

SurFree: a fast surrogate-free black-box attack

Thibault Maho, Teddy Furon, Erwan Le Merrer
Univ. Rennes, Inria, CNRS
IRISA, Rennes, France
thibault.maho@inria.fr

Abstract

Machine learning classifiers are critically prone to evasion attacks. Adversarial examples are slightly modified inputs that are then misclassified, while remaining perceptively close to their originals. Last couple of years have witnessed a striking decrease in the amount of queries a black box attack submits to the target classifier, in order to forge adversarials. This particularly concerns the black box score-based setup, where the attacker has access to top predicted probabilities: the amount of queries went from to millions of to less than a thousand.

This paper presents *SurFree*, a geometrical approach that achieves a drastic reduction in the amount of queries in the hardest setup: black box decision-based attacks (only the top-1 label is available). We first highlight that the most recent attacks in that setup, *HSJA* [3], *QEBA* [14] and *GeoDA* [23] all perform costly gradient surrogate estimations. *SurFree* proposes to bypass these, by instead focusing on careful trials along diverse directions, guided by precise indications of geometrical properties of the classifier decision boundaries. We motivate this geometric approach before performing a head-to-head comparison with previous attacks with the amount of queries as a first class citizen. We exhibit a faster distortion decay under low query amounts (few hundreds to a thousand), while remaining competitive at higher query budgets.¹

1. Introduction

The literature on adversarial examples is divided into two shares, depending on the threat model: either the attacker has full knowledge of the target classifier [2, 26, 17] (white-box setting) or she/he has an unrestricted query access to the unknown classifier [18, 1, 14, 23, 3, 28, 13, 27, 11, 5, 12, 4] (black-box setting). The latter scenario is deemed as more relevant to gauge the intrinsic robustness of classifiers in

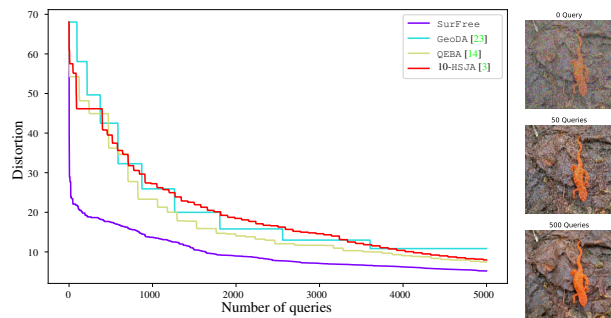


Figure 1. The perturbation distortion (ℓ_2 norm) vs. the number of queries for image ‘lizard’. Competitor attacks waste queries to estimate a gradient surrogate resulting in plateaus of distortion.

real-world applications (typically queried through an API).

Black box attacks are iterative procedures that keep on refining the quality of an adversarial example based on the pairs of submitted input / observed output. They are coined *score-based* when the attacker observes the top- k predicted probabilities or *decision-based* (a.k.a. hard label) when she/he only learns the top- k labels ($k \geq 1$). Indeed, the latter case where the output is solely the top-1 label is the most challenging because the attacker cannot rely on any rich information for crafting these adversarial examples.

It is striking that black-box attacks always use substitution to replace information they are missing. Early black-box attacks used a surrogate model (trained from a huge number of input / output pairs) mimicking the targeted model [20, 21]. The attack then boils down to a white-box setting on the surrogate with the hope that the adversarial example transfers to the target classifier. Almost all recent score-based attacks resort to gradient estimation to compensate for the lack of back-propagation, which is the key instrument of any white-box attack [4, 27, 13, 28]. The Hop-SkipJump attack (*HSJA* [3]) estimates the decision boundary by an hyperplane. As a last example, authors in [12] turn a decision-based setup into a score-based by probing noisy versions of an image to derive a score-like function from the top- k labels. The trend is thus to substitute missing infor-

¹Work supported by ANR / AID under Chaire IA SAIDA. Code available at <https://github.com/t-maho/SurFree>

mation by estimates in order to fall back to an easier setup.

The need for faster attacks consuming fewer queries is already present in the literature. Most notably, research works on score-based attacks managed to reduce query amount from millions of requests [12] to less than a thousand with most recent approaches [28]. Surprisingly, this impressive decrease has not reached comparable levels in the hard-label setup. In particular, paper [12] questions the model surrogate approach: while a considerable amount of queries is spent for training the surrogate, not a single adversarial example is forged. Moreover, access to the target model in practice is usually not free and not unlimited².

This argument should challenge any substitution mechanisms. They all consume a fair amount of queries and it is not clear whether they are worth the gain in term of distortion. Especially, many techniques trade some query amount for an accurate gradient estimate giving birth to good perturbation directions [23, 3]. During this step, the adversarial is not updated and the distortion stalls as Fig. 1 shows.

This paper considers the query amount as a central criterion. It presents a *fast* black-box decision-based attack, named `SurFree`, motivated by practical applications in which a low amount of queries is key. Fast means that it outperforms the state of the art when it comes to the distortion of adversarials under a low query budget (as exemplified in Fig. 1 with the purple curve).

The main contributions of this paper are:

- `SurFree`, a black box decision-based attack not using any substitution mechanism: no surrogate of the target model, no score reconstruction, no estimation of gradient. It is inspired by the early works [8, 1].
- a geometrical mechanism to get the biggest distortion decrease for a given direction to be explored under the assumption of a hyperplane boundary [10].
- a head to head comparison of the recent approaches with distortion as a function of query number.

Experimental results show that `SurFree` overcomes state of the art on the query amount factor (a thousand of queries), while still remaining competitive with unlimited queries (normal scenario for competitors).

2. Related Works

2.1. Watermarking

Digital watermarking embeds a secret and invisible mark into images. A watermark detector is a two-class classifier checking for the presence or absence of the mark in a query image. This community called oracle attack what we now

²see e.g., <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/face-api/> for conditions.

call a black-box attack: the attacker has the secret-keyed detector in hand as a black sealed box, and calls it iteratively to either estimate the secret key or to remove the watermark from protected images. This latter problem is equivalent to forging adversarial images in the decision based setting.

All the ingredients used nowadays were already present in this literature dating back to 1997 [7, 15]: surjection onto the boundary with binary search, estimation of the gradient at a ‘sensitive’ point lying on the boundary, dimension reduction. The new `HopSkipJumpAttack` [3] is indeed very similar to the old `Blind Newton Sensitivity Attack` [6]. The last work on this subject by this community [8] surprisingly does not use any gradient estimate but random directions; like the very first decision-based attack [1].

2.2. Black box adversarial examples

This paper operates in a black box setup; white box attacks [2, 26, 17] are considered out of its scope.

There has been a huge improvement on the amount of queries of *score-based* black box attacks. On ImageNet, the order of magnitudes of the first attacks were of some hundreds of thousands queries for one image with a reported runtime of 20 minutes in [4]. Nowadays state-of-the-art attacks make less than one thousand calls to the classifier, thanks to advanced gradient estimators [13] and Zero Order Optimization techniques [27, 28, 16].

The *decision-based* attacks followed the same trend but with a factor of ten. Brendel *et al.* report in the order of one million of queries for one image in one of the first black-box decision based paper [1, Fig. 6]. Then, the order of magnitude went down to tens of thousands [3, Fig. 4] [14, Fig. 5] and even some thousands in [23, Fig. 2]. No decision-based paper reports results with less than one thousand of calls on ImageNet. This paper explores this range of query budget.

The main engine of the *decision-based* attacks iterates the three following steps: i) the surjection (find a point on the boundary), ii) the estimation of the gradient (*i.e.* the normal vector of the tangent hyperplane), iii) the update of the adversarial example. Step ii) proceeds by bombarding the model with small perturbations around the boundary point. The main problems are the trade-off between the number of queries devoted to this task and the accuracy of the gradient estimate (see [3, Th. 2], [23, Lemma 2], [14, Th. 1]) and the impact of this accuracy on the convergence of the attack (see [23, Th. 2]). In the end, [23] recommends that step ii) consumes a number of queries following a geometric sequence w.r.t. the iteration number, whereas [3] makes it proportional to the square root of the iteration number. This paper follows the opposite strategy: no query is spent for a gradient estimate.

A second track of improvement is dimension reduction restricting the perturbation to a low dimension subspace. This a priori increases of distortion at convergence since

the attacker has fewer degrees of freedom, but it indeed facilitates the estimation of the projected gradient. The latter is more important for low query budget. The choice of the subspace incorporates prior information: it usually corresponds to a low-frequency band (of the full DCT transform [23, 14]) containing most critical information about the visual content of the image. This paper shows that the block DCT yields better results.

3. Problem statement

We introduce the following notations. The pre-trained classifier is represented as the function $f : [0, 1]^D \rightarrow \mathbb{R}^C$. For a given input image \mathbf{x} , the final decision is the top-1 label $\text{cl}(\mathbf{x}) := \arg \max_k f_k(\mathbf{x})$, $f_k(\mathbf{x})$ being the predicted probability of class k , $1 \leq k \leq C$.

The attacker does not know the function f and can only observe the decision $\text{cl}(\mathbf{x})$ for any image \mathbf{x} . From an original well classified image \mathbf{x}_o , the attack is untargeted as it looks for an image \mathbf{x}_a close to \mathbf{x}_o and s.t. $\text{cl}(\mathbf{x}_a) \neq \text{cl}(\mathbf{x}_o)$. This defines the outside region $\mathcal{O} := \{\mathbf{x} \in \mathbb{R}^D : \text{cl}(\mathbf{x}) \neq \text{cl}(\mathbf{x}_o)\}$ and the optimal adversarial image:

$$\mathbf{x}_a^* = \arg \min_{\mathbf{x} \in \mathcal{O}} \|\mathbf{x} - \mathbf{x}_o\|. \quad (1)$$

This is a hard problem and the attack is indeed an efficient algorithm finding an approximate solution.

We assume that when knowing a point $\mathbf{y} \in \mathcal{O}$, it is possible to find a point $\mathbf{x}_b \in \overline{\mathcal{O}}$ that lies on the boundary denoted by $\partial\mathcal{O}$. This is usually done by a line search in the literature [3, 14, 23]. There has been experimental evidence that the boundary is a rather smooth low curvature surface for deep neural networks [9]. This justifies that the boundary is often approximated by an hyperplane locally around a boundary point: in other words, locally around $\mathbf{x}_b \in \partial\mathcal{O}$, there exists $\mathbf{n} \in \mathbb{R}^D$, $\|\mathbf{n}\| = 1$ s.t. $\mathbf{y} \in \mathcal{O}$ if $\mathbf{y}^\top \mathbf{n} \geq \mathbf{x}_b^\top \mathbf{n}$.

4. Our Approach

The study of the recent attacks [3, 23, 14] under the query budget viewpoint, reveals the presence of plateaus (see Fig. 1). These are due to the construction of a surrogate for gradients, and appear to be particularly costly. Moreover, the budget allocated to gradient estimate in [3] does not impact the speed of convergence: fewer queries give less accurate gradient estimates yielding a smaller distortion decrease but at a higher rate. Our rationale is to set this query budget to its extreme value, *i.e.* zero. We thus trade this budget for more directions investigated with the hope that their multiplication allows for a faster distortion decrease. We now develop this idea.

4.1. Basic idea

Let us assume that we know a point on the boundary: $\mathbf{x}_b \in \partial\mathcal{O}$. We define $d := \|\mathbf{x}_b - \mathbf{x}_o\|$ and $\mathbf{u} := (\mathbf{x}_b -$

$\mathbf{x}_o)/d$ so that $\|\mathbf{u}\| = 1$. We restrict the search for a closer adversarial point in a random affine plane \mathcal{P} of dimension 2. This plane \mathcal{P} contains the point \mathbf{x}_o and is spanned by vector \mathbf{u} and a random orthogonal direction $\mathbf{v} \in \mathbb{R}^D$, $\|\mathbf{v}\| = 1$, $\mathbf{v}^\top \mathbf{u} = 0$. Note that $\mathbf{x}_b \in \mathcal{P}$.

In polar coordinates, we consider a point in \mathcal{P} that is at a distance $d(1 - \alpha)$ from \mathbf{x}_o and makes an angle θ with \mathbf{u} :

$$\mathbf{z}(\alpha, \theta) = d(1 - \alpha) (\cos(\theta)\mathbf{u} + \sin(\theta)\mathbf{v}) + \mathbf{x}_o, \quad (2)$$

with $\alpha \in [0, 1]$ and $\theta \in [-\pi, \pi]$. Note that $\mathbf{z}(0, 0) = \mathbf{x}_b$ and $\mathbf{z}(1, \theta) = \mathbf{x}_o, \forall \theta$. If $\mathbf{z}(\alpha, \theta)$ is adversarial, then the distortion decreases by $100 \times \alpha\%$.

This section shows how to choose (α, θ) to raise the probability of $\mathbf{z}(\alpha, \theta)$ being adversarial. This study makes a clear cut with [8, 1] which also consider random directions.

This section assumes that the intersection $\partial\mathcal{O} \cap \mathcal{P}$ is a line passing by \mathbf{x}_b and with normal vector $\mathbf{n} \in \mathcal{P}$, $\|\mathbf{n}\| = 1$. Without loss of generality, \mathbf{n} is pointing outside s.t. a point $\mathbf{z} \in \mathcal{P}$ is adversarial if $(\mathbf{z} - \mathbf{x}_b)^\top \mathbf{n} \geq 0$. In polar coordinates, $\mathbf{n} := \cos(\psi)\mathbf{u} + \sin(\psi)\mathbf{v}$ with $\psi \in (-\pi/2, \pi/2)$.

The point $\mathbf{z}(\alpha, \theta) \in \partial\mathcal{O} \cap \mathcal{P}$ minimizing the distance from \mathbf{x}_o is the projection of \mathbf{x}_o onto this line, obtained for $\theta = \psi$ and $\alpha = 1 - \cos(\psi)$. The attacker can not create this optimal point because angle ψ is unknown. Note that

- If $\psi = 0$, then $\mathbf{n} = \mathbf{u}$, $\mathbf{v}^\top \mathbf{n} = 0$, $(\mathbf{z}(\alpha, \theta) - \mathbf{x}_b)^\top \mathbf{n} = d((1 - \alpha)\cos(\theta) - 1) < 0$, and $\mathbf{z}(\alpha, \theta)$ is not adversarial. This corresponds to the case where $\partial\mathcal{O} \cap \mathcal{P}$ is a tangent line of the circle of center \mathbf{x}_o and radius d . This implies that \mathbf{x}_b is already optimum because it is the projection of \mathbf{x}_o onto $\partial\mathcal{O} \cap \mathcal{P}$.
- If $\theta = 0$ and $\alpha > 0$, then $(\mathbf{z}(\alpha, 0) - \mathbf{x}_b)^\top \mathbf{n} = \alpha(\mathbf{x}_o - \mathbf{x}_b)^\top \mathbf{n} < 0$ because \mathbf{x}_o is not adversarial. Therefore, $\mathbf{z}(\alpha, 0)$ is not adversarial.

For $\theta \neq 0$, calculation shows that $\mathbf{z}(\alpha, \theta)$ is adversarial if

$$g_\alpha(\theta) := \left| \frac{1 - (1 - \alpha)\cos(\theta)}{(1 - \alpha)\sin(\theta)} \right| \leq \tan(\psi)\text{sign}(\theta). \quad (3)$$

Point $\mathbf{z}(\alpha, \theta)$ might be adversarial only if ψ and θ share the same sign s.t. the rhs (3) is positive. In this case, the surprise is that (3) separates parameters (α, θ) that the attacker controls from the unknown angle ψ .

Minimizing $g_\alpha(\theta)$ raises the chances that (3) holds. Its derivative cancels for $\theta = \theta^*(\alpha) := \pm \arccos(1 - \alpha)$ (according to the sign of ψ) so that

$$g_\alpha(\theta^*(\alpha)) = \frac{\sqrt{1 - (1 - \alpha)^2}}{1 - \alpha} = |\tan(\theta^*(\alpha))|. \quad (4)$$

This quantity is an increasing function of α ranging from 0 ($\alpha = 0$) to $+\infty$ ($\alpha \rightarrow 1$). From now on, we denote by $\mathbf{z}^*(\theta) := \mathbf{z}(1 - \cos(\theta), \theta)$ a point created with this coupling.

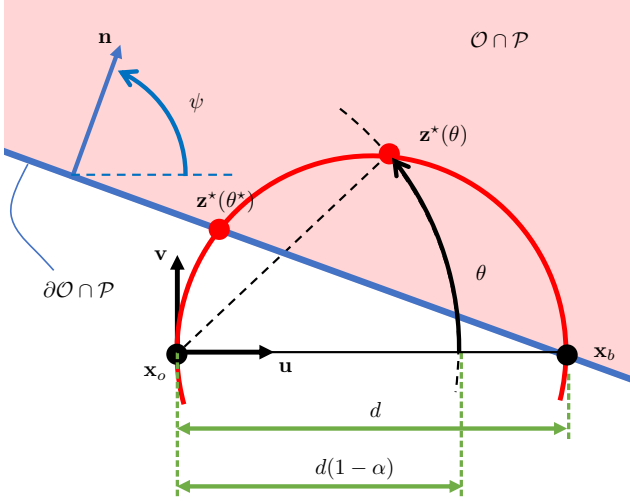


Figure 2. The geometrical configuration of the problem in \mathcal{P} .

Property 1 Consider the mid-point $\mathbf{c} = (\mathbf{x}_o + \mathbf{x}_b)/2$. The locus of the points $\mathbf{z}^*(\theta) \in \mathcal{P}$ is the circle of center \mathbf{c} and radius $d/2$. Indeed, $\mathbf{z}^*(0) = \mathbf{x}_b$ and $\mathbf{z}^*(\pm\pi/2) = \mathbf{x}_o$.

Little algebra shows that $\|\mathbf{z}^*(\theta) - \mathbf{c}\| = d/2, \forall \theta \in [-\pi/2, \pi/2]$. This circle is depicted in red in Fig. 2.

Property 2 If $\mathbf{z}^*(\theta)$ is adversarial, then so is $\mathbf{z}^*(\phi)$ for $\phi \in [0, \theta]$. Conversely, if $\mathbf{z}^*(\theta)$ is not adversarial, then so is $\mathbf{z}^*(\phi)$ for $\phi \in [\theta, \text{sign}(\theta) \cdot \pi/2]$.

This is due to the monotonicity of function $\alpha \rightarrow g_\alpha(\theta^*(\alpha))$.

Property 3 $\theta^* = \psi$ is the angle yielding a maximum distortion decrease of $\alpha = 1 - \cos(\psi)$. The point $\mathbf{z}^*(\theta^*)$ is indeed the projection of \mathbf{x}_o on the boundary line $\partial\mathcal{O} \cap \mathcal{P}$: $\mathbf{z}^*(\theta^*) = d \cos(\psi) \mathbf{n} + \mathbf{x}_o$.

This is shown by injecting (4) in (3).

4.2. Iterations over orthonormal directions

This section assumes that the boundary $\partial\mathcal{O}$ is an affine hyperplane passing through $\mathbf{x}_{b,1}$ in \mathbb{R}^D , with normal vector \mathbf{N} . We consider a random basis of $\text{span}(\mathbf{x}_{b,1} - \mathbf{x}_o)^\perp$ composed of $D - 1$ vectors $\{\mathbf{v}_i\}_{i=1}^{D-1}$. The normal vector is decomposed in spherical coordinates:

$$\begin{aligned} \mathbf{N} &= \sin(\psi_{D-1})\mathbf{v}_{D-1} + \cos(\psi_{D-1})\sin(\psi_{D-2})\mathbf{v}_{D-2} + \\ &\dots + \cos(\psi_{D-1})\dots\cos(\psi_2)\mathbf{n}_1, \end{aligned} \quad (5)$$

where $\mathbf{n}_1 := \sin(\psi_1)\mathbf{v}_1 + \cos(\psi_1)\mathbf{u}_1$ is the ℓ_2 normalized projection of \mathbf{N} onto hyperplane \mathcal{P}_1 spanned by \mathbf{v}_1 and $\mathbf{u}_1 := (\mathbf{x}_{b,1} - \mathbf{x}_o)/d$. Note that $\mathbf{N}^\top \mathbf{u}_1 = \cos(\psi_{D-1})\dots\cos(\psi_1)$. Then Prop. 3 finds $\mathbf{x}_{b,2} := \mathbf{z}^*(\theta^*) \in \mathcal{O} \cap \mathcal{P}_1$ and defines $\mathbf{u}_2 := (\mathbf{x}_{b,2} - \mathbf{x}_o)/d \cos(\psi_1) = \mathbf{n}_1$. We iterate on \mathcal{P}_2 spanned by $(\mathbf{v}_2, \mathbf{u}_2)$ to get $\mathbf{N}^\top \mathbf{u}_2 = \cos(\psi_{D-1})\dots\cos(\psi_2) \geq \mathbf{N}^\top \mathbf{u}_1$.

Property 4 Iterating this process converges to the adversarial point with minimal distortion.

Iterations increase the scalar product between \mathbf{N} and $(\mathbf{x}_{b,k} - \mathbf{x}_o) \propto \mathbf{u}_k$ given by:

$$\mathbf{N}^\top \mathbf{u}_k = \prod_{i=1}^{D-k} \cos(\psi_{D-i}). \quad (6)$$

At the end, $\mathbf{x}_{b,D} \in \mathcal{O}$ and $\mathbf{x}_{b,D} - \mathbf{x}_o$ is colinear with \mathbf{N} , thus pointing to the projection of \mathbf{x}_o to the hyperplane boundary.

A clever strategy browses the directions according to the decreasing order of their angles $(|\psi_k|)_k$ (biggest distortion decreases first). This is out of reach for the attacker oblivious to \mathbf{N} and not willing to spend queries for its estimate.

4.3. Convex boundary

Our procedure can be seen as a coordinate descent on a random basis. If the boundary $\partial\mathcal{O}$ is not a hyperplane but a smooth and convex surface, then cycling over the vectors $\{\mathbf{v}_i\}_{i=1}^{D-1}$ multiple times ensures convergence to a local minimum [19]. On one hand, this reference shows that the rate of convergence of the random coordinate descent (on expectation) is essentially the same as the *worst-case* rate of the standard gradient descent (when it is available). On the other hand, estimating the gradient in the black-box setting costs more queries than the coordinate descent of Sect. 4.1. These conflicting arguments deserve investigation.

5. The SurFree attack

This section presents the attack based on the ideas explained in Sect. 4. One iteration of SurFree is summarized in pseudo-code Alg. 1.

5.1. The algorithm

Initialisation. The algorithm needs an initial point $\mathbf{x}_{b,1} \in \partial\mathcal{O}$. It first generates a point $\mathbf{y}_0 \in \mathcal{O}$. As done in [23, 14], \mathbf{y}_0 is one image from the targeted class (targeted attack) or a noisy version of \mathbf{x}_o (untargeted attack). Defining $\mathbf{y}_\lambda = \lambda \mathbf{x}_o + (1 - \lambda)\mathbf{y}_0$, a binary search over $\lambda \in (0, 1)$ results in $\mathbf{x}_{b,1}$ adversarial and close to the boundary.

New direction. At iteration k , the point $\mathbf{x}_{b,k} \in \mathcal{O}$ and close to $\partial\mathcal{O}$ defines $\mathbf{u}_k \propto \mathbf{x}_{b,k} - \mathbf{x}_o$, $\|\mathbf{u}_k\| = 1$. Line 3 generates pseudo-randomly $\mathbf{t}_k \sim \mathcal{T}$ (see Sect. 5.2). A Gram-Schmidt procedure makes it orthogonal to \mathbf{u}_k and to the L (at most) last directions $\mathcal{V}_{k-1} := \{\mathbf{v}_j\}_{j=\max(k-L, 1)}^{k-1}$, producing the new direction \mathbf{v}_k in line 4.

Sign Search. The algorithm considers points $\mathbf{z}(\alpha, \theta)$ as defined in (2) with $\mathbf{u} = \mathbf{u}_k$, $\mathbf{v} = \mathbf{v}_k$, $d_k := \|\mathbf{x}_{b,k} - \mathbf{x}_o\|$, and the coupling $\cos(\theta) = 1 - \alpha$. The sign of θ depends on the sign of unknown ψ (see Sect. 4.1). Hence, we test some angles starting with the biggest amplitudes,

alternating + and - sign, as stored in the vector $\theta_{\max} \cdot \tau$ with $\tau := (1, -1, (T-1)/T, -(T-1)/T, \dots, 1/T, -1/T)$.

The search stops as soon as an adversarial image is found. If this fails, line 17 decreases θ_{\max} , direction \mathbf{v} is given up (line 18), and another direction is generated.

Binary Search. When the sign search finds an adversarial image at $\theta = \theta_{\max} t/T$, the binary search (line 12) refines the angle θ over the interval $\theta_{\max}[t, t + \text{sign}(t)]/T$ within ℓ steps. The result is θ^* and $\mathbf{z}^*(\theta^*)$ is the new boundary point $\mathbf{x}_{b,k+1}$ provoking a distortion decrease $\alpha^* = 1 - \cos(\theta^*)$.

5.2. Distribution of the directions

The algorithm is a random process as it draws directions from distribution \mathcal{T} according to Alg. 2. This has two roles: dimension reduction and adaptivity to the content of \mathbf{x}_o .

Dimension reduction is implemented with the help of a reversible image transformation (DCT 8×8 , or full frame DCT in Table 1). Line 3 selects a fraction ρ of the transform coefficients, typically in the low frequency subband. We draw ρD samples uniformly distributed over $\{-1, 0, 1\}$, the other transform coefficients being set to 0. The inverse transform yields the direction \mathbf{t} in the pixel domain.

Adaptivity to the visual content makes the perturbation less perceptible thanks to the masking effect well know in watermarking [8]. It shapes the adversarial perturbation like the visual content of \mathbf{x}_o . The following is a simple implementation of this principle: denote the i -th transform coefficient of image \mathbf{x}_o by $X_{o,i}$. Line 5 modulates the amplitude of a random variables by $A(|X_{o,i}|)$, where $A: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is a non decreasing function. The goal is to shape the power distribution of the perturbation as the one of the image.

5.3. Interpolation

Section 4 motivated our design assuming the boundary is an hyperplane. This extra interpolation is an *option* of SurFree inspired by the watermarking attack [8], which tackles convex surfaces with small curvature as in Fig. 3.

A given iteration starts with $\mathbf{x}_{b,k} \in \partial\mathcal{O}$ at angle $\theta = 0$ and distance d . The binary search in line 12 gives the angle θ^* of a boundary point at distance $d \cos(\theta^*)$. This option finds a third point on the boundary at angle $\theta^*/2$ thanks to a binary search between \mathbf{x}_o and $\mathbf{z}^*(\theta^*/2)$. This point, depicted in blue in Fig. 3, is at distance $\delta \leq d \cos(\theta^*/2)$.

Thanks to these three boundary points resp. at angle 0, $\theta^*/2$, and θ^* , we interpolate the mapping from angle to distance (of the surjection of $\mathbf{z}(\alpha, \theta)$ onto the boundary) by a second order polynomial and find its minimum at:

$$\hat{\theta} = \frac{\theta^* 4\delta - d(\cos(\theta^*) + 3)}{4 2\delta - d(\cos(\theta^*) + 1)}. \quad (7)$$

This option concludes by a binary search finding the point on the boundary between \mathbf{x}_o and $\mathbf{z}^*(\hat{\theta})$. The new point $\mathbf{x}_{b,k+1}$ is the closest point we found on the boundary.

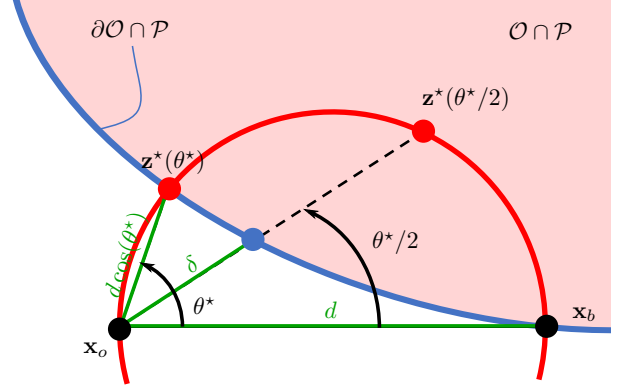


Figure 3. Interpolation mechanism to refine the boundary point.

Algorithm 1 One iteration of SurFree

Require: Original image \mathbf{x}_o , boundary point $\mathbf{x}_{b,k} \in \partial\mathcal{O}$, previous directions $\mathcal{V}_{k-1} := \{\mathbf{v}_j\}_{j=\max(k-L,1)}^{k-1}$

Ensure: Output $\mathbf{x}_{b,k+1} \in \partial\mathcal{O}$, \mathcal{V}_k

- 1: **New direction**
 - 2: $\mathbf{u}_k = \eta(\mathbf{x}_{b,k} - \mathbf{x}_o)$ ▷ $\eta(\mathbf{x}) := \mathbf{x}/\|\mathbf{x}\|$
 - 3: $\mathbf{t}_k \sim \mathcal{T}$ ▷ Algorithm 2
 - 4: $\mathbf{v}_k = \eta\left(\text{proj}_{\text{span}(\mathcal{V}_{k-1} \cup \mathbf{u}_k)^\perp}(\mathbf{t}_k)\right)$ ▷ Gram-Schmidt
 - 5: $\mathcal{V}_k = \mathcal{V}_{k-1} \cup \{\mathbf{v}_k\}$
 - 6: **Sign Search**
 - 7: $j = 1, \tau = (T, -T, (T-1), -(T-1), \dots, 1, -1)/T$
 - 8: **while** $\mathbf{z}^*(\theta_{\max} \cdot \tau_j) \notin \mathcal{O} \wedge j \leq 2T$ **do**
 - 9: $j \leftarrow j + 1$
 - 10: **if** $j < 2T$ **then**
 - 11: **Binary Search**
 - 12: $\theta^* = \text{BS}(\theta_{\max} \cdot \tau_j; \theta_{\max}(\tau_j + \text{sign}(\tau_j)/T))$
 - 13: $\theta_{\max} \leftarrow \theta_{\max}/(1 - \kappa)$
 - 14: Return $\mathbf{x}_{b,k+1} = \mathbf{z}^*(\theta^*)$
 - 15: **or Interpolation Sect. 5.3**
 - 16: **else** ▷ Sign Search failed
 - 17: $\theta_{\max} \leftarrow \theta_{\max} \times (1 - \kappa)$ ▷ Geometric decay
 - 18: Go to line 3 ▷ Give up
-

6. Experimental Work

We first specify the experimental setup and the parameters of our approach. We then perform an ablation study on SurFree (subsection 6.2), for it allows to precise gains on the two considered metrics. Subsection 6.3 performs a head-to-head comparison of all the competing approaches.

6.1. Datasets and Experimental Setup

Datasets For MNIST, we use a pre-trained CNN network that is composed of 2 convolutional layers and 2 fully connected Layers. Its accuracy is 99.14%. A subset of 100 *correctly* classified images have been randomly chosen to perform the ablation study. Our attack generates directions

Algorithm 2 Draw direction $\mathbf{t} \sim \mathcal{T}$

Require: Original image \mathbf{x}_o , frequency subband \mathcal{F} s.t.

$|\mathcal{F}| = \rho D$, $A(\cdot)$ shaping function

Ensure: A random direction \mathbf{t} perceptually shaped as \mathbf{x}_o

- 1: $\mathbf{X}_o = \text{DCT}(\mathbf{x}_o)$
 - 2: **for** $j = 1 : n$ **do**
 - 3: **if** $j \in \mathcal{F}$ **then**
 - 4: $r \sim \mathcal{U}_{\{-1,0,1\}}$ $\triangleright r \in \{-1, 0, +1\}$
 - 5: $T_j = A(|X_{o,j}|) \times r$
 - 6: **else**
 - 7: $T_j = 0$
 - 8: **Return** $\mathbf{t} = \eta(\text{DCT}^{-1}(\mathbf{T}))$ $\triangleright \eta(\mathbf{x}) := \mathbf{x}/\|\mathbf{x}\|$
-

on the pixel domain without any dimension reduction.

The ImageNet dataset is tackled by a pre-trained ResNet18, made available for the PyTorch environment [22]. Its top-1 accuracy is 0.6976. We randomly selected 350 *correctly* classified images from the ILSVRC2012’s validation set with size $D = 3 \times 224 \times 224$.

Setup and Code We now detail the specific parameters of SurFree, for both MNIST and ImageNet. We set empirically the following values in Alg. 1: $T = 3$, $L = 100$, $\theta_{\max} = 30$, $\kappa = 0.02$, at most $\ell = 10$ steps for the binary search (with an early stop if the range is lower than 1% of d). We develop SurFree on top of the FoolBox library.

Evaluation Metrics The two core evaluation metrics are the amount of queries, and the resulting distortion on the attacked image. The distortion is measured with the ℓ_2 norm over the space $[0, 1]^D$ (with D the number of pixels times the number of colour channel). For a given \mathbf{x}_o , it is the smallest distortion obtained over the sequence of queries $(\mathbf{q}_j)_{j=1}^k$ that happen to be adversarial:

$$d(k, \mathbf{x}_o) := \min_{1 \leq j \leq k : \text{cl}(\mathbf{q}_j) \neq \text{cl}(\mathbf{x}_o)} \|\mathbf{q}_j - \mathbf{x}_o\|_2 \quad (8)$$

The mean over N original images gives a characteristic of the attack efficiency revealing its capacity to find an adversary close to the original image and especially its speed.

$$d(k) := \frac{1}{N} \sum_{i=1}^N d(k, \mathbf{x}_{o,i}) \quad (9)$$

We define the success rate as the probability of getting a distortion lower than a target d_t within a query budget K :

$$S(d_t, K) := \frac{|\{i : d(K, \mathbf{x}_{o,i}) \leq d_t\}|}{N} \quad (10)$$

6.2. Ablation Studies

Impact of the components - MNIST This first ablation evaluates how the hyperplane hypothesis [10] meets a practical experimentation, and how the interpolation mechanism

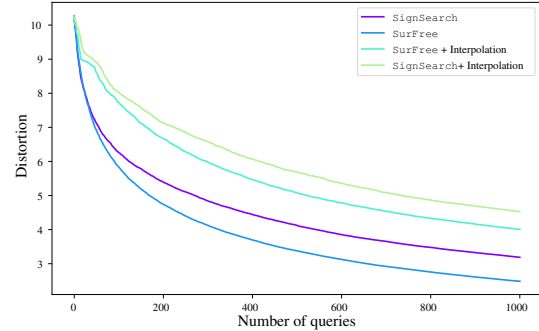


Figure 4. Ablation study on SurFree. Mean distortion $d(k)$ (9) vs. number k of queries on MNIST.

of Sect. 5.3 is able to compensate this hypothesis. To this end, four variants of our attack are tested in Fig. 4 and 5: SignSearch stops at line 10 of Alg. 1 whereas SurFree is the regular attack, ‘+Interpolation’ uses the option 5.3.

Our attack is highly random due to the generation of directions. This may yield unstable results with adversarial images of scattered distortion. Fig. 4 shows the distortion decrease averaged over 100 images and Fig. 5 the standard deviation for one image attacked 20 times.

This outlines the trade-off between the complexity of one iteration in terms of query number and the gain in the distortion decrease. The Interpolation option may yield substantial decrease depending on the direction. This explains its large standard deviation. Yet, its costs (2 more binary searches) slows down the speed. SignSearch is less costly and offers competitive distortions only at the beginning. SurFree strikes the right trade-off both in term of averaged distortion and standard deviation. Compared to SignSearch, it always exhausts the explored direction giving the best gain under the hyperplane boundary assumption. The first important insight is that this hypothesis seems to be good enough to ensure a rapid decay.

The ablation study also tested different values for some parameters of SurFree. The value of κ has no significant impact provided that $\kappa > 0$. Parameter T doesn’t benefit from higher value because of the finer search in line 12.

Impact of the direction generation domain - ImageNet

The literature reports that black-box attacks have difficulty in handling large images like ImageNet. Attack become slow because the space is too large to be explored efficiently. All competing attacks resort to a dimension reduction, typically by leveraging a full DCT transform [14, 23]. Yet, dimension reduction lowers the degrees of freedom for the attacker: the closest adversarial as defined in (1) has a bigger distortion under this constraint. The distortion supposedly converge faster but to a bigger limit.

SurFree is no exception. Table 1 shows that the distortion in the full pixel domain is bigger within the first

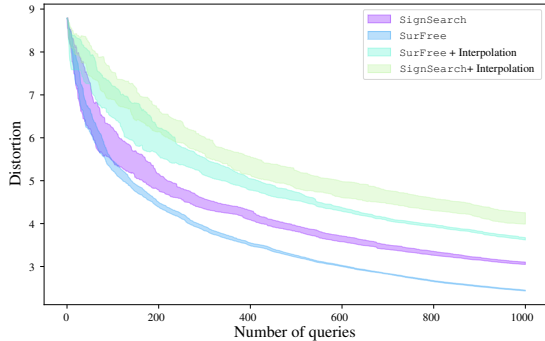


Figure 5. Ablation study on SurFree. The deviation of the distortion over 20 runs of SurFree on one MNIST image.

Space	Shaping $A(x)$	Dim. Reduc. ρ	$K = 100$	$K = 1000$
Pixel	-	100%	27.23	17.20
DCT_{full}	cst	50%	26.50	15.35
DCT_{full}	cst	25%	32.08	21.23
$DCT_{8 \times 8}$	cst	50%	19.49	10.69
$DCT_{8 \times 8}$	cst	25%	18.26	9.93
$DCT_{8 \times 8}$	x	50%	20.11	11.96
$DCT_{8 \times 8}$	x	25%	20.29	12.22
$DCT_{8 \times 8}$	$\tanh(x)$	50%	17.38	10.22
$DCT_{8 \times 8}$	$\tanh(x)$	25%	18.20	10.61

Table 1. Mean distortion $d(K)$ when random directions are generated with different subspaces and shaping (ImageNet).

thousand queries. For the same query budget, constraining the perturbation to lie in a smaller low-frequency subspace defined with the full DCT as in [14, 23] is beneficial. Yet, this frequency reduction have to be controlled, at the risk of suppressing too many frequencies and obtaining a more important distortion: distortion reported for a reduction of $\rho = 25\%$ are always larger than those for 50%.

We now question the type of DCT transform. Indeed, while the DCT full frame is widely acclaimed, we prefer the block-based DCT as used in JPEG. It gives a better space-frequency localization trade-off. Table 1 shows that it does change the distortions a lot. The 4 last rows of Table 1 focus on the adaptivity to the visual content of the original image (see Sect. 5.2). Amplitude function $A(x) = x$ concentrates the perturbation power too much on some high amplitude coefficients when the original image has sharp edges. $\tanh(x)$ is a good compromise between the constant and the identity functions. It offers early distortion drop and reaches similar levels than $A(x) = cst$ in the long run. Our design is driven by the small query budget requirement so we choose \tanh and $\rho = 50\%$ on $DCT_{8 \times 8}$.

6.3. Benchmarking

We compare to recent algorithms considered as state-of-the-art decision-based black-box attacks: HSJA [3], GeoDA [23] and QEBA [14]. These 3 algorithms leverage

gradient surrogates. The benchmark does not include older attacks like OPT [5] and BA [1] because they have proven less efficient than the three above-mentioned references.

We use the authors code for these algorithms: HSJA [3] is integrated in the FoolBox library [24, 25]. For GeoDA [23] and QEBA [14], we pull implementations from their respective GitHub repositories^{3,4} with default parameters. For GeoDA [23], the number of queries devoted to the gradient estimates follow a geometric progression of common ratio $\lambda^{-2/3}$ with $\lambda = 0.6$, and the dimension reduction focuses on 5,625 coefficients of the full DCT transform. Concerning QEBA [14], $\rho = 25\%$ dimension reduction on low frequency full DCT coefficients. HSJA [3] works on the pixel domain, the number of queries devoted to gradient estimates scales as $N_0 \sqrt{j}$ with j the iteration number. We tested two versions with $N_0 \in \{10, 100\}$, which is directly observable with the larger plateaus on Fig. 6.

A very important point is that all attacks are initialized with the same first adversarial example in order to avoid favoring a competitor by giving it an easier initialization.

Performance evaluation: distortion vs. queries Figure 6 displays the distortion of the perturbation (ℓ_2 norm)

³QEBA: <https://github.com/AI-secure/QEBA>

⁴GeoDA: <https://github.com/thisisalirah/GeoDA>

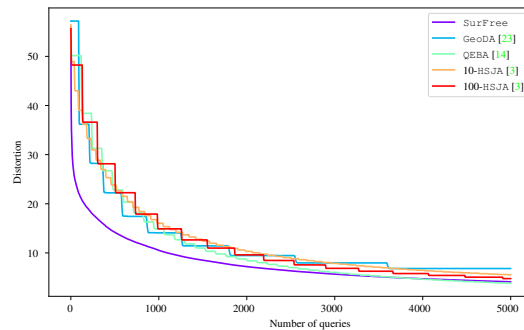


Figure 6. Benchmark on ImageNet. The amount of queries k (x -axis) w.r.t. mean distortion $d(k)$ (y -axis).

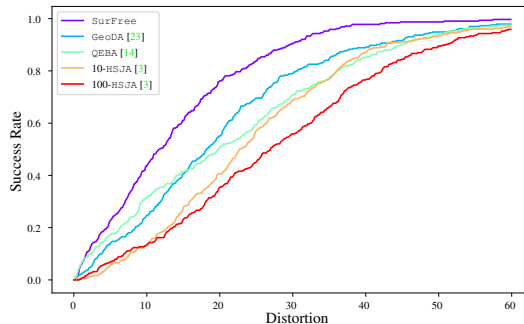


Figure 7. Success rate $S(d_t, K)$ (10) vs. target distortion d_t with $K = 500$ queries over ImageNet.

target d_t	$K = 500$ queries				$K = 1,000$ queries				$K = 2,000$ queries			
	HSJA [3]	GeoDA [23]	QEBA [14]	SurFree	HSJA [3]	GeoDA [23]	QEBA [14]	SurFree	HSJA [3]	GeoDA [23]	QEBA [14]	SurFree
30	0.56	0.79	0.71	0.90	0.88	0.93	0.88	0.96	0.98	0.96	0.97	0.99
10	0.13	0.25	0.32	0.44	0.23	0.52	0.46	0.57	0.40	0.70	0.69	0.73
5	0.07	0.14	0.17	0.23	0.09	0.21	0.30	0.31	0.13	0.39	0.47	0.50

Table 2. Success rate $S(d_t, K)$ for achieving a targeted distortion d_t under a limited query budget K (ImageNet).


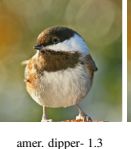
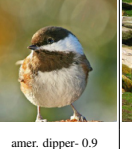
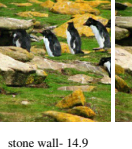
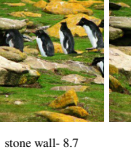



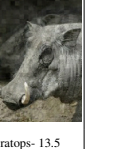

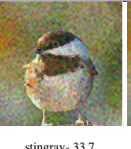
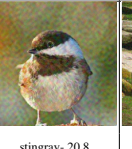
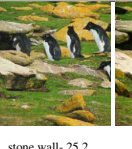
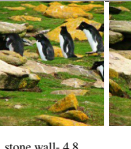




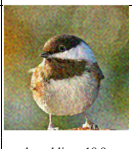
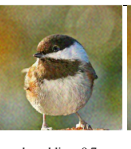

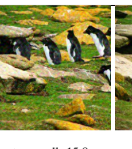
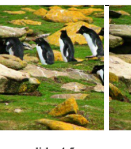
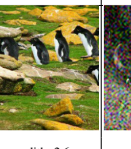



attack	$K = 100$	$K = 500$	$K = 1000$	$K = 100$	$K = 500$	$K = 1000$	$K = 100$	$K = 500$	$K = 1000$
SurFree	 amer. dipper- 2.6	 amer. dipper- 1.3	 amer. dipper- 0.9	 stone wall- 14.9	 stone wall- 8.7	 stone wall- 5.4	 cliff dwelling- 21.9	 cliff dwelling- 18.4	 triceratops- 13.5
QEBA [14]	 stingray- 60.6	 stingray- 33.7	 stingray- 20.8	 stone wall- 25.2	 stone wall- 4.8	 stone wall- 2.6	 wombat- 58.3	 wombat- 24.3	 wombat- 13.6
GeoDA [23]	 brambling- 18.9	 brambling- 9.7	 brambling- 5.8	 stone wall- 15.8	 megalith- 4.5	 megalith- 2.6	 armadillo- 49.4	 tusker- 31.3	 tusker- 18.9

Table 3. Visual trajectories for an easy (chickadee), a medium (king penguin), and a difficult image (warthog) - predicted label and distortion

versus the amount of queries. SurFree presents a smooth curve, resulting from the averaging over 350 images. Even with this averaging, the other attacks still show large plateaus (as highlighted in Fig. 1 for one image) because gradient estimates are scheduled at the same instants for any image. Note that these plateaus are not shown in the papers because the distortion is seen as a function of the iteration number, not the query number. The two most recent attacks, QEBA [14] and GeoDA [23] indeed beat HSJA [3] as reported in the corresponding papers. SurFree dives significantly faster than all attacks to lower distortions (notably from 1 to 750 queries), while QEBA [14] prevails at around 3,750 queries. Note that SurFree is also first with DCT full but for a shorter period (≈ 800 queries). For completeness, here are the scores at 10,000 queries: 2.09 (QEBA [14]) < 2.72 (SurFree) < 3.48 (HSJA_10) < 4.63 (GeoDA [23]). Although a small query budget drives its design, SurFree is not off in the long run. Similar results are observed for MNIST (pixel domain, without dimension reduction) where SurFree is ahead up to $\approx 5,000$ queries. For runtimes, here are the times to attack 1 image on ResNet18 at 1,000 queries: 4.1s (HSJA [3]) < 7.8s (SurFree) < 9.6s (GeoDA [23]) < 9.8s (QEBA [14]). With a comparable domain, SurFree is faster than GeoDA [23] and QEBA [14] by 20%.

Performance evaluation: Success rate We now consider three query budgets, $K \in \{500, 1,000, 2,000\}$, which are

rather low with regards to the state-of-the-art (see Sect. 2.2).

Table 2 details how the success rate $S(d_t, K)$ varies for some setup (d_t, K) (10). Fig. 7 shows the success rate $S(d_t, 500)$ increase with d_t . GeoDA [23] is superior to QEBA [14] for large target distortions only. Both schemes outperform HSJA [3]. SurFree remains the best attack for any target distortion up to this 2,000 query budget.

Finally, Table 3 displays the visual trajectories of three attacked images witnessed as easy, medium, and difficult to attack for SurFree. While all three attacks affect differently the images, SurFree gives relatively less annoying artefacts. We also note a drawback of QEBA [14]: the adversarials often keep the label of the random starting point (e.g. stingray), hence sometimes converging to a local minimum which is far from the optimal solution (1).

7. Conclusion

The performance of black box decision-based attacks reveals important gaps when it comes to the required amount of queries. Core to the three state-of-the-art approaches this papers considers is the estimation of gradients. This step is particularly costly, with regards to our novel geometrical attack SurFree. The trial of multiple directions together with a simple mechanism getting the best distortion decrease along a given direction allow a fast convergence to qualitative adversarials, within an order of hundreds of queries solely. This sets a new stage for future works.

References

- [1] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.
- [2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symp. on Security and Privacy*, 2017.
- [3] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. HopSkipJumpAttack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (sp)*, pages 1277–1294. IEEE, 2020.
- [4] 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, AISeC '17*, pages 15–26, New York, NY, USA, 2017. Association for Computing Machinery.
- [5] 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 *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- [6] P. Comesana, L. Perez-Freire, and F. Perez-Gonzalez. Blind newton sensitivity attack. *IEE Proceedings - Information Security*, 153(3):115–125, 2006.
- [7] I. J. Cox and J. . M. G. Linnartz. Public watermarks and resistance to tampering. In *Proceedings of International Conference on Image Processing*, volume 3, pages 3–6, 1997.
- [8] John W. Earl. Tangential sensitivity analysis of watermarks using prior information. In Edward J. Delp III and Ping Wah Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 449 – 460. International Society for Optics and Photonics, SPIE, 2007.
- [9] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 1632–1640. Curran Associates, Inc., 2016.
- [10] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [11] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pages 2484–2493, 2019.
- [12] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In Jennifer Dy and Andreas Krause, editors, *Proceedings of Machine Learning Research*, volume 80, pages 2137–2146, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.
- [13] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019.
- [14] H. Li, X. Xu, X. Zhang, S. Yang, and B. Li. Qeba: Query-efficient boundary-based blackbox attack. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1218–1227, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society.
- [15] J. Linnartz and Marten van Dijk. Analysis of the sensitivity attack against electronic watermarks in images. In *Information Hiding*, 1998.
- [16] S. Liu, P. Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.
- [17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [18] N. Narodytska and S. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1310–1318, 2017.
- [19] Yu. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.
- [20] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.
- [21] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17*, pages 506–519, New York, NY, USA, 2017. Association for Computing Machinery.
- [22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alche-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [23] 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.

- [24] 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, 34th International Conference on Machine Learning*, 2017.
- [25] 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.
- [26] Jerome Rony, Luiz G. Hafemann, Luiz S. Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [27] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.
- [28] Pu Zhao, Pin-Yu Chen, Siyue Wang, and Xue Lin. Towards query-efficient black-box adversary with zeroth-order natural gradient descent. *CoRR*, abs/2002.07891, 2020.