u}}{||\mathbf{x} - \langle \mathbf{x}, \mathbf{u} \rangle \cdot \mathbf{u}||_2} \cdot R \cdot \cos\gamma \tag{10}$$

Therefore, $\mathbf{k}$ can be derived as:

$$\mathbf{k} = \mathbf{k}' + h \cdot \mathbf{u} = \frac{\mathbf{x} - \langle \mathbf{x}, \mathbf{u} \rangle \cdot \mathbf{u}}{||\mathbf{x} - \langle \mathbf{x}, \mathbf{u} \rangle \cdot \mathbf{u}||_2} \cdot R \cdot \cos\gamma + h \cdot \mathbf{u} \tag{11}$$

Finally, because $\mathbf{x}_{t-1}$ has been moved to the origin, we need to move $\mathbf{k}$ back by adding $\mathbf{x}_{t-1}$.

We remark that although the above derivation is illustrated in $\mathbb{R}^3$, it can be directly applied to higher dimensions. The reason is Theorem 1, which essentially reduces any dimension space to $\mathbb{R}^2$: to find the optimal $\mathbf{k}$, we only need to focus on the *plane* V spanned by $\mathbf{u}$ and $\mathbf{x} - \mathbf{x}_{t-1}$.

3.5 Generalized Tangent Attack

When the decision boundary is not locally-flat enough, the boundary point obtained via the tangent point may not be the optimal (see Fig 3b for a illustration). 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 R will reduce the convergence rate of our algorithm: the distance between $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ is proportional to R. Therefore, we propose a method to change the shape of the semi-sphere based on the curvature of the decision boundary, instead of reducing R. Due to space limitation, we only offer a brief sketch of ideas here, and leave a formal description to the appendix.

First, notice that although in high-dimension space the shape of the decision boundary can be very bizarre, the important thing for our algorithm is only the situation in the 2D plane spanned by $\mathbf{x}$ and $\mathbf{u}$. In particular, if the decision boundary is "downwardly" curved (as opposed to the example in Fig 3b), then searching along the tangent line is still better than the HSJA solution. So the only "bad case" we have to deal with is the case in Fig 3b.

Now, let us focus on the 2D plane V spanned by $\mathbf{x}$ and $\mathbf{u}$, then $\mathbf{u}$ and $\mathbf{v} := \frac{\mathbf{x} - \langle \mathbf{x}, \mathbf{u} \rangle \cdot \mathbf{u}}{||\mathbf{x} - \langle \mathbf{x}, \mathbf{u} \rangle \cdot \mathbf{u}||_2}$ form an orthonormal basis of V. We can then project $\mathbf{x}$ onto V as $\mathrm{Proj}_V \mathbf{x} = x_0 \mathbf{v} + z_0 \mathbf{u}$, where $(x_0, z_0) = (||\mathbf{x}||_2 \cdot \sin\theta, -||\mathbf{x}||_2 \cdot \cos\theta)$ and θ is the angle between $\mathbf{x}$ and $-\mathbf{u}$, *i.e.,* $\theta = \arccos \frac{\langle \mathbf{x}, -\mathbf{u} \rangle}{||\mathbf{x}||_2 \cdot ||\mathbf{u}||_2}$. Now, the ellipsoid becomes a ellipse on V. We denote L as the radius along $\mathbf{u}$, and S for the radius along $\mathbf{v}$. Suppose $\mathbf{k} \in V$ is identified as $\mathbf{k} = x_k \mathbf{v} + z_k \mathbf{u}$, then (x_k, z_k) are as follows.

$$\begin{aligned} x_k &= \frac{S^2 \left(L^2 - z_0 \cdot \frac{L^2 S^2 z_0 + L^2 x_0 \sqrt{-L^2 S^2 + L^2 x_0^2 + S^2 z_0^2}}{L^2 x_0^2 + S^2 z_0^2} \right)}{L^2 \cdot x_0} \\ z_k &= \frac{L^2 S^2 z_0 + L^2 x_0 \sqrt{-L^2 S^2 + L^2 x_0^2 + S^2 z_0^2}}{L^2 x_0^2 + S^2 z_0^2} \end{aligned} \tag{12}$$

The complete derivation of Eq. (12) is presented in the appendix. In the implementation, the value of L is determined by using the process of calculating R in the hemisphere version of Tangent Attack. 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 boundary of curved surface while preserving a relatively large step size.

3.6 The Complete Algorithm

The whole procedure is shown in Algorithm 1. It first performs a binary search to map the initial sample $\tilde{\mathbf{x}}_0$ to the decision boundary[1], then it iterates $T-1$ times to obtain the adversarial example close to $\mathbf{x}$. In the first iteration, we sample B_0 probes around the boundary image to estimate the gradient, which is increased by multiplying $\sqrt{t}$ at the t-th iteration. This is because the error of the estimated gradient has a great influence on later iterations, and more probes can help reduce such estimation errors. Then, a *while* loop is performed to guess the reasonable radius R by repeatedly halving the radius until the tangent point $\mathbf{k}$ becomes successful. Lastly, we map $\mathbf{k}$ back to the boundary via binary search to end this iteration.

[1]The binary search always maps $\mathbf{x}_{\mathrm{adv}}$ onto the adversarial side of H that satisfies $\phi(\mathbf{x}_{\mathrm{adv}}) = 1$.

Algorithm 1 Tangent Attack

Input: benign image $\mathbf{x}$, attack success indicator function $\phi(\cdot)$ defined in Eq. (1), initial batch size B_0, iteration T, mode $m \in$ {semi-ellipsoid, hemisphere}, radius ratio r.
Initialize $\tilde{\mathbf{x}}_0$ that satisfies $\phi(\tilde{\mathbf{x}}_0) = 1$.
$\mathbf{x}_0 \leftarrow \text{BinarySearch}(\tilde{\mathbf{x}}_0, \mathbf{x}, \phi)$. {boundary search}
$d_0 = ||\mathbf{x}_0 - \mathbf{x}||_2$.
for t **in** $1, 2, \ldots, T-1$ **do**
  Sample $B_t \leftarrow B_0\sqrt{t}$ vectors to estimate gradient $\mathbf{u}$.
  Initialize $R \leftarrow d_{t-1}/\sqrt{t}$. {the initial radius}
  while true do
    Compute $\mathbf{k}$ based on Eq (11) **if** m = hemisphere **else** Eq (12)
    $R \leftarrow \frac{R}{2}$ {search the radius, and we set $L = R, S = \frac{L}{r}$ if m = semi-ellipsoid}
    if $\phi(\mathbf{k}) = 1$ **then**
      break.
    end if
  end while
  $\mathbf{k} \leftarrow \text{Clip}(\mathbf{k}, 0, 1)$
  $\mathbf{x}_t \leftarrow \text{BinarySearch}(\mathbf{k}, \mathbf{x}, \phi)$. {boundary search}
  $d_t = ||\mathbf{x}_t - \mathbf{x}||_2$.
end for

4 Experiment

4.1 Experimental Setting

Datasets. Tangent Attack is evaluated on two datasets, *i.e.,* CIFAR-10 and ImageNet with image resolutions of $32 \times 32 \times 3$ and $299 \times 299 \times 3$, respectively. We randomly select 1,000 correctly classified images from the validation sets for experiments. The images of ImageNet are resized to meet the input resolution requirements of different target models.

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 γ that controls the termination of the binary search is set to 1.0 in the CIFAR-10 dataset and 1,000 in the ImageNet dataset because of its high-resolution images. The radius ratio r is set empirically, *i.e.,* 1.5 and 1.1 for CIFAR-10 and ImageNet datasets, respectively. Besides, we also set r to 1.5 when attacking defensive models. In the targeted attack, 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. Our approach is compared with four most recent state-of-the-art hard-label attacks: Boundary Attack (BA) [2], Sign-OPT [9], SVM-OPT [9], and HopSkipJumpAttack (HSJA) [6]. 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 γ). For the consistency of experiments, we translate the implementations of Sign-OPT, SVM-OPT and HSJA into the PyTorch version. 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, and our approach. For fair comparison, we set the initial direction θ_0 of Sign-OPT and SVM-OPT to the direction of a randomly selected target class image. Detailed configuration are presented in the appendix. **Target Models.** In CIFAR-10 dataset, we select four target models from the PyTorch-Classification package[2]: (1) a 272-layer PyramidNet+ShakeDrop network (PyramidNet-272) [19, 37] trained using AutoAugment [12] ; (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) [39]; and (4) a wide residual network with 40 layers (WRN-40) [39]. In the ImageNet dataset, we select four target models from the pretrainedmodels package[3]: (1) Inception-v3 [36], (2) Inception-v4 [35], (3) ResNet-101 [20], and (4) SENet-154 [21].

Evaluation Metric. Following previous studies [38], we report the mean ℓ_2 distortions as $\frac{1}{|\mathbf{X}|}\sum_{\mathbf{x}\in\mathbf{X}}(\|\mathbf{x}_{\text{adv}} - \mathbf{x}\|_2)$ under different query budgets, where $\mathbf{X}$ is the test set.

[2]Pretrained weights: `https://github.com/bearpaw/pytorch-classification`

[3]Pretrained weights: `https://github.com/Cadene/pretrained-models.pytorch`

Table 1: Mean ℓ_2 distortions of different query budgets on ImageNet dataset, where $r = 1.1$.

Target Model	Method	Targeted Attack						Untargeted Attack					
		@300	@1K	@2K	@5K	@8K	@10K	@300	@1K	@2K	@5K	@8K	@10K
Inception-v3	BA	111.798	108.044	106.283	102.715	86.931	78.326	-	107.558	102.309	95.776	78.668	60.296
	Sign-OPT	103.939	87.706	71.291	46.744	34.640	29.414	121.085	79.158	43.642	16.625	10.557	8.680
	SVM-OPT	**101.630**	82.950	67.965	46.275	35.694	31.106	121.135	66.027	36.763	15.736	10.501	8.789
	HSJA	111.562	95.295	82.111	52.544	37.395	30.425	103.605	57.295	37.185	15.484	9.989	7.967
	Ours (hemisphere)	103.781	**80.327**	**66.708**	**42.121**	**30.846**	**25.566**	94.752	52.523	35.229	15.040	9.748	7.793
	Ours (ellipsoid)	103.724	81.089	67.168	42.434	31.011	25.587	**94.668**	**52.037**	**34.540**	**14.643**	**9.540**	**7.618**
Inception-v4	BA	110.343	106.616	104.586	100.321	84.058	75.507	-	116.075	111.474	104.451	86.572	66.283
	Sign-OPT	101.620	85.731	69.719	46.416	34.957	30.004	132.991	86.431	48.292	18.678	11.567	9.262
	SVM-OPT	**99.856**	81.342	66.982	45.667	35.477	31.152	132.227	72.920	41.095	17.611	11.418	9.372
	HSJA	109.670	93.916	80.937	52.358	37.773	30.958	110.727	63.731	42.290	17.936	11.367	8.911
	Ours (hemisphere)	101.666	**78.683**	**65.304**	**41.629**	30.993	25.958	101.207	58.616	40.314	17.639	11.304	8.907
	Ours (ellipsoid)	101.495	79.210	65.888	42.002	**30.965**	**25.847**	**101.173**	**58.225**	**39.788**	**17.265**	**11.008**	**8.677**
SENet-154	BA	81.090	77.723	76.122	71.967	55.953	47.652	-	75.998	71.671	66.983	53.917	40.725
	Sign-OPT	75.722	62.876	49.191	30.155	21.333	17.672	70.035	47.705	27.314	10.890	6.643	5.245
	SVM-OPT	74.658	58.677	46.827	30.264	22.461	19.186	69.854	40.291	23.692	10.494	6.666	5.409
	HSJA	77.035	63.488	51.802	30.138	19.680	16.261	71.248	38.035	24.895	10.218	5.855	4.842
	Ours (hemisphere)	70.739	55.256	**43.694**	**24.961**	**16.756**	**13.876**	**65.589**	**35.689**	24.037	10.039	5.774	4.766
	Ours (ellipsoid)	**70.591**	**55.224**	44.047	25.041	16.854	14.047	65.871	35.768	**23.954**	**9.959**	**5.733**	**4.734**
ResNet-101	BA	81.565	77.903	76.366	72.392	58.746	51.679	-	64.007	60.389	56.544	44.175	31.371
	Sign-OPT	76.732	63.939	51.231	32.439	23.160	19.248	56.244	38.282	21.985	10.048	7.050	6.050
	SVM-OPT	77.031	61.417	49.842	32.806	24.553	20.964	55.894	32.638	19.409	9.830	7.185	6.281
	HSJA	76.121	63.091	52.301	31.018	20.472	16.911	56.264	27.443	17.717	7.649	4.723	**4.019**
	Ours (hemisphere)	**72.434**	**57.969**	**47.142**	**27.699**	**18.788**	**15.414**	53.197	26.777	17.651	7.730	4.822	4.107
	Ours (ellipsoid)	72.459	58.320	47.297	27.905	19.045	15.633	**53.058**	**26.631**	**17.384**	**7.602**	**4.720**	4.026

Table 2: Mean ℓ_2 distortions with different query budgets on CIFAR-10 dataset, where $r = 1.5$.

Target Model	Method	Targeted Attack						Untargeted Attack					
-		@300	@1K	@2K	@5K	@8K	@10K	@300	@1K	@2K	@5K	@8K	@10K
PyramidNet-272	BA	8.651	8.073	8.013	6.387	4.189	3.333	-	5.636	4.725	4.414	2.750	1.696
	Sign-OPT	8.279	6.331	4.250	1.718	0.960	0.718	4.387	2.334	1.178	0.403	0.267	0.226
	SVM-OPT	9.207	6.801	4.530	2.010	1.207	0.947	4.481	2.318	1.093	0.414	0.276	0.236
	HSJA	7.917	4.329	2.523	**0.793**	**0.489**	**0.397**	4.505	1.279	0.713	0.333	0.255	0.227
	Ours (hemisphere)	7.943	**4.267**	2.488	0.809	0.503	0.406	**4.256**	1.275	0.710	0.329	0.253	0.226
	Ours (ellipsoid)	**7.816**	4.277	**2.469**	0.803	0.505	0.412	4.432	**1.270**	**0.702**	**0.329**	**0.252**	**0.225**
GDAS	BA	8.487	7.885	7.821	6.034	3.632	2.703	-	2.717	2.514	2.373	1.642	1.106
	Sign-OPT	8.372	6.514	4.351	1.827	0.987	0.711	4.917	4.159	3.260	1.352	0.452	0.250
	SVM-OPT	9.529	7.243	5.092	2.347	1.317	0.958	4.909	3.950	2.736	1.082	0.371	0.234
	HSJA	7.714	3.566	1.966	0.591	0.365	0.301	2.188	0.756	0.483	0.261	0.208	0.189
	Ours (hemisphere)	**7.674**	**3.529**	**1.946**	0.585	0.366	0.302	2.190	0.774	0.485	0.257	0.206	0.187
	Ours (ellipsoid)	7.697	3.558	1.959	**0.583**	**0.361**	**0.298**	**2.161**	**0.745**	**0.476**	**0.255**	**0.204**	**0.185**
WRN-28	BA	8.688	8.046	7.984	5.786	2.486	1.555	-	4.425	3.648	3.435	1.543	0.832
	Sign-OPT	8.258	5.576	3.260	1.087	0.593	0.459	3.093	1.494	0.828	0.319	**0.239**	**0.213**
	SVM-OPT	9.516	5.968	3.744	1.367	0.728	0.553	2.977	1.466	0.723	0.325	0.245	0.221
	HSJA	6.810	2.603	1.326	0.518	0.389	0.347	3.052	0.797	0.508	0.299	0.250	0.232
	Ours (hemisphere)	6.802	2.556	1.311	0.519	0.394	0.353	**2.974**	0.785	**0.496**	**0.293**	0.249	0.233
	Ours (ellipsoid)	**6.755**	**2.543**	**1.281**	**0.513**	**0.387**	**0.345**	2.995	**0.782**	0.502	0.298	0.250	0.232
WRN-40	BA	8.658	8.014	7.953	5.738	2.484	1.566	-	4.377	3.586	3.367	1.487	0.821
	Sign-OPT	8.156	5.579	3.300	1.186	0.646	0.501	4.754	3.239	1.885	0.311	**0.226**	**0.201**
	SVM-OPT	9.339	6.061	3.840	1.445	0.800	0.605	4.457	2.756	0.739	0.310	0.229	0.206
	HSJA	6.909	2.648	1.330	0.528	0.400	**0.357**	2.992	0.777	0.498	0.290	0.242	0.225
	Ours (hemisphere)	6.944	**2.579**	**1.295**	**0.523**	**0.398**	0.358	**2.926**	**0.770**	**0.490**	0.288	0.243	0.227
	Ours (ellipsoid)	**6.783**	2.605	1.320	0.535	0.403	0.361	2.952	0.772	0.492	**0.288**	0.241	0.223

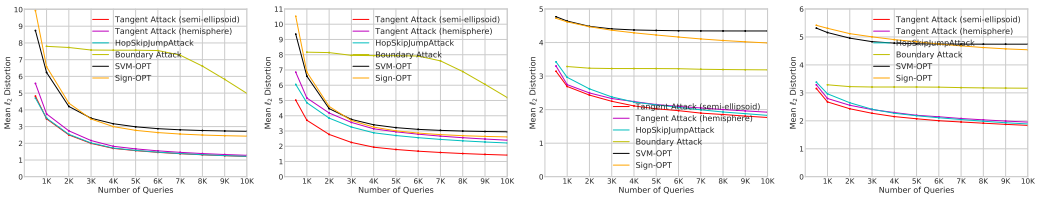
4.2 Comparisons with State-of-the-Art Methods

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

(1) In experiments of semi-ellipsoid version, CIFAR-10 requires a larger radius ratio r than ImageNet to achieve satisfactory performance. We speculate that the reason is that the target models of ImageNet has relatively flat decision boundaries compared with those of CIFAR-10.

(2) Tangent Attack (hemisphere) is more effective in targeted attacks, while the semi-ellipsoid version performs better in untargeted ones. This is because the adversarial region of target class is relatively narrow, and consequently the decision boundary becomes more flat.

Results of Attacks on Defensive Models. We perform untargeted attacks on defensive models of the CIFAR-10 dataset. Fig. 4 shows the results. We select 4 defensive models: (1) a model obtained through adversarial training abbreviated as AT [29]; (2) an improved AT that optimizes a regularized surrogate loss named TRADES [40]; (3) an image-transformation-

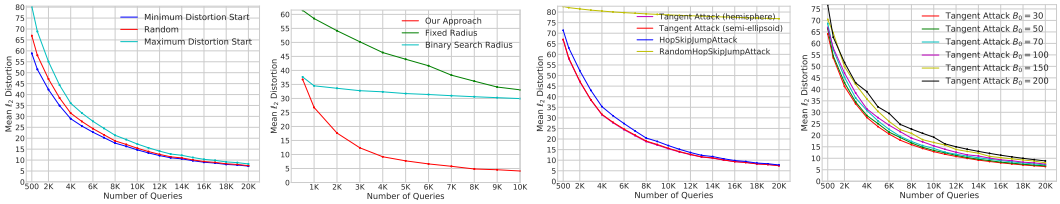


(a) AT (PGD-10 $\epsilon = 8/255$) (b) TRADES ($\epsilon = 8/255$) (c) JPEG (d) Feature Distillation

Figure 4: Experimental results of the attacks of defensive models with ResNet-50 backbone.

based defense called JPEG [18]; and (4) a DNN-oriented compression defense called Feature Distillation [27]. All defensive models adopt ResNet-50 as their backbones. Previous studies [40, 41] have shown that TRADES induces a significant increase in the curvature of the decision boundary. Fig. 4b shows that Tangent Attack (semi-ellipsoid) performs best in attacking TRADES. This superiority is also demonstrated in attacks on other defenses, proving the effectiveness of semi-ellipsoid version in attacking the models with curved decision boundaries.

4.3 Comprehensive Understanding of Tangent Attack



(a) initial $\tilde{\mathbf{x}}_0$ selection (b) radius schedules (c) jump directions (d) initial batch size B_0

Figure 5: Ablation studies of initial sample selection, radius schedules, jump directions and the initial batch size, which are obtained by attacking a ResNet-101 on the ImageNet dataset.

We conduct control experiments in Fig. 5 to understand our approach in depth.

Initialization. Our algorithm starts with an image $\tilde{\mathbf{x}}_0$ selected from the target class, and we study three selection strategies: (1) a randomly selected image, (2) the image with the smallest distance to the original image, and (3) the image with the largest distance to the original image. Fig. 5a shows that the strategy of (2) achieves the best performance.

The Schedule of Radius. The radius R is calculated adaptively to get a balance between feasibility and progress: it should be small enough to ensure the tangent point stays in the adversarial region, while large enough to make non-negligible progress towards the desired boundary point. Theoretically, the binary search algorithm can obtain an optimal R that gives the largest progress while ensuring the feasibility. However, the determination of such radius will cause a high query cost. Figure 5b demonstrates that our schedule produces the best performance, where "Fixed" indicates using a fixed radius value $R = (||\mathbf{x}_0 - \mathbf{x}||_2)/T$.

Jump Direction. Fig. 5c shows the effects of different jump directions. RandomHopSkipJumpAttack is a variant of HSJA which adopts a random vector $\mathbf{r}$ that satisfies $\langle \mathbf{r}, \mathbf{u} \rangle > 0$ as the jump direction. Fig. 5c proves the benefit of jumping to the optimal tangent point.

Initial Batch Size. Fig. 5d shows that a smaller B_0 performs better since it saves queries.

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 ℓ_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 curved decision boundaries. Lastly, we evaluate our approach through extensive experiments and show its superior performance compared with four attacks.

References

[1] 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. Springer, 2018.

[2] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

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

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

[5] Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020.

[6] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. HopSkipJumpAttack: a query-efficient decision-based adversarial attack. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020.

[7] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[8] 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*, 2019.

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

[10] Minhao Cheng, Simranjit Singh, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. 2020.

[11] 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 10932–10942, 2019.

[12] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[13] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 321–338, Santa Clara, CA, Aug. 2019. USENIX Association.

[14] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[15] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In *Proceedings of the IEEE 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 *Proceedings of the 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 *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1632–1640, Red Hook, NY, USA, 2016. Curran Associates Inc.

[18] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

[19] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the 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 *Proceedings of the 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[22] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018.

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

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

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

[26] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *Proceedings of 5th International Conference on Learning Representations*, 2017.

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

[28] Chen Ma, Li Chen, and Jun-Hai Yong. Simulating unknown target models for query-efficient black-box attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

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

[30] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Surfree: a fast surrogate-free black-box attack. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[31] Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *International Conference on Machine Learning (ICML)*, 2019.

[32] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, June 2016.

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

[34] Yucheng Shi, Yahong Han, and Qi Tian. Polishing decision-based adversarial noise with a customized sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1030–1038, 2020.

[35] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

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

[37] Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba, and Koichi Kise. Shakedrop regularization for deep residual learning. *arXiv preprint arXiv:1802.02375*, 2018.

[38] 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*, 2021.

[39] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[40] 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. PMLR, 2019.

[41] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11278–11287. PMLR, 13–18 Jul 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes] , it is included in appendix: although our approach is applicable to ℓ_∞ norm attack, the performance cannot be guaranteed.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] , our study is about adversarial attacks which have security considerations, and we leave them in appendix.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] , Section 3.3.

(b) Did you include complete proofs of all theoretical results? [Yes] , the proof of Theorem 1 is provided in the appendix.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] , error bars are not reported because it would be too computationally expensive.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] , in appendix: the experiments of all compared methods including our approach are conducted by using PyTorch 1.7.1 on a NVIDIA 1080Ti GPU.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] , we have cited the references related to CIFAR-10 and ImageNet datasets, as well as the target models (see footnotes of the last paragraph in Section 4.1).

(b) Did you mention the license of the assets? [Yes] , in appendix: the ImageNet dataset uses license of "Custom (research, non-commercial)".

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] ImageNet and CIFAR-10 are widely-used public dataset that can be freely downloaded by anyone.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] See the last answer: ImageNet and CIFAR-10 datasets are public datasets not created by us. The creators have already cleaned and anonymized the data.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] See above: the ImageNet and CIFAR-10 datasets are not created by us.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]