When does gradient estimation improve black-box adversarial attacks?

Enoal Gesny, Eva Giboulot, Teddy Furon

When does gradient estimation improve black-box adversarial attacks?

Enoal Gesny
Univ. Rennes, Inria, CNRS, IRISA
Rennes, France

Eva Giboulot
Univ. Rennes, Inria, CNRS, IRISA
Rennes, France

Teddy Furon
Univ. Rennes, Inria, CNRS, IRISA
Rennes, France

***Abstract*—The recent black-box adversarial attack `SurFree` demonstrated its high effectiveness resorting to a purely geometric construction. The method drastically reduced the number of queries necessary to craft low-distortion adversarial example compared to the preceding art which relied on costly gradient estimation. Recently, `CGBA` proposed to reintroduce gradient information to `SurFree`. Despite promising empirical results, no theoretical study of the method was provided. This paper fills this gap by providing a comprehensive analysis of the performance of `SurFree` and `CGBA`. Notably, we express conditions under which using the gradient information is guaranteed to improve upon `SurFree` performance. We also provide the theoretical distortion of each attack at a given iteration, demonstrating the convergence of `CGBA` to the optimal adversarial image. Finally, we study the optimal query allocation schedule for `CGBA`. The accompanying code is to be found at https://github.com/EnoalG/Use-of-gradient-for-black-box-attacks.**

***Index Terms*—Adversarial examples, black-box**

I. Introduction

Adversarial attacks perturb an input with minimal distortion to delude a classifier. The literature considers two setups that give birth to two attack strategies. In the white-box setup, the threat analysis states that the attacker knows the internals of the classifier [1]–[3]. On the contrary, the black-box setup is more difficult without this knowledge. A classifier induces a partition of the input space into regions, each associated with a given class. Probing the entire space to draw a map of this partition is not tractable. The attacker needs to query a lot the black-box classifier to find an adversarial example and then to reduce its distortion. This paper focuses especially on the hardest *decision-based* variant where the attacker only observes the predicted class for any input, contrary to *score-based* attacks where logits or probits are returned [4]–[7].

This type of attack, known as '*Oracle attacks,*' dates back to the late 1990s in the watermarking literature [8], [9]. Indeed, a zero-bit watermark detector is nothing more than a binary classifier. Unfortunately, the recent literature on adversarial machine learning does not refer to these pioneer works. It reinvents the wheel, resorting to the same tools, such as binary search, sensitive points, gradient estimation, and local approximation of the frontier by tangent hyperplane. Paper [10] is a notable exception. Yet, the literature on adversarial machine learning also proposes improvements: The state-of-the-art in black-box decision-based attacks is the recent `CGBA` [11].

Black-box attacks follow an iterative process refining the quality of the adversarial example. The biggest challenge is to provide theoretical guarantees, such as convergence to the nearest adversarial example. This is usually done under some strong assumptions, such as the frontier between class regions in the input space is a hyperplane. The distortion of the closest adversarial examples found within a given number of queries monitors the rate of convergence. Another issue is the number of queries dedicated to the estimation of the gradient at each iteration. Does the gradient estimation always lead to an improvement in the convergence rate?

This paper answers this question by providing an expression of the optimal number of queries for estimating the gradient to outperform `SurFree`. This is done through a theoretical justification of the approach of `CGBA` and the derivation of new results about the performance of `SurFree` and `CGBA`. In particular, we derive an expression of the expected distortion of both algorithms at each iteration of the adversarial search. An outline of the proof of each proposition can be found in the appendix. The detailed version can be found in the supplementary material.

II. Related Work

A. Watermarking Literature

In an oracle attack, the attacker has unlimited access to a watermark detector and one of the two following goals: 1) To remove the watermark of protected content, 2) to disclose a part of the secret key [12]–[14]. The former option is also known as the 'closest point attack'. The latter is not detailed as it is specific to watermarking.

These two goals resort to the same core process: the sensitivity attack, whose roots date back to [15]. Its first step is to find a point on the frontier. This is called a *sensitive point* because a small perturbation flips the detection output with a good probability. To find a sensitive vector, the attacker needs two images detected as watermarked and non-watermarked. One possible choice strongly distorts a watermarked image until it is deemed "non-watermarked." In the image space, the segment bridging the two intersects the frontier. A dichotomy line search finds this sensitive point.

The second step adds a small random perturbation and submits the modified vector to the detector. By repeating this process, the attacker can have a local approximation of the frontier, only valid in the neighborhood of the sensitive point: The frontier is approximated by a tangent hyperplane.

Once the frontier is locally estimated, the attacker knows in which direction to push the image to get closer to the original while staying close to the frontier. From this new point, the attacker again finds a sensitive vector and approximates the frontier again. This process is iterated until the improvement in quality of the sensitive content is no longer meaningful.

This attack is called BNSA (Blind Newton Sensitivity Attack) by its inventors [16]–[18]. Its main advantage is that no assumption is needed regarding the shape of the decoding region. BNSA converges to the global minimum distortion if the decoding region is convex, and to a local minimizer otherwise. The gradient estimation consumes as many queries as the number of space dimension. The attack focuses on a subspace to reduce this constraint.

Later, J. Earl proposes a method consuming fewer detection queries because no gradient estimation is needed [19]. The quality of the attacked content keeps improving with the number of queries. The quality improvement is huge for the first iterations but then slowly converges.

B. Adversarial Machine Learning Literature

In machine learning, an oracle attack is called a black-box attack. The black box is no longer a watermark detector but a classifier. The goal of an untargeted attack is to forge an input as close as possible to an original image while being not classified as the ground truth.

Although not referring to any work of the watermarking literature, the most well-known black-box attacks are clearly an application of BNSA to classifiers [20]–[22]. Of these methods, the most efficient is GeoDA [21] which finds an adversarial example by first estimating the vector normal to the classifier's decision boundary. It then perturbs the original image along the normal vector until finding an adversarial example. Its estimator is based on averaging multiple Gaussian perturbations of an image on the decision boundary.

These works offer a theoretical study of the best allocation strategy for the query budget. For instance, the number of queries spent for the gradient estimation should scale exponentially with a rate $2/3$ with the number of iterations [21]. Yet, their recommendations are not identical and, as observed in [10], this doesn't make a difference when observing how the distortion is decreasing with the number of queries.

The powerful attack SurFree [10] acknowledges that it is inspired by the work of J. Earl [19]. It is faster than the previous attacks because no gradient estimation is needed. It is a kind of coordinate descent. It iterates the following step: randomly pick a 2D affine hyperplane and find the optimal adversarial example constrained on this hyperplane.

The latest improvement is the attack CGBA [11]. It adopts the way SurFree finds the optimal point on a 2D hyperplane, yet this hyperplane is no longer random but generated by the estimation of the gradient at the sensible point. In a nutshell,

$$\texttt{CGBA} = \texttt{SurFree} + \texttt{GeoDA}.$$

Again, no recommendation is given w.r.t. the number of queries spent for the gradient estimation.

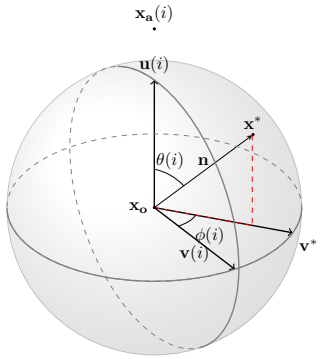


Fig. 1: Geometrical representation of the problem statement in SurFree– see notations from Sec. III-B.

III. Problem Statement and Main bricks

This section introduces the problem and the main bricks, SurFree and the gradient estimation, building CGBA.

A. Problem statement

The classifier $f : [0,1]^D \to \mathbb{R}^C$ takes images as input and outputs a vector of predicted probabilities associated to each of the C classes. The predicted class is the most likely:

$$\text{cl}(\mathbf{x}) := \underset{k \in [\![C]\!]}{\arg\max}\, f_k(\mathbf{x}). \tag{1}$$

We denote the original image as $\mathbf{x_o}$. The outside region is defined as $\mathcal{O} := \{\mathbf{x} \in [0,1]^D : \text{cl}(\mathbf{x}) \neq \text{cl}(\mathbf{x_o})\}$. The problem is to find the optimal adversarial example $\mathbf{x}^*$:

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{O}} ||\mathbf{x} - \mathbf{x_o}||. \tag{2}$$

The decision boundary $\partial\mathcal{O}$ is considered to be a hyperplane denoted by $\mathcal{H}$. The vector $\mathbf{n} \in \mathbb{R}^D$ normal to the decision boundary points outside such that $\mathbf{n}^\top(\mathbf{x} - \mathbf{x_a}) > 0$ if $\mathbf{x}$ is adversarial. Assuming we know a point $\mathbf{y} \in \mathcal{O}$, a binary search finds a point $\mathbf{x_a} \in \mathcal{H}$ between $\mathbf{x_o}$ and $\mathbf{y}$.

B. SurFree

The basic idea of SurFree is to iteratively refine the adversarial points by restricting the search to a new random 2-D plane $\mathcal{P}(i)$. At the beginning of the i-th iteration, $\mathbf{x_a}(i) \in \mathcal{H}$ is the closest adversarial example. The plane is constructed in two steps. First, we compute $\mathbf{u}(i) \in \mathbb{R}^D$ as follows:

$$\mathbf{u}(i) := \frac{\mathbf{x_a}(i) - \mathbf{x_o}}{||\mathbf{x_a}(i) - \mathbf{x_o}||}. \tag{3}$$

Secondly, a vector $\mathbf{v}(i) \in \mathbb{R}^D$ orthonormal to $\mathbf{u}(i)$ is randomly sampled. The plane is then constructed as $\mathcal{P}(i) := (\mathbf{x_o}, \mathbf{u}(i), \mathbf{v}(i))$. This plane $\mathcal{P}(i)$ intersects the hyperplane $\mathcal{H}$ in a line $\mathcal{L}(i)$. The circle $\mathcal{C}(i)$ with diameter $[\mathbf{x_o}, \mathbf{x_a}(i)]$ is drawn in $\mathcal{P}(i)$. This circle always intersects $\mathcal{L}(i)$ in at most two points: $\mathbf{x_a}(i)$ and the orthogonal projection of $\mathbf{x_o}$ onto $\mathcal{L}(i)$. The latter point becomes $\mathbf{x_a}(i+1)$, the adversarial point in $\mathcal{P}(i)$ leading to the smallest distortion. Note that one iteration of SurFree consumes k queries spent on the binary

search of $\mathbf{x_a}(i+1)$ over the circle $\mathcal{C}(i)$. This paper assumes that this 'search budget' is constant.

This creates the distortion series $d(i) := \|\mathbf{x_a}(i) - \mathbf{x_o}\|$, with $d(i+1) \leq d(i)$ converging to the global minimum $d^\star := \|\mathbf{x}^* - \mathbf{x_o}\|$, where $\mathbf{x}^*$ is the projection of $\mathbf{x_o}$ onto $\mathcal{H}$ [10, Prop. 4]. Another quantity of interest is the angle $\theta(i)$ between $\mathbf{x}^* - \mathbf{x_o}$ and $\mathbf{x_a}(i) - \mathbf{x_o}$ s.t. $\cos\theta(i) = \mathbf{n}^\top \mathbf{u}(i)$. This angle dictates how the distortion decreases over iterations: $d(i) = d^\star / \cos\theta(i)$.

C. Estimation of the normal vector

In a black-box setup, the attacker cannot compute the gradient of the classifier. Yet, he can estimate it for a point $\mathbf{x_a}$ on the boundary. This amounts to compute an estimation $\hat{\mathbf{n}}$ of the normal vector $\mathbf{n}$ of the tangent hyperplane.

`CGBA` and `GeoDA` use the same estimator:

$$\hat{\mathbf{n}} = \frac{\sum_{q=1}^{Q} \phi(\mathbf{x_a} + \mathbf{z}_q)\mathbf{z}_q}{||\sum_{q=1}^{Q} \phi(\mathbf{x_a} + \mathbf{z}_q)\mathbf{z}_q||_2}, \tag{4}$$

where $\mathbf{z}_q \sim \mathcal{N}(\mathbf{0}_D, \mathbf{I}_D)$ and ϕ is the indicator function of the class of $\mathbf{x}$:

$$\phi(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathrm{cl}(\mathbf{x}) \neq \mathrm{cl}(\mathbf{x_o}), \\ -1 & \text{if } \mathrm{cl}(\mathbf{x}) = \mathrm{cl}(\mathbf{x_o}). \end{cases} \tag{5}$$

Proposition 1. *For $1 < Q \ll D$, the following approximation is accurate:*

$$\mathbb{E}[(\hat{\mathbf{n}}^\top \mathbf{n})^2] \approx \frac{1}{1 + \frac{\pi}{2}\frac{D-1}{Q+\pi/2-1}} \approx \frac{1}{1 + \frac{\pi}{2}\frac{D-1}{Q}}$$

IV. Comparison of `SurFree` and `CGBA`

This section provides a careful theoretical comparison of the performance of `SurFree` and `CGBA`. A sketch of every proof can be found in the appendix.

A. About `SurFree`

The main proposal of `CGBA` is to integrate gradient information into `SurFree`. We start by justifying why such information can be useful:

Proposition 2. *At any iteration of `SurFree`, there exists an optimal vector $\mathbf{v}^*(i)$ which makes $\mathbf{x_a}(i+1) = \mathbf{x}^*$:*

$$\mathbf{v}^*(i) = \frac{\mathbf{n} - (\mathbf{n}^\top \mathbf{u}(i))\mathbf{u}(i)}{\sqrt{1 - (\mathbf{n}^\top \mathbf{u}(i))^2}} \tag{6}$$

The attacker thus needs a single iteration of `SurFree` if $\mathbf{n}$ is known perfectly.

This demonstrates that the knowledge of the normal vector allows a significant improvement over picking random directions in `SurFree`. Yet, since $\mathbf{n}$ is not accessible to the attacker in a black-box setting, it remains to be shown if its estimation still leads to a significant gain in efficiency: For the same query budget, does `CGBA` construct adversarial points with lower distortion? To answer this question, we first need to characterize the rate of distortion of `SurFree`:

Proposition 3. *At any iteration of `SurFree`, we have*

$$\cos^2\theta(i+1) = \cos^2\theta(i) + \sin^2\theta(i)\cos^2\phi(i), \tag{7}$$

where $\phi(i)$ is defined s.t. $\cos\phi(i) = \mathbf{v}(i)^\top \mathbf{v}^*(i)$.

The corollary is that $\cos\theta(i+1) \geq \cos\theta(i)$ so that the angle $\theta(i)$ converges to zero. This shows that $\mathbf{x_a}(i)$ converges to the optimal adversarial point $\mathbf{x}^*$. Note that $\mathbf{x_a}(i+1) = \mathbf{x}^*$ if $\cos\theta(i+1) = 1$. This happens if $\cos\theta(i) = 1$ ($\mathbf{x_a}(i)$ is already optimal) or if $\cos\phi(i) = 1$ which means that $\mathbf{v}(i) = \mathbf{v}^*(i)$.

Let us define $c(i) := \mathbb{E}[\cos^2\theta(i)]$ with $c(0) := \cos^2\theta(0)$.

Proposition 4. *As $\mathbf{v}(i)$ is randomly sampled in a $(D-1)$ dimensional space, then $\mathbb{E}[\cos^2\phi(i)] = (D-1)^{-1}$, $\forall i$. On expectation $\cos^2\theta(i)$ obeys to the following series:*

$$c(i+1) = c(i) + (1 - c(i))\eta_{\texttt{SurFree}} \tag{8}$$

$$= 1 - (1 - \eta_{\texttt{SurFree}})^i (1 - c(0)), \tag{9}$$

with $\eta_{\texttt{SurFree}} := (D-1)^{-1}$. This series converges to 1.

The corollary is an approximation of the decreasing rate of the distortion in expectation:

$$\mathbb{E}[d^2(i)] \approx d^{\star,2} \frac{1}{1 + (1 - \eta_{\texttt{SurFree}})^i(\cos^2\theta(0) - 1))}. \tag{10}$$

B. About `CGBA`

We can express the expected rate of distortion of `CGBA` using the same technique as for `SurFree`:

Proposition 5. *Let $\hat{\mathbf{v}}^*(i)$ be the estimation of the optimal vector $\mathbf{v}^*(i)$ (17). In `CGBA`, the expectation of $\cos^2\theta(i)$ obeys (9) but with $\mathbb{E}[\cos^2\phi(i)] = \mathbf{v}^*(i)^T \hat{\mathbf{v}}^*(i) = \eta_{\texttt{CGBA}}$, $\forall i$:*

$$\eta_{\texttt{CGBA}} \approx \frac{1}{\frac{\pi}{2}\frac{D-1}{Q}} \quad \textit{when} \quad Q \ll D. \tag{11}$$

Even for a moderate Q, $\eta_{\texttt{CGBA}}$ is larger than $\eta_{\texttt{SurFree}}$ so that the distortion of `CGBA` converges faster than the one of `SurFree`. However, `CGBA` consumes Q queries for the estimation $\hat{\mathbf{v}}^*(i)$ and k queries for the search overhead, whereas `SurFree` spends only k queries per iteration. Fig. 2 compares of the rate of distortion of `SurFree` and `CGBA`.

Note that (10) and (11) link the size of the input to the convergence speed for both `SurFree` and `CGBA`. In particular, for larger images, `SurFree` converges more slowly while `CGBA` necessitates more estimation queries Q per iteration to converge as fast as for smaller images.

Knowing both expected rates of distortion, we provide the minimum number of estimation queries $Q^*(k)$ such that the expected distortion of `CGBA` is lower than `SurFree` for the same amount of queries.

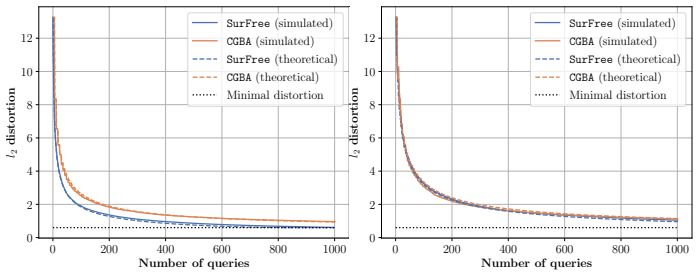
Proposition 6. *Assuming a search overhead $k \geq 2$ and $i \ll D$, if we set $Q = Q^*(k)$ such that:*

$$Q^*(k) := \left\lceil \frac{\pi k}{2k - \pi} \right\rceil, \tag{12}$$

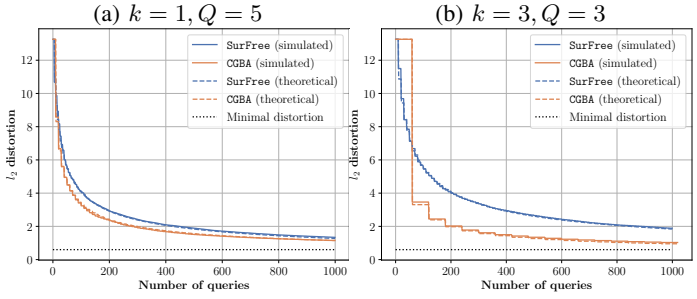
then:

$$\mathbb{E}[d^2_{\texttt{SurFree}}(i(1 + Q^*(k)/k))] \geq \mathbb{E}[d^2_{\texttt{CGBA}}(i)], \tag{13}$$

where `CGBA` consumes in i iterations $i(Q^*(k) + k)$ queries, i.e. as many as `SurFree` does in $i(1 + Q^*(k)/k)$ iterations.



(a) $k = 1, Q = 5$ (b) $k = 3, Q = 3$



(c) $k = 5, Q = 5$ (d) $k = 10, Q = 50$

Fig. 2: Distortion of adversarial examples as a function of the number of queries. Per iteration, k queries for the binary search, Q queries to estimate the normal vector $\mathbf{n}$. The simulated curves are averaged over 20 runs in a $D = 1000$ dimension space. CGBA quickly converges to a low distortion adversarial point whatever (k, Q), whereas SurFree's performance is extremely dependent on the search overhead k.

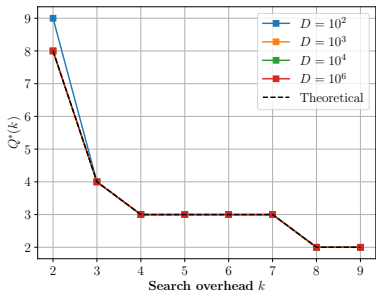


Fig. 3: Number of queries Q necessary for CGBA to outperform SurFree given a query search overhead k. The black curve is the approximation (12), color lines are the empirical values.

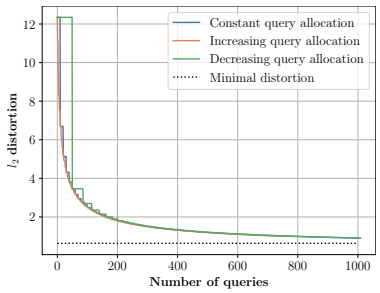


Fig. 4: Distortion depending on the query allocation in CGBA (simulated for $D = 1000$ dim.). Constant: $Q(i) = 10$, Increasing: $Q(i) = \lceil \sqrt{i} \rceil$, Decreasing $Q(i) = \lceil 50/\sqrt{i} \rceil$. The distortion converges to the same value whatever the allocation.

In other words, we can always find a Q such that CGBA is more efficient – i.e needs a lower number of queries (not iterations) – than SurFree in converging towards the optimal adversarial point. Figure 3 shows that there is no significant departure from the approximation (12) even for spaces with a comparatively small dimension such as $D = 100$.

This section ends by studying the impact of the query allocation schedule. Should the estimation query budget be constant over each iteration? Or should we increase or decrease the number of queries along the way? We simulated CGBA with each of these possible strategies for a space of dimension $D = 1000$ and report the results in Fig. 4. Still assuming the decision boundary to be a hyperplane, no strategy seems to strongly outperform the others. This is in sharp contrast to [20] and the allocation strategy proposed in the CGBA paper.

V. Experiments

This section presents our results on real-world datasets and classifiers. These experimental results are to be compared to our theoretical analysis performed under the assumption that the decision boundary is a hyperplane in the previous section.

A. Experimental setup

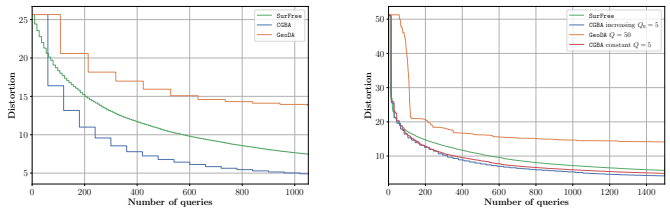
Dataset We work on two datasets, MNIST and ImageNet. For MNIST, we use a pre-trained CNN network composed of 2 linear layers to be as close as possible to our theoretical assumptions. The attacks are performed on a subset of 100 correctly classified images among the test set. The dimension of the space in this case is $D = 1 \times 28 \times 28 = 784$. For ImageNet, we use a pre-trained ResNet18 with nonlinear layers, departing from the separating hyperplance assumption. A subset of 100 correctly classified images among the 2012 ImageNet validation set have been chosen to perform the attacks. The dimension is $D = 3 \times 224 \times 224 = 150,528$.

Setup and Code We compare the performance of GeoDA, which is based purely on the use of the normal vector $\mathbf{n}$, SurFree, which does not use any gradient information and CGBA which combines the two approaches. We use our own implementation of CGBA to closely follow the presentation made in this paper, replacing SurFree random direction with the optimal vector of Prop. 2 computed from the estimated normal $\hat{\mathbf{n}}$ but performing the binary search of the angle in the same way as in SurFree.

When dealing with large dimensional spaces it is common practice in the black-box attack literature to restrict the search of adversarial examples within a smaller subspace [10], [11], [21], [22]. We follow what is currently considered the best practice for ImageNet, by performing the search in the DCT domain and focusing the search over the 50% lowest frequencies per 8×8 block – see Section 5.2 and 6.1 in [10].

Evaluation metrics Similarly as for our theoretical results, the two main quantities of interest to the attacker are: 1) the number of queries used to perform the attack and 2) the resulting distortion of the adversarial image. We thus report:

- The l_2-distortion computed over pixels values in $[0, 1]^D$,



(a) MNIST (b) ImageNet

Fig. 5: Comparison of the mean distortion (14) depending on the number of queries between SurFree, GeoDA and CGBA.

averaged over the N images in each dataset. We report this average $\bar{d}(i)$ for different amount i of queries:

$$\bar{d}(i) := \frac{1}{N}\sum_{j=1}^{N} ||\mathbf{x_o}^{(j)} - \mathbf{x_a}^{(j)}(i)||_2. \quad (14)$$

- The success rate defined as the rate of images that successfully deceive the classifier for a distortion equal or lower than target distortion d_t and query budget K:

$$S(d_t, K) = \frac{1}{N}\sum_{j=1}^{N}[||\mathbf{x_o}^{(j)} - \mathbf{x_a}^{(j)}(K)||_2 \leq d_t]. \quad (15)$$

B. Benchmark

a) Linear classifer – MNIST : We report the results for the MNIST dataset in Fig. 5a. We fix the number of query to estimate the normal vector to $Q = 50$ for both GeoDA and CGBA. On average, the search overhead is close to $k = 10$ for each algorithm. Estimation and search create a distortion plateau between each iteration. Now, observe that CGBA is more effective than SurFree from the first iteration and converges more quickly than the other attacks. This is aligned with our theoretical analysis: $Q > \left\lceil \frac{\pi k}{2k-\pi} \right\rceil$ and as such, CGBA should indeed have lower distortion than SurFree for every iterations. Also observe that GeoDA, though having access to the same estimation of the normal vector, is not able to use this information as efficiently as CGBA. This demonstrates the effectiveness of the geometrical approach of SurFree and the importance of merging the two approaches.

b) Non-linear classifier – ImageNet: Fig. 5b shows the experimental results. For CGBA, we report the average distortion for two different query schedules concerning the estimation: a constant one at $Q = 5$ and an increasing one with $Q(i) = 5\sqrt{i+1}$. Once again, CGBA outperforms SurFree for a very low number of queries, showing that the hyperplane approximation is somewhat robust for non-linear classifiers. However, CGBA performs approximately the same as SurFree for a few iterations before clearly outperforming it. Furthermore, the increasing schedule does help CGBA reaching a slightly lower distortion, something which is clearly not observed for linear classifiers – see Fig. 4. The effectiveness of CGBA is even clearer in Table I reporting the success rate of the attacks for ImageNet. Here CGBA– with a constant schedule at $Q = 5$ – outperforms SurFree whatever the query budget and target distortion chosen. This is especially true for low target distortion with gain ranging from 6% to close to 15%.

TABLE I: Attack success rate for achieving a targeted distortion d_t under a limited query budget K (ImageNet).

target d_t	GeoDA	SurFree	CGBA
$K = 500$			
30	0.82	**0.95**	**0.95**
10	0.45	0.56	**0.65**
5	0.35	0.4	**0.54**
$K = 1000$			
30	0.85	0.97	**0.99**
10	0.47	0.72	**0.75**
5	0.36	0.52	**0.60**
$K = 1500$			
30	0.87	**1.0**	**1.0**
10	0.50	0.79	**0.85**
5	0.38	0.59	**0.63**

c) Impact of the query allocation schedule: Figure 6 studies the impact of the allocation schedule more closely for ImageNet. Note that, in the same way as for the theoretical results in Fig. 4, every schedule converges quickly – here in approximately in 500 queries – to the same trajectory. Starting with a small number of queries allows faster convergence during the first iterations. There is no significant difference between CGBA with an increasing number of queries and CGBA with a constant schedule of $Q = 5$ queries.

VI. Conclusions

An important debate in black-box attacks is whether the surrogate gradient estimation is necessary or not to obtain highly efficient algorithms. The geometric construction of SurFree shows that estimation of the gradient is a waste of queries while CGBA shows that incorporating the gradient information back into SurFree could lead to improvements. This paper shows that this claim holds true as long as a sufficient number of query is provided for the estimation of the normal vector compared to the binary search overhead of SurFree. Furthermore, it provides an explicit expression of

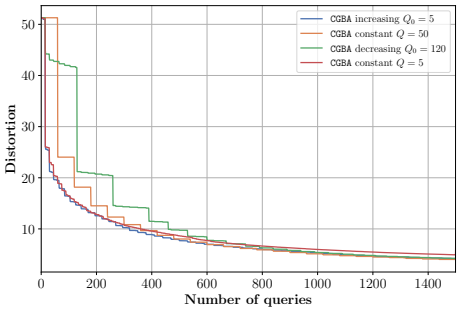


Fig. 6: Distortion (14) under different query allocation schedules. The decreasing (increasing) schedule uses $Q(i) = \frac{120}{\sqrt{i}}$ (resp. $Q(i) = 5\sqrt{i}$) queries for estimating $\mathbf{n}$ at iteration i.

the distortion of both SurFree and CGBA as a function of the dimension of the space and query budget, showing for the first time the convergence of the latter.

ACKNOWLEDGMENT

French ANR/AID under Chaire ANR-20-CHIA-0011-01.

REFERENCES

[1] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symp. on Security and Privacy*, 2017.

[2] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses," in *CVPR*, June 2019.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[4] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth Order Optimization based black-box attacks to deep neural networks without training substitute models," in *AISec*, 2017.

[5] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *AAAI*, 2019.

[6] A. Ilyas, L. Engstrom, and A. Madry, "Prior convictions: Black-box adversarial attacks with bandits and priors," in *ICLR*, 2019.

[7] P. Zhao, P. Chen, S. Wang, and X. Lin, "Towards query-efficient black-box adversary with zeroth-order natural gradient descent," in *AAAI*, 2020.

[8] I. J. Cox and J. . M. G. Linnartz, "Public watermarks and resistance to tampering," in *Proceedings of Int. Conf. on Image Processing*, 1997.

[9] J. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," in *Information Hiding*, 1998.

[10] T. Maho, T. Furon, and E. Le Merrer, "SurFree: a fast surrogate-free black-box attack," in *CVPR*, 2021.

[11] M. F. Reza, A. Rahmati, T. Wu, and H. Dai, "CGBA: Curvature-aware geometric black-box attack," in *ICCV*, 2023.

[12] J.-P. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," in *Information Hiding*, 1998.

[13] T. Kalker, J.-P. Linnartz, and M. van Dijk, "Watermark estimation through detector analysis," in *Proc. of ICIP*, Oct 1998.

[14] M. El Choubassi and P. Moulin, "Noniterative algorithms for sensitivity analysis attacks," *IEEE Trans. Information Forensics and Security*, vol. 2, no. 2, pp. 113–126, June 2007.

[15] I. J. Cox and J.-P. Linnartz, "Public watermarks and resistance to tampering," in *Proc. of ICIP*, Oct 1997.

[16] P. Comesaña, L. Pérez-Freire, and F. Pérez-González, "Blind Newton sensitivity attack," *IEE Proc. on Information Security*, vol. 153, no. 3, pp. 115–125, 2006.

[17] ——, "The return of the sensitivity attack," in *Int. Work. Digital Watermarking*, 2005.

[18] P. Comesaña and F. Pérez-González, "Breaking the BOWS watermarking system: Key guessing and sensitivity attacks," *EURASIP Journal on Information Security*, vol. 2007, no. 2, February 2007, article ID 25308.

[19] J. Earl, "Tangential sensitivity analysis of watermarks using prior information," in *SPIE-IS&T Electronic Imaging*, vol. 6505, 2007.

[20] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A query-efficient decision-based attack," in *IEEE Symp. S&P*, 2020.

[21] A. Rahmati, S. Moosavi-Dezfooli, P. Frossard, and H. Dai, "GeoDA: a geometric framework for black-box adversarial attacks," in *CVPR*, 2020.

[22] M. Cheng, H. Zhang, C. J. Hsieh, T. Le, P. Y. Chen, and J. Yi, "Query-efficient hard-label black-box attack: An optimization-based approach," in *ICLR*, 2019.

PROOF OF PROPOSITION 1

Without loss of generality, suppose $\mathbf{n} = (1, 0, \dots, 0)$. Then:

$$\left(\hat{\mathbf{n}}^\top \mathbf{n}\right)^2 = \frac{A^2}{A^2 + B^2}, \quad A = t^{-1} \sum_{i=1}^{t} |X_i(1)|, \quad B = \|Y\|,$$

where $Y \sim \mathcal{N}(\mathbf{0}_{D-1}, \sigma^2/t\mathbf{I}_{D-1})$. Note that:

$$\mathbb{E}[A] = \sigma\sqrt{\frac{2}{\pi}}, \quad \mathbb{V}[A] = \frac{\sigma^2}{n}\left(1 - \frac{2}{\pi}\right). \tag{16}$$

Approximating the expectation as the ratio of expectations:

$$\mathbb{E}[(\hat{\mathbf{n}}^\top \mathbf{n})^2] \approx \frac{\frac{2}{\pi}\sigma^2}{\frac{2}{\pi}\sigma^2 + \sigma^2 \frac{D-1}{Q}} = \frac{1}{1 + \frac{\pi}{2}\frac{D-1}{Q}}, \forall Q \ll D$$

PROOF OF PROPOSITION 2 AND 3

In plane $\mathcal{P}$, The line (L) intersect the circle $\mathcal{C}$ in two points: $\mathbf{x_a}(i)$ and the orthogonal projection of $\mathbf{x_o}$ in $\mathcal{L}$. Thus if $\mathbf{x}^* \in \mathcal{P}$, then $\mathbf{x_a}(i+1) = \mathbf{x}^* = \mathbf{x_o} + d^*\mathbf{n}$. To do so, $\mathcal{P}$ contains $\mathbf{x_o}$ and is spanned by $\mathbf{u}(i)$ and $\mathbf{n}$. Thus, $\mathbf{v}^*(i)$ is the Gram-Schmidt orthogonalization of $\mathbf{n}$ with respect to $\mathbf{u}(i)$.

$$\mathbf{v}^*(i) = \frac{\mathbf{n} - (\mathbf{u}(i)^\top \mathbf{n})\mathbf{u}(i)}{||\mathbf{n} - (\mathbf{u}(i)^\top \mathbf{n})\mathbf{u}(i)||} = \frac{\mathbf{n} - (\mathbf{n}^\top \mathbf{u}(i))\mathbf{u}(i)}{\sqrt{1 - (\mathbf{n}^\top \mathbf{u}(i))^2}}. \tag{17}$$

We now calculate the distance $d(i+1)$ using SurFree. Consider the coordinate system with origin $\mathbf{x_o}$ and basis ($\mathbf{u}(i)$, $\mathbf{v}^*(i)$, $\mathbf{e_1}(i)$, …, $\mathbf{e_{d-1}}(i)$). The normal vector $\mathbf{n}$ writes as:

$$\mathbf{n} = \begin{bmatrix} \cos\theta(i) \\ \sin\theta(i)\cos\phi(i) \\ \vdots \\ \sin\theta(i)\sin\phi(i)\dots\sin\psi(d-2)\cos\psi(d-1) \\ \sin\theta(i)\sin\phi(i)\dots\sin\psi(d-2)\sin\psi(d-1) \end{bmatrix}$$

We have $\mathbf{x_a}(i) = d(i)\mathbf{u}(i)$, and $\mathbf{n}^\top \mathbf{u}(i) = \cos\theta(i)$. We look for $\mathbf{x_a}(i+1) = \alpha\mathbf{u}(i) + \beta\mathbf{v}(i) \in \mathcal{L}$. This implies that $\beta = -k(i)(\alpha - d(i))$. Point $\mathbf{x_a}(i+1)$ is also the closest from the origin $\mathbf{x_o}$. Minimizing $\alpha^2 + \beta^2$ yields: $\forall i$

$$d^2(i+1) = \|\mathbf{x_a}(i+1)\|^2 = d(i)^2 \frac{\cos^2\theta(i)}{\cos^2\theta(i) + \sin^2\theta(i)\cos^2\phi(i)}.$$

Knowing that $d^\star = d(i)\cos(\theta(i)), \forall i$, we obtain:

$$\cos^2\theta(i+1) = \cos^2\theta(i) + \sin^2\theta(i)\cos^2\phi(i). \tag{18}$$

PROOF OF PROPOSITION 4

In SurFree, $\mathbf{v}(i)$ is a random direction in a hyperspace of dimension $D-1$ because $\mathbf{v}(i) \perp \mathbf{u}(i)$, while $\mathbf{v}^*$ is fixed. Thus $\mathbb{E}[\mathbf{v}(i)^\top \mathbf{v}^*] = 0$ and $\mathbb{V}[\mathbf{v}(i)^\top \mathbf{v}^*] = 1/(D-1)$. In other words, $\mathbb{E}[\cos^2\phi(i)] = 1/(D-1)$.

PROOF OF PROPOSITION 5

The demonstration is fully identical to the proof of Prop. 4 but with $\eta_{\texttt{CGBA}}$. Proposition 1 shows that: $\eta_{\texttt{CGBA}} \approx \frac{1}{\frac{\pi}{2}\frac{D-1}{Q}}$ when $Q \ll D$.

PROOF OF PROPOSITION 6

$Q^*(k)$ is defined as the smallest integer Q such that:

$$\mathbb{E}[d^2_{\texttt{SurFree}}(i(1 + Q/k))] \geq \mathbb{E}[d^2_{\texttt{CGBA}}(i)]. \tag{19}$$

Eq. (9) implies: $(1 - \eta_{\texttt{SurFree}})^{i(Q/k+1)} \geq (1 - \eta_{\texttt{CGBA}})^i$. Assuming that $1 \leq Q \ll D$ leads to a second order equation independent on i whose solution is:

$$Q^*(k) \approx k\left(\frac{D\left(1 - \frac{\pi}{2k}\right) - \sqrt{D^2\left(\frac{\pi}{2k} - 1\right)^2 - \frac{2D\pi}{k}}}{2}\right) \tag{20}$$

$$\approx -\frac{k\pi}{\pi - 2k}, \text{ when } D\left(1 - \frac{\pi}{2k}\right) \to +\infty.$$