RamBoAttack: A Robust Query Efficient Deep Neural Network Decision Exploit

Viet Quoc Vo
The University of Adelaide
viet.vo@adelaide.edu.au

Ehsan Abbasnejad
The University of Adelaide
ehsan.abbasnejad@adelaide.edu.au

Damith C. Ranasinghe
The University of Adelaide
damith.ranasinghe@adelaide.edu.au

***Abstract*—Machine learning models are critically susceptible to evasion attacks from adversarial examples. Generally, adversarial examples—modified inputs deceptively similar to the original input—are constructed under whitebox access settings by adversaries with full access to the model. However, recent attacks have shown a remarkable reduction in the number of queries to craft adversarial examples using blackbox attacks. Particularly alarming is the now, *practical*, ability to exploit simply the classification decision (*hard-label only*) from a trained model's *access interface* provided by a growing number of Machine Learning as a Service (MLaaS) providers—including Google, Microsoft, IBM—and used by a plethora of applications incorporating these models. An adversary's ability to exploit *only* the predicted hard-label from a model-query to craft adversarial examples is distinguished as a *decision-based* attack.**

In our study, we first deep-dive into recent state-of-the-art decision-based attacks in ICLR and S&P to highlight the costly nature of discovering low distortion adversarial examples employing approximate gradient estimation methods. We develop a *robust* class of *query efficient* attacks capable of avoiding entrapment in a local minimum and misdirection from noisy gradients seen in gradient estimation methods. The attack method we propose, *RamBoAttack*, exploits the notion of Randomized Block Coordinate Descent to explore the hidden classifier manifold, targeting perturbations to manipulate only localized input features to address the issues of gradient estimation methods. Importantly, the *RamBoAttack* is demonstrably more robust to the different sample inputs available to an adversary and/or the targeted class. Overall, for a given target class, *RamBoAttack* is demonstrated to be more robust at achieving a lower distortion and higher attack success rate within a given query budget. We curate our results using the large-scale high-resolution `ImageNet` dataset and open-source our attack, test samples and artifacts.

I. Introduction

Demonstrations of super human performance from Machine Learning (ML) models, particularly Deep Neural Networks (DNNs), are leading to the industrialization of Machine Learning exemplified by self-driving cars [10] and MLaaS from a plethora of providers, including IBM Watson Visual Recognition [4], Amazon Rekognition [1] or Microsoft's Cognitive Services [2]. Now, at the cost-per-service level, any system can easily integrate *intelligence* into applications. The increasingly inevitable, wide spread proliferation of machine learning in systems are creating the incentives and *new* attack surfaces to exploit, for malevolent actors.

Network and Distributed Systems Security (NDSS) Symposium 2022
24-28 April 2022, San Diego, CA, USA
ISBN 1-891562-74-6
https://dx.doi.org/10.14722/ndss.2022.24200
www.ndss-symposium.org

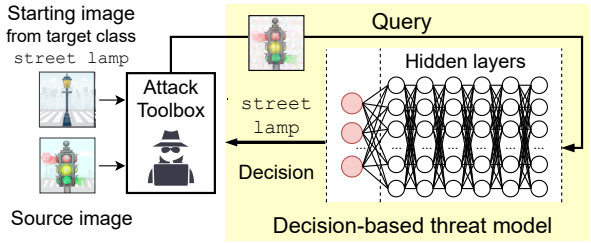

Fig. 1. An illustration of blackbox attack in the severely restricted threat model of a decision-based attack. In a decision-based threat model, an adversary with a source image and starting image from target class, crafts a sample, query the model and observe the decision returned by the model.

Adversarial Attacks in White-box Settings. In particular, machine learning models are critically vulnerable to evasion attacks from carefully crafted adversarial examples. An adversary crafts small perturbations, when added to an input, cause a failure—simply misclassifying the input in an *untargeted attack* or hijacking the decision of a model to generate a decision pre-selected by the adversary [28] in a *targeted attack*. Effective attack methods for generating adversarial examples in *white-box* attacks, assuming full knowledge and access to the machine learning models, exist [24], [21], [9], [30].

Adversarial Attacks in Blackbox Settings. In contrast, on commercial and industrial systems, an attacker has limited or no knowledge of model architecture, parameters or weights. Access may be limited to only full or partial output scores. Chen et al. [12] and Ilyas et al. [18] proposed methods to exploit models revealing output scores to craft adversarial examples under so-called *score-based attacks*. In the *most* restricted threat model illustrated in Fig. 1, the information exposed to an attacker is limited to the *hard-label* only—the most confident label predicted or *decision*, for instance logo or landmark detection on Google Cloud Vision [3].

Adversarial attacks in such a decision-based scenario are the most restrictive and challenging attack setting given the severely limited access to information, but, these settings present a realistic and pragmatic threat model.

Decision-Based (Hard-Label) Adversarial Attacks. Recent studies demonstrated the practicability of blackbox attacks under the highly restrictive decision-based attack setting relying *solely* on the label obtained from model queries. The Boundary Attack of Brendel et al. [6] in ICLR demonstrated the feasibility of an attack and obtained adversarial examples comparable with state-of-the-art white-box attack methods in both *targeted* and *untargeted* scenarios.

For a realistic attack, achieving attack success with a limited query budget is important because: i) MLaaS providers limit the rate of queries to their services; ii) throttling at a service provider limits large-scale attacks; and iii) a provider can employ methods to recognize a large number of rapid queries in succession with similar inputs to detect malicious activity and thwart query inefficient attacks. Furthermore, from both an attacker perspective and a defense perspective, reducing the number of queries *reduces the cost of mounting the attack* as well the time for evaluating the model and potential defenses[1].

A. Our Motivation and Attack Focus

Recent studies formulated the decision-based attack as an optimization problem to propose algorithms based on gradient estimation methods [14], [11] and demonstrated attacks with significantly reduced number of queries. However, the existing attacks suffer from the following problems:

- *Entrapment in a local minima.* In gradient estimation methods, as eluded to by Cheng *et al.* [14], the search for an adversarial example can experience an entrapment problem in a local minimum where extra queries expended by the attacker fails to achieve a lower distortion adversarial example.
- *Unreliabiilty of gradient estimations.* Further, as the magnitude of estimated gradients diminish on approach to a local minima or a plateau, the estimated gradients may become noisy and susceptible to misdirection.
- *Sensitivity to the starting image.* Then, intuitively, we can expect that the initialization of optimization frameworks with an *available* or intended starting image, a *necessity* in decision-based attacks, to hinder an attacker from reaching an imperceptible adversarial example. But, there is no known method to determine *a good starting image prior to an attack*. Thus, the success of an attack can be expected to be sensitive to the available starting image; an attempt to discover a better starting image or target class through *trial and error* can not only lead to detection and discovery by effectively increasing the numbers of queries needed, but also limit the scope of the attack by reducing the number of classes that can be targeted.

In general, developing decision-based attacks poses a challenging optimization problem because only binary information from output labels are available to us from the target model as opposed to output values from a function.

Therefore, we seek to understand the fragility of gradient estimation methods and to develop a more robust and query efficient attack. Consequently, we expend our efforts to answer the following research questions (RQ) .

RQ1: How can we assess the robustness of decision-based blackbox attacks to understand their fragility? (Section II-C)

RQ2: What is the impact of the source and starting target class images accessible to an adversary on the success of an attack? (Section II-D & extensive results in Section IV-D)

RQ3: How can an adversary construct a robust and query efficient attack for achieving low distortion adversarial examples for any starting image from the targeted class and avoid the pitfalls of gradient estimation based attack methods? (Section III & IV)

B. Our Contributions

In our efforts to: i) address the RQs; ii) better understand and assess the vulnerabilities of DNNs to adversarial attacks in the pragmatic decision-based threat model; and iii) explore more robust attacks, we summarize our contributions below:

1) Our study presents the *first* systematic investigation of state-of-the-art decision-based attacks to understand their robustness. Through extensive experiments, we highlight the problem of *hard* cases where attackers struggle to flip the prediction of images towards a target class, even with increasing query budgets–see Fig. 2.
2) Motivated by our findings, *we propose a new attack*—RamBoAttack—that is demonstrably more robust. We propose a search algorithm analogous to Randomized Block Coordinate Descent—BlockDescent—to address the entrapment problem where gradient estimation fails to provide a useful direction to descend and propose to combine BlockDescent with gradient estimation frameworks to attain query efficiency. In contrast to existing approaches, BlockDescent focuses on altering local regions of the input commensurate with the filter sizes employed by DNNs to forge adversarial examples.
3) We provide new insights into query efficient mechanisms for crafting adversarial perturbation to attack DNNs. Our decision-based blackbox attack method relying on localized alterations to inputs discovers effective adversarial perturbations attempting to exploit the model's reliance on salient features of the target class to correctly classify an input to a target label in the *hard* cases. We illustrate clear correlations between perturbations found and added to inputs, and salient regions on target class images with the aid of a visual explanation tool.
4) Overall, RamBoAttack is a more robust and query efficient approach for generating an adversarial example of high attack success rate compared to existing counterparts. Importantly, our attack method is *significantly less impacted by a starting image* from a target class accessible to an adversary.
5) Recognizing the need for reliable and reproducible evaluation strategies, we introduce two evaluation protocols applied across `CIFAR10` and `ImageNet`. We *release the datasets constructed for our extensive study to support future benchmarking of blackbox attacks* under a decision-based setting—*https://ramboattack.github.io/*.

[1]For example, we consumed over 1,700 hours on two dedicated modern GPUs with 48 GB memory to curate the results in our study.

II. Decision-Based Attacks

In this section, we: i) formalize an adversarial attack as an optimization problem; ii) revisit current state-of-the-art methods; and iii) analyze the results to present some intuitions into state-of-the-art attacks based on our observations.

A. Adversarial Threat Model

We adopt the threat model proposed in prior works [11], [15], [6]. Under the decision-based blackbox setting, adversaries have no prior knowledge such as model architecture or parameters but have limited access to the output of a victim model—*the model's decision* as illustrated in Fig. 1. Furthermore, an adversary can make numerous queries to a victim's machine learning model via an access interface and receive the model's decision. The adversary must have *at least one image from a target class that is classified correctly by the victim model if the adversary aims to carry out a targeted attack*. This image is the *starting image* used to initialize the attack. The adversary also holds at least one image from a *source class* correctly classified by the model. The *objective* of the adversary is to discover the minimum (imperceptible) perturbation—quantitatively measured by the common distortion measure adopted in recent studies—to flip the decision for the *source image* to the targeted class using the minimum number of queries to the model.

B. Problem Formulation

Given a source image $\boldsymbol{x} \in \mathbb{R}^{C\times W\times H}$ its ground truth label y from the label set $\mathcal{Y} = \{1, 2, \cdots, K\}$, where K denote the number of classes, C, W and H denotes the number of channels, width and height of an image, respectively. Given a pre-trained multi-class classification model $f : \boldsymbol{x} \rightarrow y$ so that $f(\boldsymbol{x}) = y$, in a targeted attack, an adversary aims to modify an input $\boldsymbol{x}$ to craft an optimal adversarial example $\boldsymbol{x^*} \in \mathbb{R}^{C\times W\times H}$ that is classified as the class label desired by the adversary when used as an input for the victim model. In an untargeted attack, an adversary manipulates input $\boldsymbol{x}$ to change the decision of a classifier to any class label other than its ground-truth label. To simplify the descriptions, we refer to the desired class label as the *target class* while the class of the input $\boldsymbol{x}$ is called the *source class*.

Measuring Distortion. l_2-norm is widely adopted, in *all of the recent works* as in [6], [7], [13], [14], [12], to measure the distortion and similarity between a generated adversarial example and the source sample. Therefore, in this paper, our approach focuses on l_2-norm. Then, let $D(\boldsymbol{x}, \boldsymbol{x^*})$ be the l_2-distance that measures the similarity between $\boldsymbol{x}$ and $\boldsymbol{x^*}$.

Optimization Problem. The main aim of adversarial attacks is to minimize the distortion measured by D while ensuring the perturbed input data is classified as a target class—for a targeted attack—or non-source class—for an untargeted attack. Therefore, an adversarial attack can be formulated as a constrained optimization problem:

$$\begin{aligned} \min_{x^*} \quad & D(\boldsymbol{x}, \boldsymbol{x^*}) \\ \text{s.t.} \quad & \mathcal{C}(f(\boldsymbol{x^*})) = 1, \\ & \boldsymbol{x}, \boldsymbol{x^*} \in [0, 1]^{C\times W\times H}, \end{aligned} \tag{1}$$

Here, $\mathcal{C}(f(\boldsymbol{x^*}))$ is an adversarial criterion that takes the value 1 if the attack requirement is satisfied and 0 otherwise. This requirement is satisfied if $f(\boldsymbol{x^*}) \neq y$ for an untargeted attack or $f(\boldsymbol{x^*}) = y^*$ for a targeted attack (i.e. for the instance $\boldsymbol{x^*}$ to be misclassified as targeted class label y^*).

C. Understanding Robustness

The two current query efficient attack methods employ gradient approximation frameworks, whilst the earlier method relied on a stochastic approach. We briefly summarize these methods before delving into our systematic study to understand their robustness.

Random Walk along a Decision Boundary. The first attack under a decision-based threat model proposed by Brendel et al. [6] initialized an image in a target class and in each iteration, sampled a perturbation from a Gaussian distribution and projected the perturbation onto a sphere around a source image. If this perturbation yields an adversarial example, the attack makes a small movement towards the source image and repeats these steps until the decision boundary is reached. Subsequently, by traveling along the decision boundary based on sampling, projecting and moving towards the source image, the adversarial example is refined until an adversarial example with a desirable distortion is discovered.

Optimization Frameworks. In the absence of a means for computing the gradient for solving (1), the attacks in [13] and [14] attempt to solve the optimization problem using methods to estimate the gradient. Cheng et al. [13] samples directions from a Gaussian distribution and applies a zeroth-order gradient estimation method in their OPT-attack, then Cheng et al. [14] leveraged their former optimization framework and proposed a zeroth-order optimization algorithm called Sign-OPT that is much faster to converge. Chen et al. [11] introduced a different optimization framework named HopSkipJumpAttack using a Monte Carlo method to find the approximate gradient direction to descend.

Evaluating Robustness. To understand the robustness of recent attack methods and illustrate the costly nature of discovering low distortion adversarial examples with these attacks, we propose an *exhaustive but tractable* experiment using the relatively small number of classes albeit with a significantly large validation set offered by `CIFAR10` dataset. The protocol for assessing robustness of each state-of-the-art method described is carefully described in Appendix B.

Hard Cases. Empirically, we define a *hard* case as a pair of source and starting images—the starting image is from a given target class—where a given decision-based attack fails to yield an adversarial example with a distortion below a desirable threshold using a set query budget.

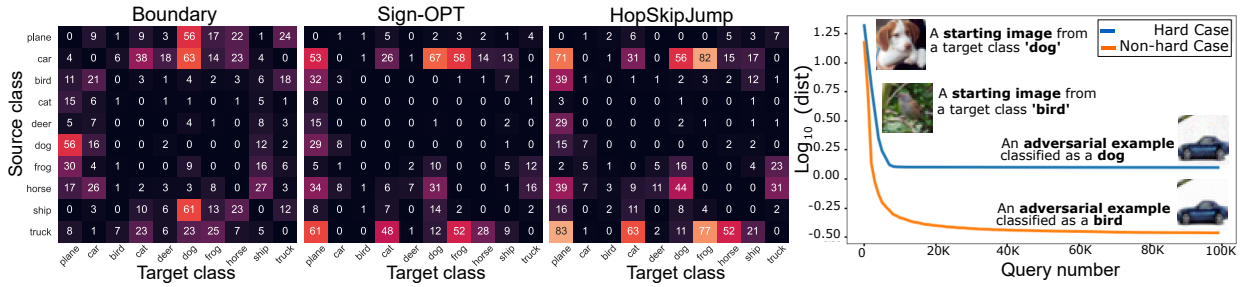

Fig. 2. (Left) The number of *hard* cases from `CIFAR10` found for Boundary Attack (BA), Sign-OPT and HopSkipJump categorized by different source and target class (starting image) pairs at a distortion threshold = 0.9 and a 50,000 query budget. (Right) The line chart shows a significant difference between a *hard* versus *non-hard* case; *interestingly, increasing the query budget to even 100,000 does not yield a lower distortion adversarial example for the hard case*.

D. Observations from Assessing Attacks

We make the following observations from our results summarized in Figures 2 and 3.

Observation 1: Hard Cases. *In decision-based attacks, specific classes and/or samples from classes are more difficult to attack than others.* As illustrated in Fig. 2(left), the current attack methods are not uniformly effective against all pairs of source and starting images from target classes.

Interestingly, any of the gradient estimation methods can approximate the true gradient given enough queries (or samples) to the target model. However, solutions can become entrapped in various local minima. Further, approaching a local minimum or a plateau can considerably undermine the quality of that approximation; for instance, estimated gradients may become noisy when the gradient magnitude diminishes whilst approaching a local minimum. As shown in Fig. 2 (right), even with 100K queries, the solutions based on the gradient direction estimation methods do not improve the distortion of the adversarial sample for the `car` classified as a `dog` (Hard case).

Observation 2: Attack initialization. *An attack algorithm's ability to find a low distortion adversarial example with a given query budget is dependent on the starting image from a target class selected for initializing the attack algorithm.* Interestingly, Chen et al. [11] in their S&P2020 paper briefly noted the potential for an algorithm to get trapped in a bad local minimum based on the starting image used to initialize an attack. Our systematic study confirms this intuition.

In this case, the achievable distortion of an adversarial example is highly dependent on the starting image and the behavior of the algorithm. This observation is illustrated by comparing the results of starting *image 1* with *image 2* for different attack methods in Fig. 3 and by 100 samples randomly selected from the *hard* set of each method—see Section IV-C and IV-D for more details. *The results demonstrates the dependence of attack success on the starting image accessible to an adversary*.

Currently, there is no effective initialization method to determine a good starting image, prior to mounting an attack. Therefore, developing robust attack that is less sensitive to the choice of starting image remains an open challenge.

Observation 3: Perturbation Region. *Current attack approaches aim to perturb the whole image to traverse the decision boundary to find an adversarial example with minimum distortion.* In other words, these methods always manipulate the whole image at a time and result in perturbations that is spread over the entire image as illustrated by perturbation heat maps in Fig. 3. Another interesting remark drawn from these figures is that the main features (for example edges) of the starting image remains super-imposed in an adversarial example. However, most of the state-of-the-art classifiers in computer vision utilize convolutional filters to extract local patterns in an image; further, visual explanation tools demonstrate the reliance of classifiers on key salient features of an image. *Therefore, whether an attack could achieve a lower distortion adversarial example by targeting the filter operation over local features in contrast to manipulation of the whole image remains an interesting direction to explore*.

E. An Intuition into Attack Methods

To understand and illustrate the underlying cause of the first two observations, we use Boundary attack (BA) [6], Sign-OPT [14] and HopSkipJump [11] to attack a Toy model. The *decision boundary* of the Toy model in a 2D *input space* illustrates a constraint of the optimization problem in (1). This decision boundary is represented by $g(z_1, z_2) = (z_1 - 2)(z_1 - 1)^2(z_1 + 1)^3 - z_2 = 0$ where z_1 and z_2 denote two coordinates of a point such as a starting point or a source point as illustrated in Fig. 4. A point above the boundary is classified as in target class; otherwise, it belongs to the source class. The black dot (•) *source point* denotes a source class example whilst black dot (•) *starting point* denotes a starting target class example. All three methods are initialized with the same starting point, we then employ the attacks to search for an adversarial point within the target class and closest to the source point; alternatively, we aim to solve the optimization problem in (1), where the objective is to minimize the l_2 distance to the source point subject to the constraint imposed by the decision boundary, using these attack algorithms.

Fig. 4 illustrates several intermediate adversarial example points denoted by blue dots and a final adversarial example achieved by each method denoted by a yellow triangle for one example attack execution. Given the stochastic nature of the

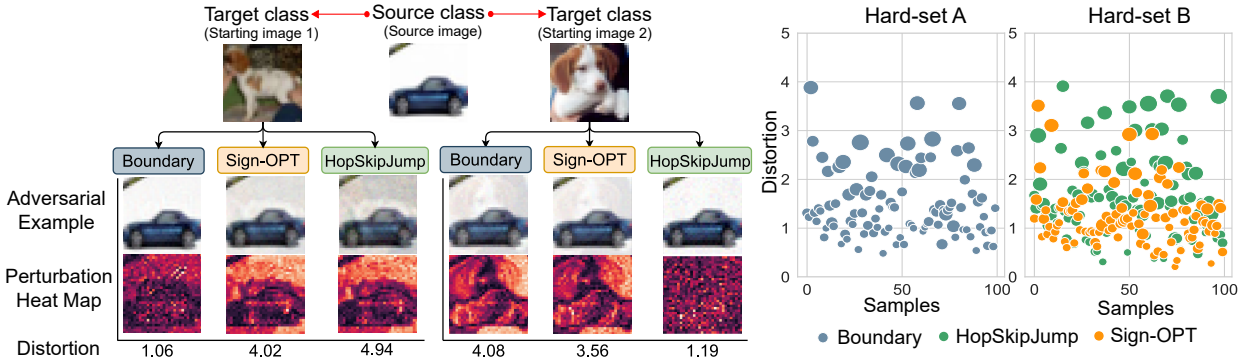

Fig. 3. (Left) Consider an attack to discover an adversarial example for the source image of class `car` such that the car is predicted as belonging to the target class `dog`. We demonstrate the very different results an adversary can obtain based on the availability of a target class *Starting image1* and *Starting image2*. For Boundary, initializing the attack with *Starting image1* is better. In contrast, Sign-OPT and HopSkipJump discover better adversarial examples if an adversary initialized them with *Starting image2*. (Right) The scatter plot illustrates this attack scenario with 100 samples randomly selected from their own *hard* set. The y-axis denotes the average distortion and the size of each bubble denotes the variation in distortion for each source image with respect to 10 different starting images from hard targets. It shows that all the attack methods are highly dependent on a starting image in *hard* cases.

algorithms, we execute each attack 100,000 times with different random seeds. All of the methods, except HopSkipJump fails to find the optimal solution—global minimum—and HopSkipJump only managed to reach the optimal solution in 2.5 % of the attempts. As illustrated in Fig. 4, the approximate gradient appears to be noisy and the methods traverses the decision boundary in an incorrect direction towards the local minimum rather than the global minimum. Although not illustrated here, changing the starting coordinate can lead all of these methods to discover the global minimum.

III. Proposed Attack Framework

We observe that: i) gradient estimation methods in attacks face an entrapment problem in a highly complex loss landscape; ii) current attacks focus on altering all of the coordinates of an image simultaneously to forge a perturbation; and iii) the success of current attacks are sensitive to the chosen or available starting image possessed by an adversary.

We propose an analogous Randomized Block Coordinate Descent method—named BlockDescent—that aims to manipulate local features and target convolutional filter outputs by modifying values of coordinates in a square-block region and in different color channels with targeted perturbations. We propose localized changes to affect convolutional filter outputs and pixel values as a means of impacting on salient features and may be even mimic salient features of the target. This leads to potential redirection and escape from entrapment in a bad local minimum with minimal but effective changes to the image to mislead the classifier. In other words, we propose taking a direct path along some coordinates towards a source image whilst retaining the target class label to prevent the problem encountered by gradient estimation methods—entrapment in a local minimum as shown in Fig.4.

Further, when employing gradient estimation methods, the gradient values decrease as we move closer to the source image leading to increasingly larger number of perturbations needed to converge. This issue is exacerbated if there is a plateau in the decision boundary; now the gradient estimation methods are as effective as a random search. We conjecture that the *hard* cases are examples of where the gradient of the distortions are generally small and, thus, leads to a local optima. However, we observe that the gradient estimation methods are effective

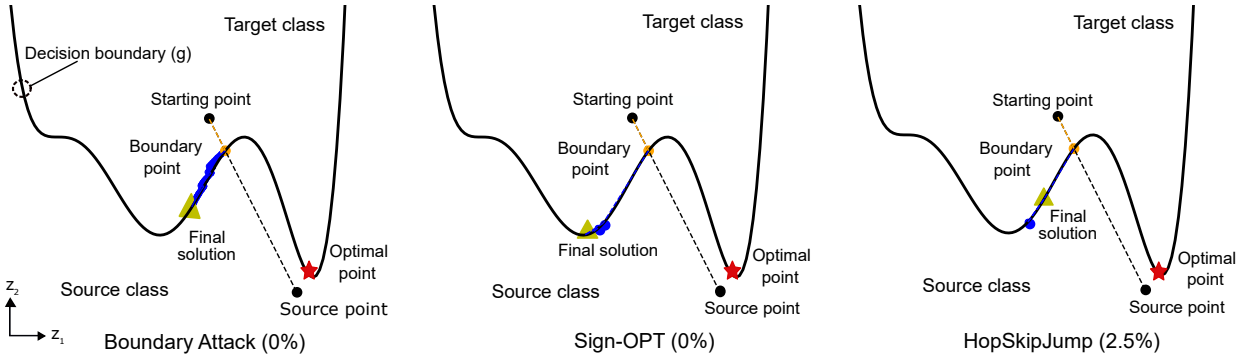

Fig. 4. **2D (z_1 and z_2) Input Space Example.** An illustration of the execution of the three different decision based attack methods (Boundary, Sign-OPT and HopSkipJump) to attack a toy model employing 2D inputs. The attacks result in different final solutions denoted by a yellow triangle (▲). We executed the algorithms 100,000 times; both Boundary Attack and Sign-OPT failed to find the global minimum (the Optimal Point *closest* to the Source point) and HopSkipJump only found the global minimum 2.5% of the time. This illustrates the *problem faced by current attack methods* when attacking a machine learning model *whose decision boundary in the input space is multi-dimensional and highly complex* for realistic and practical image inputs.

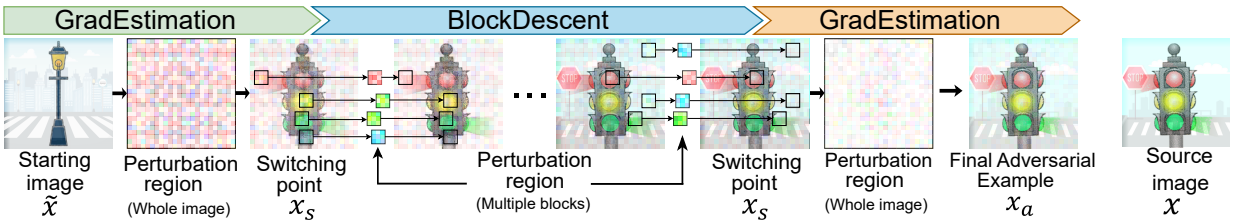

Fig. 5. A ***pictorial illustration*** of RamBoAttack to craft an adversarial example. In a targeted attack, the first component (GradEstimation) initializes an attack with a starting image from a target class (e.g. we use a clip art `street lamp` for illustration) and then manipulates this image to search for adversarial examples that looks like an image from source class e.g `traffic light`. The attack switches to the second component, BlockDescent, when it reaches its own local minimum. BlockDescent helps to redirect away from that local minimum by manipulating multiple blocks—or making local changes to the current adversarial example. Subsequently, the adversarial example crafted by BlockDescent is refined by the third component (GradEstimation).

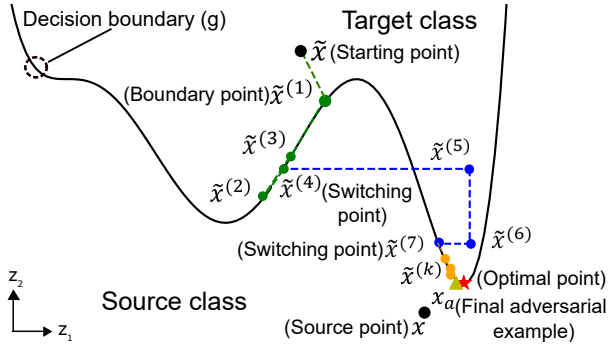

Fig. 6. **2D (z_1 and z_2) Input Space Example.** An illustration of our RamBoAttack against the toy model in Fig. 4. If the first gradient estimation method—GradEstimation in Algorithm 1—leads to entrapment in a local minimum—denoted by $\tilde{\boldsymbol{x}}^{(1)}, \cdots, \tilde{\boldsymbol{x}}^{(4)}$ at the start—there is no effective mechanism to escape. However BlockDescent moves away from the local minimum. This is illustrated by $\tilde{\boldsymbol{x}}^{(5)}, \cdots, \tilde{\boldsymbol{x}}^{(7)}$ when the number of modified coordinates is one in the 2D input space. Subsequently, the third component applying a gradient estimation method searches for a better adversarial example $\tilde{\boldsymbol{x}}^{(k)}$ in the neighborhood region and reaches the nearly optimal solution $\boldsymbol{x}_\mathrm{a}$. In contrast to results in Fig. 4, when evaluating RamBoAttack over 100,000 runs against the Toy model, we observed our attack to ***always*** find the optimal or near optimal solution.

in two cases: **(a)** initial stages of optimizing Eq. (1) or **(b)** at close proximity to the source image. In (a), the gradients are sufficiently large to be estimated effectively, and in (b) small changes and refinements (*i.e.* few perturbation iterations) facilitate a decent to the optimum.

Consequently, we propose a new framework using gradient estimation for the initial descent—case (a)—supported by BlockDescent to escape entrapment and noisy gradient problems and refining the adversarial example supported by a gradient estimation based descent to forge a robust and query efficient attack. Importantly, BlockDescent is *insensitive to the choice of starting images*, although it is effectively initialized with a gradient estimation, because BlockDescent manipulates blocks that causes a move away from the direction set by a starting images. The new framework we propose, RamBoAttack, is illustrated in Fig. 5.

Summary. *Gradient estimation methods are fast but face the potential problem of getting trapped in a bad local minimum, particularly in hard cases. BlockDescent, on the other hand, is slower—selecting to manipulate local regions—but is capable of tackling the problems faced by gradient estimation attacks. Therefore, we develop a hybrid framework called RamBoAttack for query efficient decision-based attacks that can exploit the merits of both approaches. In particular, our derivative-free optimization method considers, for the first time, an approach to manipulate blocks of coordinates in the input image to influence the outcome of convolution operations used in deep neural networks as a means for misguiding a networks decision and generating adversarial examples with minimal manipulations.*

A. Approach

Our proposed attack thus comprises of BlockDescent and two components of gradient estimation—GradEstimation—as shown in Fig. 5 and described in Algorithm 1. The gradient estimation algorithms used by these two components can be the same or different from each other. When starting an attack, particularly in targeted setting, the first component is initialized with a starting image $\tilde{\boldsymbol{x}}$ from a target class and approaches the decision boundary via a binary search—the first step in a gradient estimation method. We employ the gradient estimation method to search for adversarial examples until reaching its own local minimum. We call it a switching point $\boldsymbol{x}_\mathrm{s}$ because from this point, gradient estimation method switches to BlockDescent. If the gradient estimation method is entrapped in local minimum, BlockDescent helps to move away from that local minimum. Subsequently, when local changes are insufficient, the attack switches to the third component to refine the adversarial example crafted by BlockDescent which is considered as the second switching point. This refinement aims to search for the final adversarial example $\boldsymbol{x}_\mathrm{a}$ with a lower distortion.

Fig. 6 illustrates RamBoAttack against the Toy model used in Section II-E and demonstrates the effectiveness of the attack we propose. Particularly, the first gradient estimation approach searches for and reaches the adversarial examples $\tilde{\boldsymbol{x}}^{(1)}$, $\tilde{\boldsymbol{x}}^{(2)}$, $\tilde{\boldsymbol{x}}^{(3)}$ at different steps towards approaching the source point but is stuck at $\tilde{\boldsymbol{x}}^{(4)}$ which is a local minimum of the objective function $D(\boldsymbol{x}, \boldsymbol{x}^*)$ subject to the constraint defined by the decision boundary $g(z_1, z_2)$. Henceforth, BlockDescent searches for next adversarial examples $\tilde{\boldsymbol{x}}^{(5)}, \cdots, \tilde{\boldsymbol{x}}^{(7)}$ by modifying one coordinate at a time—in this 2D example—by applying δ changes. Subsequently, the second gradient es-

Algorithm 1: RamBoAttack

Input: source image $\boldsymbol{x}$, starting image $\tilde{\boldsymbol{x}}$, model f
gradient estimation function $g_1,\ g_2$, reduction scale λ,
input dimensions N,square extension n,
block number m, query number T_1, T_2

$\boldsymbol{x}_\text{s} \leftarrow$ GradEstimation $(\boldsymbol{x}, \tilde{\boldsymbol{x}}, f, g_1, T_1)$
$\boldsymbol{x}_\text{s} \leftarrow$ BlockDescent $(\boldsymbol{x}, \boldsymbol{x}_\text{s}, f, \lambda, N, n, m)$
$\boldsymbol{x}_\text{a} \leftarrow$ GradEstimation $(\boldsymbol{x}, \boldsymbol{x}_\text{s}, f, g_2, T_2)$
return $\boldsymbol{x}_a$

Algorithm 2: GradEstimation

Input: source image $\boldsymbol{x}$, switching image $\boldsymbol{x}_\text{s}$, model f
gradient estimation function g, query number T

$n_\text{q} \leftarrow 0$, $switch \leftarrow False$
$d \leftarrow D(\boldsymbol{x}, \boldsymbol{x}')$
while *not* $(switch)$ **do**
  $\boldsymbol{x}',\ i \leftarrow g(f, \boldsymbol{x}, \boldsymbol{x}')$
  $n_\text{q} \leftarrow n_\text{q} + i$
  if $n_q\ >\ T$ **then**
    $\Delta \leftarrow d - D(\boldsymbol{x}, \boldsymbol{x}')$, $d \leftarrow D(\boldsymbol{x}, \boldsymbol{x}')$
    $n_\text{q} \leftarrow 0$
    if $\Delta < \epsilon_s$ **then**
      $switch \leftarrow True$
end while
return $\boldsymbol{x}'$

Algorithm 3: BlockDescent

Input: source image $\boldsymbol{x}$, switching image $\boldsymbol{x}_\text{s}$,model f
reduction scale λ, input dimension N
square extension n, block number m

$k \leftarrow 0$, $n_\text{q} \leftarrow 0$, $switch \leftarrow False$
$\delta \leftarrow P_\text{i}(|\boldsymbol{x} - \boldsymbol{x}_\text{s}|)$, $\tilde{\boldsymbol{x}}^{(k)} \leftarrow \boldsymbol{x}_\text{s}$, $D_{n_\text{q}} \leftarrow D(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{(k)})$
while *not (switch)* **do**
  $j \leftarrow 0$
  while $j <\ N$ ***and*** *not* $(switch)$ **do**
    /* Craft a new sample */
    $\boldsymbol{x}' \leftarrow \tilde{\boldsymbol{x}}^{(k)}$
    for $t = 1, 2, \cdots, m$ **do**
      Uniformly select a set $\{c, w, h\}$ at random without replacement
      $\boldsymbol{x}'_{\text{B}_\text{t}} \leftarrow \boldsymbol{x}'_{[c,w-n:w+n,h-n:h+n]}$
      $\boldsymbol{x}_{\text{B}_\text{t}} \leftarrow \boldsymbol{x}_{[c,w-n:w+n,h-n:h+n]}$
      /* Perturbation region */
      $M \leftarrow \text{sign}(\boldsymbol{x}_{\text{B}_\text{t}} - \boldsymbol{x}'_{\text{B}_\text{t}})$
      $\boldsymbol{x}'_{\text{B}_\text{t}} \leftarrow \boldsymbol{x}'_{\text{B}_\text{t}} + M \times \delta$
    end for
    /* Evaluate crafted sample */
    if $D_{n_q} > D(\boldsymbol{x}, \boldsymbol{x}')$ **then**
      $n_\text{q} \leftarrow n_\text{q} + 1$
      if $\mathcal{C}(f(\boldsymbol{x}')) = 1$ **then**
        $\tilde{\boldsymbol{x}}^{(k+1)} \leftarrow \boldsymbol{x}'$
        $k \leftarrow k + 1$
      $D_{n_\text{q}} \leftarrow D(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{(k)})$
      Compute Δ using Equation 3
      if $\Delta < \epsilon_s$ **then**
        $switch \leftarrow True$
    $j \leftarrow j + m$
  end while
  $\delta \leftarrow \frac{\delta}{\lambda}$
end while
return $\tilde{\boldsymbol{x}}^{(k)}$

timation method continues searching for adversarial examples $\tilde{\boldsymbol{x}}^{(k)}$ in the neighborhood areas until reaching the near optimal $\boldsymbol{x}_\text{a}$. Most importantly, in contrast to experiment in Fig. 4 when evaluating RamBoAttack over 100,000 attacks on the Toy model, our proposed attack always reached the optimal or near optimal solution.

When to switch to BlockDescent? The gradient estimation methods are designed to work alone rather than with other methods. Therefore, we develop a sub-module GradEstimation to call these methods and determine when to switch from a gradient estimation method to BlockDescent. Empirically, gradient estimation methods reach their local minimum when they cannot find any better adversarial example after several steps of searching. In practice, this can be determined by the distortion reduction rate Δ after every T queries—a time frame to calculate Δ. However, in gradient estimation methods, the number of queries per iteration is varied so we relax this by accumulating the number of queries after each iteration. Whenever it exceeds T, we compute Δ and if this distortion reduction rate is below a switching threshold ϵ_s, it switches to BlockDescent (see Algorithm 2).

B. BlockDescent

We recognize that the architecture of most machine learning models in computer vision is based on a Convolutional Neural Network (CNN) built on convolution operations. These convolution operations are defined as $c \times q \times q$ where q is the size of the filter and c is the number of channels to extract local patterns of an image. Consequently, we hypothesize that altering a block of coordinates as a square-shaped region with an appropriate size can target significant filter outputs potentially having a significant impact on the network's decision. Perturbing these coordinates can result in an adversarial example with fewer queries since we target regions of the input to impact actual convolutional filters and potentially discover salient features to mimic. Inspired by this, we adopt a notion of square-block perturbation regions and introduce BlockDescent that manipulates blocks of size n. BlockDescent has two stages: i) crafting a sample; and ii) its evaluation as described in Algorithm 3.

Crafting a Sample. In each iteration, the first stage of BlockDescent aims to yield a sample $\boldsymbol{x}'$ that is initialized with $\boldsymbol{x}^{(k)}$ which is an adversarial example at k-th step. To increase convergent rate and reduce query number, BlockDe-

scent modifies several blocks of coordinates concurrently. It firstly selects m different coordinates across different channels (R, G, B) of an image by choosing a set $S = \{S_1, S_2, \cdots, S_{\mathrm{m}}\}$ where $S_{\mathrm{t}} = \{c_{\mathrm{t}}, w_{\mathrm{t}}, h_{\mathrm{t}}\}$ is selected uniformly at random such that $c_{\mathrm{t}} \in [1, C]$, $w_{\mathrm{t}} \in [1, W]$ and $h_{\mathrm{t}} \in [1, H]$, where $t = 1, 2, \cdots, m$ and C, W, H denote the number of channel, width and height of an image. This random selection is sampling without replacement and each selected coordinate $x'_{\mathrm{c,w,h}}$ is a center of a square block $\boldsymbol{x}'_{\mathrm{B_t}}$, where $\boldsymbol{x}'_{\mathrm{B_t}}$ represents $\boldsymbol{x}'_{[c_{\mathrm{t}}, w_{\mathrm{t}}-n:w_{\mathrm{t}}+n, h_{\mathrm{t}}-n:h_{\mathrm{t}}+n]}$. Likewise, m corresponding blocks $\boldsymbol{x}_{\mathrm{B_t}}$ are yielded from the source image $\boldsymbol{x}$. A mask M with the same size as $\boldsymbol{x}'_{\mathrm{B_t}}$ can be defined as $M = sign(\boldsymbol{x}_{\mathrm{B_t}} - \boldsymbol{x}'_{\mathrm{B_t}})$. This mask is used to identify the direction of perturbation for each element of a block $\boldsymbol{x}'_{\mathrm{B_t}}$. When each element of a block which is a coordinate of an image is manipulated to move along this direction, it tends to move towards to its corresponding element in the source image. The sample $\boldsymbol{x}'$ is crafted when each of m blocks of coordinates is updated as below:

$$\boldsymbol{x}'_{\mathrm{B_t}} \leftarrow \boldsymbol{x}'_{\mathrm{B_t}} + M \times \delta \tag{2}$$

where δ is a scalar which denotes an amount of perturbation for each element and it reduces by λ after each cycle. One cycle is ended when all coordinates are selected for perturbation. If δ is initialized with a small value, it is slow convergent and results in query inefficiency from the beginning. Whilst, for large initial δ, modifying blocks of coordinate almost leads to a sample moving further from the source image from beginning rather than moving closer. Consequently, it requires several cycles until δ reduces to a suitable value. To tackle this issue, we exploit the distribution of the absolute difference between all coordinates of a sample and their corresponding coordinate in a source image and use i-th percentile P_{i} of this distribution to specify a proper initial δ. In Equation 2, only selected square blocks are perturbed while the rest of $\tilde{\boldsymbol{x}}$ remains unchanged.

Evaluate Crafted Sample. In the second stage, to ensure a descent of distortion and improve query efficiency, a crafted sample $\boldsymbol{x}'$ is only evaluated by the victim model if it moves closer to $\boldsymbol{x}$. If the adversarial criteria is then satisfied ($\mathcal{C}(f(\boldsymbol{x}')) = 1$), the perturbation will make a change to update the next adversarial example as $\tilde{\boldsymbol{x}}^{(k+1)} = \boldsymbol{x}'$. Otherwise the perturbation will be discarded.

Determining When to Switch to the Next Component. Similar to the switching criterion of gradient estimation methods, BlockDescent should switch to the next component when it cannot find any better adversarial example that can be empirically measured by distortion reduction rate Δ per T queries. However, we observe that BlockDescent is a gradient-free optimization so Δ is highly varied for each subsequent query. As such we cannot simply apply the same criterion as gradient estimation methods. Consequently, to determine a better switching criterion for BlockDescent, we adopt a smoothing technique based on Simple Moving Average to measure the distortion reduction rate Δ. In practice, Δ is computed as follows:

$$\Delta \leftarrow \frac{1}{T} \sum_{l=n_q-2T}^{n_q-T} (D_l - D_{(l+T)}) \tag{3}$$

where D_{l} is a distance between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}^{(k)}$ at query l, n_{q} is n_{q}-th query. If Δ is smaller than a switching threshold ϵ_{s}, BlockDescent switch to the next component.

IV. Experiments and Evaluations

A. Experiment Settings and Summary

Attacks and Datasets. In this section, we evaluate the effectiveness of our RamBoAttack versus current state-of-the-art attacks—Boundary attack (Boundary) [6], Sign-OPT [14] and HopSkipJump [11] on two standard datasets: `CIFAR10` [19] and `ImageNet` [16]. All hyper-parameters of our RamBoAttack are described in Appendix A and all of the evaluation sets are described in Section IV-B, IV-C, Appendix B and C.

Models. For a fair comparison, for `CIFAR10`, we used the CNN architecture used by Cheng et al. [13], [14] comprising of of four convolutional layers, two max-pooling layers and two fully connected layers. For evaluation on `ImageNet`, we use a pre-trained ResNet-50 [17] provided by torchvision [22] which obtains a 76.15% Top-1 test accuracy. In addition, all images are normalized into pixel scale of $[0, 1]$.

Evaluation Measures. To evaluate the performance of method, prior works use different metrics such as a score based on the median squared l_2-norm [6] and median l_2-norm distortion versus the number of queries [14], [11]. However, median metric is not able to highlight the existence of the so-called *hard* cases and their impact on the performance of an attack so the evaluation may be less reliable. Therefore, in addition to median, we report average l_2-norm distortion. We also adopt Attack Success Rate (ASR) used in [11] to measures the attack success of crafted adversarial samples, obtained with a given query budget, at various distortion limits.

Gradient Estimation Selection for RamBoAttack. We apply two state-of-the-art gradient estimation methods, HopSkipJump and Sign-OPT, and derive two RamBoAttack attacks: i) RamBoAttack (HSJA), composed of HopSkipJump, BlockDescent and Sign-OPT; and ii) RamBoAttack (SOPT), composed of Sign-OPT and BlockDescent. We do not use HopSkipJump for the second gradient descent stage because we observe Sign-OPT to be more effective at refining adversarial samples—as also observed in [14].

Experimental Regime. We summarize the extensive experiments conducted with `CIFA10` and `ImageNet` datasets. All experiments are performed on one RTX TITAN GPU and one 2080Ti GPU. The total running time for all experiments is approximately 1,826 hours.

- *Robustness of RamBoAttack:* Given the observations in Section II-D, we aim to investigate the robustness of our RamBoAttack by assessing the existence of a *hard* set for our RamBoAttack. We execute the exhaustive evaluation protocol used in Section II-D and compare results with state-of-the-art attacks in Section IV-B.

- *Attacking Hard Sets:* Most attacks demonstrate impressive performance in *non-hard* cases whilst struggling with *hard* cases. Therefore, we compare and demonstrate the performance differences—in terms of query efficiency, attack success rate and distortion—that exists on *hard* evaluation sets in Section IV-C.
- *Impact of the Starting Image:* We observed the impact of the starting image from the target class on the success of the attack in Section II-D. Hence, the exhaustive experimental evaluations in Section IV-D explores the sensitivity of an attack's success to the choice of the attacker's starting image. An important consideration to evade detection when trial-and-error testing of starting images are needed to find easy samples or when access to samples (source or target class) are restricted.
- *Attack Insights:* We observed clear correlations between perturbations yielded by our RamBoAttack and salient regions of target images embedded inconspicuously in adversarial examples. Section IV-E investigate these artifacts resulting from the localized perturbation method in BlockDescent.
- *Attacks Against Defended Models:* Decision-based attacks are able to fool standard models. This naturally leads to the critical question of whether or not such attacks are able to bypass defensed models. Thus, the experiments in Section IV-F aim to investigate the robustness of decision-based attacks against defense mechanisms.
- *Validation on Balance Datasets:* Constructing *hard* and *non-hard* sets for all decision-based attack methods through exhaustive evaluations to assess robustness is extremely time consuming. Therefore, we propose a reliable and reproducible attack evaluation strategy to validate robustness. We differ the proposed evaluation protocol and results to Appendix C and release all of the constructed sets for comparisons in future studies.
- *Untargeted Attack Validation:* In addition to targeted attacks, for completeness, we evaluate our RamBoAttack and other state-of-the-art attacks on `CIFAR10` and `ImageNet` under the untargeted attack setting. We defer these results to Appendix D.

B. Robustness of RamBoAttack

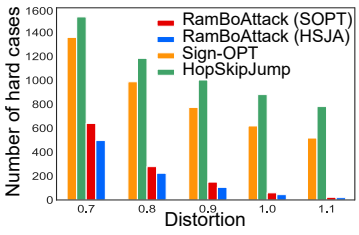

Fig. 7. The number of *hard* cases found for Sign-OPT, HopSkipJump and RamBoAttack over a range of distortion thresholds with a budget of 50,000 queries (detailed results are in Appendix I & Fig. 32).

We carry out a comprehensive experiment, similar to that in Section II-D. In this experiment we use a range of distortion threshold of 0.7 to 1.1. Notably, both [11] and [14] reported their methods to achieve a distortion level below 0.3 after 10,000 queries; hence our proposed values are not guaranteed to discover *hard* cases because the smallest value, 0.7, is much higher than 0.3 achieved in other studies. The main aim is to illustrate how our RamBoAttacks are able to craft more adversarial example with distortions below a range of distortions from 0.7 to 1.1 for each sample of the entire `CIFAR10` test set. We compare the performance of the RamBoAttack with Sign-OPT and HopSkipJump. Fig. 7 shows a remarkably low number of *hard* cases for the RamBoAttack. The total number of hard cases achieved for our RamBoAttack is approximately 10 times lower for the distortion ranges from 0.9 to 1.1. For distortion at 0.7 and 0.8, the number of *hard* cases drops approximately 2 times and 5 times, respectively in comparison with the other attack methods. Interestingly, as expected, *hard* pairs encountered by Sign-OPT and HopSkipJump are resolved with RamBoAttack as shown in Appendix I—see Fig. 32.

C. Attacking Hard Sets

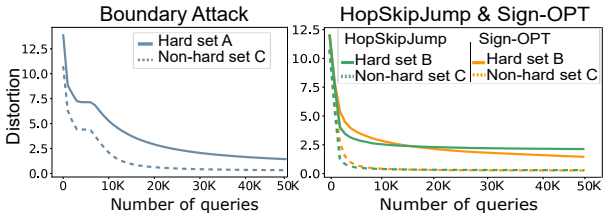

Fig. 8. A distortion comparison versus queries for each method using their own *hard* versus *non-hard* cases.

Evaluations on `CIFAR10`. From `CIFAR10` test set, we generate a *hard* set for Boundary Attack called *hard-set* A and another *hard* set for both Sign-OPT and HopSkipJump called *hard-set* B. The *hard-set* A and B are composed of 400 *hard* sample pairs of a source image and a starting image. A *hard* sample is selected when a distortion between a source image and its adversarial example found after 50,000 queries is larger than or equal to 0.9. For a fair comparison, each method is employed to craft an adversarial example for each source image initialized with a given starting image. In addition, we also construct a common *non-hard* set for all three attacks called *non-hard set* C to compare and highlight the significant difference between evaluation results on *hard* and *non-hard* sets as shown in Fig. 8. In particular, Fig. 8 illustrates that the average distortion versus queries on the common *non-hard set* C achieved by these methods is significantly lower than that obtained on theirs own *hard* set after 50,000 queries.

We evaluate our RamBoAttack on *hard-set* A & B. Fig. 9 shows that Boundary Attack, Sign-OPT and HopSkipJump do not efficiently find an adversarial example with low distortion; however, RamBoAttack can achieve better performance on the *hard-sets*. We defer detailed evaluations on *non-hard-sets* to Appendix C; as expected, RamBoAttack performs comparably-well on these sets. Histogram charts in Fig. 10 demonstrate that for each *hard-set*, our attacks are able to find lower distortion adversarial examples for most *hard* cases and the distortion distribution on both *hard-sets*: i) are shifted to smaller distortion regions; and ii) show significantly smaller spread or variance.

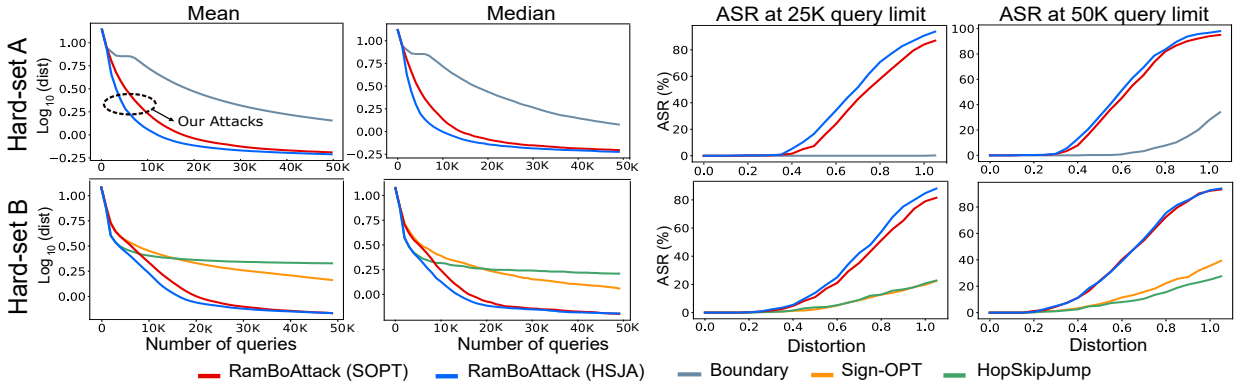

Fig. 9. Distortion (dist) on a $\log_{10}$ scale vs number of queries. The first row shows the results for our RamBoAttacks versus Boundary attack on *hard-set* A whilst the second row illustrates the results for our RamBoAttacks versus HopSkipJump and Sign-OPT on *hard-set* B. Our RamBoAttacks are ***more query efficient*** in *hard* cases. Hence our attack is demonstrably more robust and query efficient.

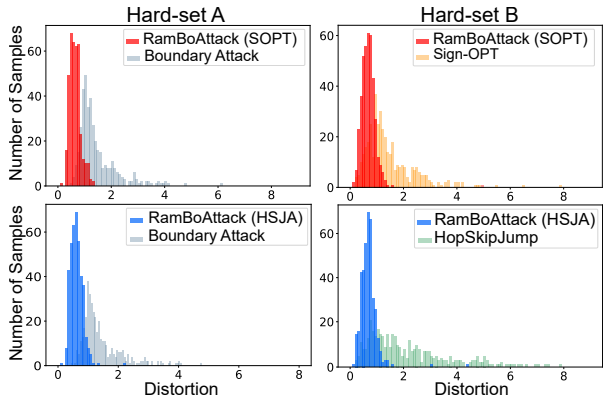

Fig. 10. On both *hard-set* A and B selected from `CIFAR10`, the distortion distribution yielded by our RamBoAttacks are shifted left and indicates an overall smaller distortion compared to other attacks.

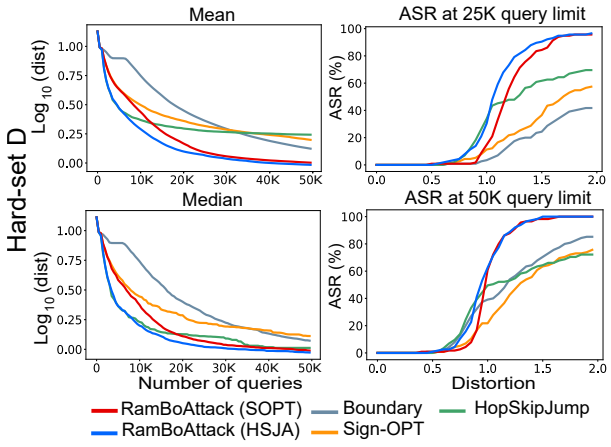

Fig. 11. Distortion in a $\log_{10}$ scale vs number of queries on *hard-set* D. Our RamBoAttack is ***more query efficient*** and achieves a higher ASR on this *hard-set*. Hence, our attack is demonstrably more robust and query efficient.

Although we observe RamBoAttack to result in fewer hard samples in comparison to other methods at various distortion thresholds, we construct a *hard* set for RamBoAttack called *hard-set* D based on the same criteria used to generate *hard-set* A and B to assess if the *hard-set* for RamBoAttack could somehow be easier for the other attack methods. The total number of samples for this set is 115 sample pairs because RamBoAttack has a much lower number of *hard* cases than their counterparts (namely BA, HopSkipJump and Sign-OPT) at a given distortion threshold as illustrated in Fig. 7. We summarize the results from our evaluations in Fig. 11. As expected, RamBoAttacks are more query efficient and are able to craft lower mean and median distortion adversarial examples as well as achieve higher attack success rates at both query budgets. In particular, at distortion levels above 1.0, in comparison to other attacks, RamBoAttacks obtain much higher attack success rates—notably, with significant margins at the lower query budget of 25K, since RamBoAttacks employ BlockDescent when the gradient estimation method is unable to make progress (potentially being stuck in a bad local minimum), to discover better solutions and craft lower distortion adversarial samples.

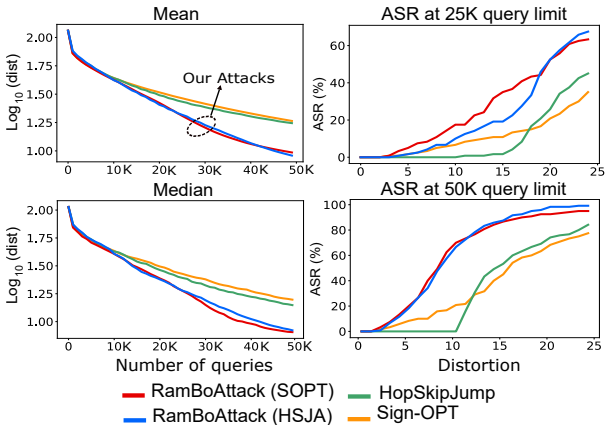

Fig. 12. Distortion (dist) in a $\log_{10}$ scale vs number of queries on *hard* `ImageNet` evaluation sets. The results on the *hard-set* show our RamBoAttacks are ***more query efficient***. Hence our attack is demonstrably more robust and query efficient.

Evaluation on `ImageNet`. To demonstrate the robustness of our attacks on a large scale model and dataset, we compose a *hard-set* with 120 *hard* sample pairs from `ImageNet`. A hard sample is selected when a distortion between a source image

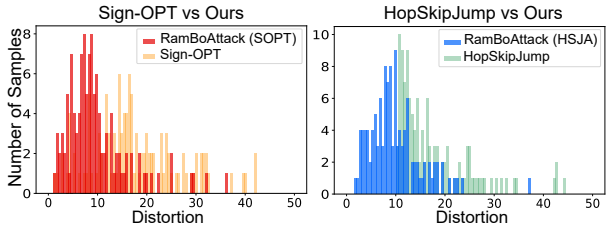

Fig. 13. On the *hard-set* selected from `ImageNet`, the distortion distributions yielded by our RamBoAttacks indicate an overall smaller distortion compared to other attacks. The distributions is shifted to the left and has significantly less variance compared to other attacks.

and its adversarial example found after 50,000 queries by Sign-OPT and HopSkipJump is larger than or equal to 15. Notably, we do not compose a *hard* set for Boundary Attack because it cannot yield low distortion adversarial examples efficiently on large scale datasets. Fig. 12 demonstrates that our RamBoAttacks outperform both Sign-OPT and HopSkipJump on the *hard-set*. We defer detailed evaluations on *non-hard-sets* to Appendix C; notably, RamBoAttacks achieve improved results on the more complex `ImageNet` dataset. The histograms in Fig. 13 show distortion distributions for our attacks shifted significantly to smaller distortion regions with smaller variance and fewer outliers compared to other attacks.

D. Impact of Various Starting Images

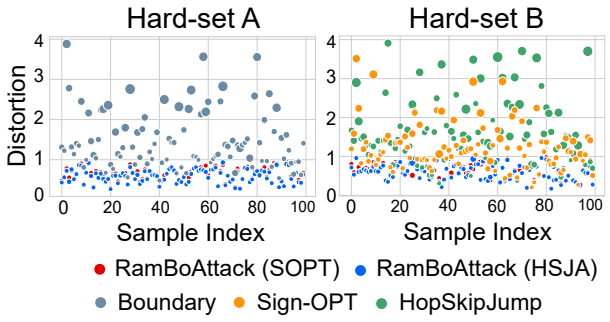

Fig. 14. An illustration of the sensitivity of different attacks to various starting images. Each method is evaluated on each subset and the charts show the average and variance of distortion for each subset achieved by different methods. y-axis denotes the average distortion while the size of each bubble denotes the distortion variation. Compared with Boundary, Sign-OPT and HopSkipJump attacks, our RamBoAttacks are ***much less sensitive to the choice of a starting image***.

In this experiment, we first compose *subset* A and B by selecting 100 random *hard* sample pairs from *hard-set* A and B, respectively (see Section IV-C for these sets). Our RamBoAttacks are compared with Boundary attack on subset A and with Sign-OPT and HopSkipJump, on subset B. In Section IV-C, each method needs to yield an adversarial example for a pair of a given source image and a given starting image. In contrast, in this experiment, the given starting image is replaced by 10 starting images randomly selected from the `CIFAR10` evaluation set and correctly classified by the model. All evaluations are executed with a 50K query budget.

In Fig. 14, the size of a bubble denotes the standard deviation while y-axis indicates average distortion. We can see that our RamBoAttacks *consistently* achieve smaller mean and standard deviation than Sign-OPT, HopSkipJump and Boundary Attack on subset A and B. A robust method should be less susceptible to the selection of a starting image and yield a low distortion adversarial example most chosen starting images. We can observe from Fig. 14 that our RamBoAttacks are more robust than Sign-OPT, HopSkipJump and Boundary attacks as a consequence of being less sensitive to the chosen starting images. For completeness, we also carry out this experiment on the *non-hard* subset C—please see Appendix E.

E. Attack Insights

Perturbation Regions. First, we develop a simple technique to transform a perturbation with size $C \times W \times H$ to a Perturbation Heat Map (PHM) with size $W \times H$ that is able to visualize perturbation magnitude of each pixel. This transformation is defined as:

$$PHM_{i,j} \leftarrow \mathrm{A}_{i,j} / \max(\mathrm{A}), \tag{4}$$

where $\mathrm{A}_{i,j} = \sum_{c=1}^{C} |(\boldsymbol{x} - \boldsymbol{x}_\mathrm{a})_{c,i,j}|$; $c \in [1, C]$, $i \in [1, W]$ and $j \in [1, H]$. Second, since Grad-CAM [26] is a popular visual explanation technique for visualizing salient features in an input image to understand a CNN model's decision, we use it to investigate the adversarial perturbations generated by our attack and the salient features in the target image largely responsible for a model's decision for the classification of an input to a target class.

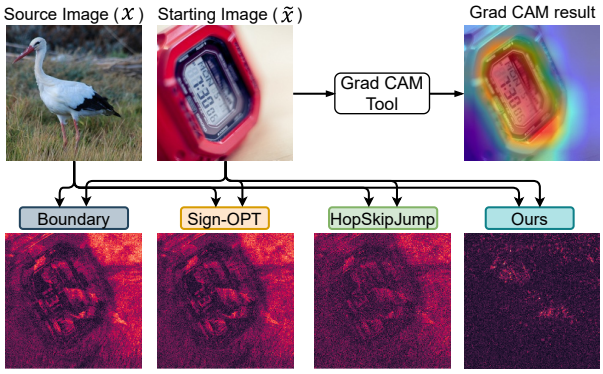

Fig. 15. Grad-CAM tool visualizes salient features of the starting image or target class: `digital watch`. Perturbation heat map (PHM) visualizes the normalized perturbation magnitude at each pixel. Comparing different perturbations crafted by different attacks highlights that the localized perturbations yielded by RamBoAttack concentrate on salient areas illustrated by GRAD-CAM and embeds these targeted perturbations in the source image to fool the classifier to predict the target class; even though, RamBoAttack does not exploit the knowledge of salient regions to generate perturbations—see additional examples in Appendix Fig. 29

In all methods, we observe the attacks to embed the target image in the source image in a deceptive manner. However, in *hard* cases, based on PHM and Grad-CAM outcomes, we observe a strong connection between adversarial perturbations found and salient regions in starting images as illustrated in

Fig. 15 for RamBoAttacks. It shows that our RamBoAttacks are able to discover and limit manipulations of pixels to salient regions responsible for determining the classification decision of an inputs to the target class to craft adversarial examples. These salient regions consist of the most discriminative local structures of a starting image against a source image. Because BlockDescent is able to manipulate local regions, RamBoAttacks are able to exploit only these discriminative regions and employ less adversarial perturbations than Sign-OPT and HopSkipJump to promote features of a starting image and suppress the feature of the source image. Therefore, it may shed light on why RamBoAttacks with the BlockDescent core component are able to tackle the so-called *hard* cases. Moreover, in *hard* cases, we observe that our RamBoAttacks are able to yield perturbations with more semantic structures when compared with Sign-OPT or HopSkipJump.

Visualization of `ImageNet` Hard versus Non-hard Cases. Fig. 16 illustrates adversarial examples in *non-hard* cases and *hard* cases yielded by Boundary Attack, Sign-OPT, HopSkipJump and our RamBoAttack (HSJA) after 50K and 100K queries, respectively. The second row of Fig. 16 shows each corresponding adversarial example and the third row illustrates PHM of each adversarial example. The last row shows the l_2 distortion between each adversarial example and the source image.

For the adversarial example of *non-hard* cases, all methods are able to craft low distortion adversarial examples except Boundary attack. These adversarial examples and their corresponding distortions are comparable. On the contrary, adversarial examples in *hard* cases yielded by Boundary Attack, Sign-OPT and HopSkipJump have *noticeably higher distortion* than the one crafted by our attack. We observe Boundary Attack, Sign-OPT and HopSkipJump to experience potential entrapment when searching for a low distortion adversarial example, even when the budget is increased to 100K queries.

Convergence. The problem considered in this paper is non-convex and non-differentiable. As such, providing a guaranteed global minimum is not possible. However, our insight is that the gradient estimation in blackbox attacks is unreliable particularly in the vicinity of the local minima. To remedy the problem, we propose RamBoAttack as a generic method to overcome this issue. We employ a gradient estimation method in the initial descent using any of the existing alternatives (before BlockDescent) and subsequently in the refinement stage (after BlockDescent). Hence, employing the gradient estimation in [14], for instance, would imply that the theoretical convergence analysis therein is still valid for our method.

F. Attacks Against Defended Models

In this section, we evaluate the robustness of various attacks against three different defense mechanisms including region-based classification, adversarial training and defensive distillation. We choose these defense methods due to their own strengths; to illustrate, region-based classifiers can pragmatically alleviate various adversarial attacks without sacrificing classification accuracy on benign inputs [8] whilst adversarial training [21], [29] is one of the most effective defense mechanisms against adversarial attacks [5] and defensive distillation [25] employ's a form of gradient masking.

For a *baseline*, we choose C&W attack [9], a state-of-the-art *white-box attack*. The adversarial training based models used in this experiment is trained with Projected Gradient Descent (PGD) adversarial training proposed in [21]. The experiment is conducted on the balance set withdrawn from `CIFAR10` described in Appendix C. We evaluate our RamBoAttack and current state-of-the-art decision-based attacks at different query budgets: 5K, 10K and 25K.

Based on the results, deferred to Appendix F, we observe that RamBoAttacks are *more robust* than Boudnary, Sign-OPT, HopSkipJump and even C&W (*white-box attack baseline*) when attacking a region-based classifier. In attacks against models using adversarial training and defensive distillation, RamBoAttacks are able to achieve comparable performance to Sign-OPT and HopSkipJump but outperform Boundary and C&W attack—*white-box* attack baseline.

V. Related Work

Transfer Approaches. Malicious adversaries are able to exploit transferability of adversarial example generated on an ensemble DNN to attack against a target neural network as shown by Liu et al. [20]. Papernot et al. [23] introduced a transfer attack by training a surrogate model with output queried from a target model. Even though this approach does not require prior knowledge and full access to a model, it must have access to a full or partial training dataset in order that they can train a surrogate model to synthesize adversarial examples. Moreover, for complex target models, the transfer approach has limited effectiveness [27].

Random Search Approaches. In decision-based setting, Brendel et al. [6] and Brunner et al. [7] proposed Boundary Attack (BA) and Biased Boundary Attack (Biased BA) respectively that require limited information and access to a target DNN model such as top-k predicted labels. Instead of searching on Gaussian distribution like BA, Biased BA exploits low frequency perturbations based on Perlin Noise and combines with regional masking as well as gradients from surrogate models. Even though both of them work surprisingly well, they do not gain query efficiency and require a large number of queries to explore a high-dimensional space. Another attack method introduced by Ilyas et al. [18] exploits discretized score based on the ranking of the adversarial label. However, since this method requires top k sorted label results from a deep learning model to estimate the discretized score, it cannot work in top 1 label scenario like BA or Bias BA.

Optimization Approaches. In score-based scenario, attackers can query a deep learning model to receive probability outputs or confident scores. Therefore, Chen et al. [12] can formulate an optimization problem to directly optimize an objective function based on these outputs. This method is considered as a derivative-free optimization method. Nevertheless, in the

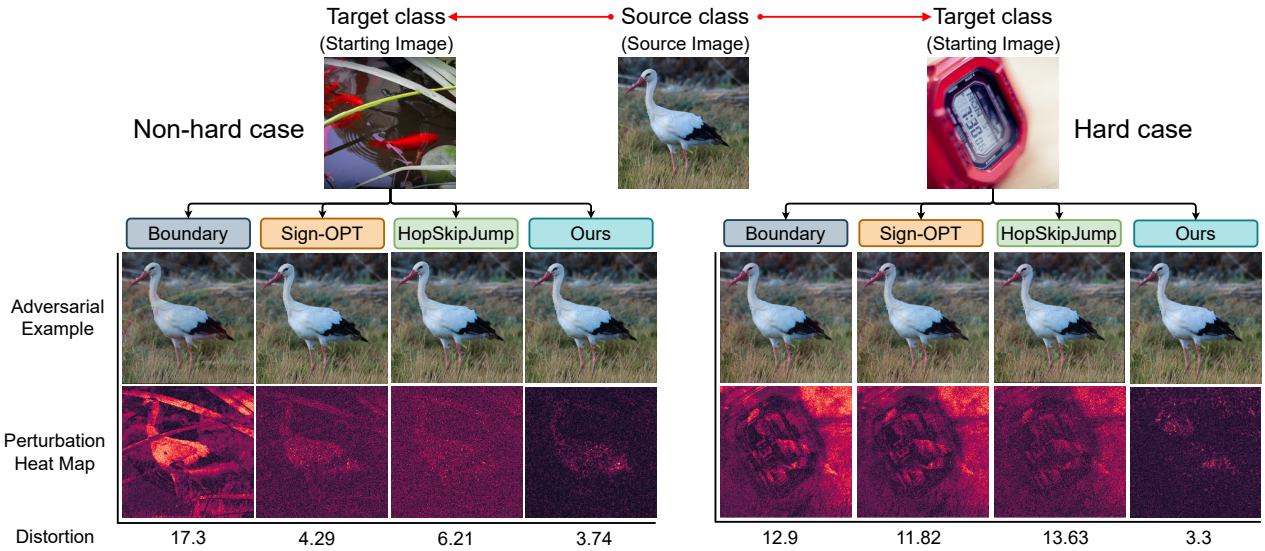

Fig. 16. An illustration of a *non-hard* case (`white stork` to `goldfish`) versus a *hard* case (`white stork` to `digital watch`) on `ImageNet`. Adversarial examples in *non-hard* cases and *hard* cases are yielded after 50K and 100K queries, respectively. Except for Boundary attack, adversarial examples crafted by different attacks in *non-hard* cases are somewhat different whilst in the *hard* case, our RamBoAttack is clearly able to craft an adversarial example with much smaller distortion than other attacks due to the ability of our BlockDescent formulation to *target effective localized perturbations*.

decision-based setting, adversaries have no access to confident scores or class probability to gain gradient information. Hence, the formulated optimization problem proposed by Chen et al. [12] cannot be applied. In Section II-C we discuss in detail optimization-based framework under decision-based setting and refer the reader to the section for further details.

VI. Conclusion

Overall, we propose a new attack method in a decision based setting; RamBoAttack. In contrast to modifying a whole image as in current attacks, we exploit localized perturbations to yield more effective and low distortion adversarial examples in the so-called *hard* cases. Our empirical results demonstrate that our attack outperforms current state-of-the-art attacks. Interestingly, while the main proposed component, BlockDescent, is able to significantly improve the performance and robustness of attacks in the so-called *hard* cases, it does not degrade performance in *non-hard* cases. As a result, validation results on small and large scale evaluation sets demonstrate that RamBoAttack is *more robust and query efficient* than current state-of-the-art attacks.

References

[1] "Amazon Machine Learning". [Online]. Available: https://aws.amazon.com/machine-learning/

[2] "Azure Cognitive Service". [Online]. Available: https://azure.microsoft.com/en-us/services/cognitive-services/

[3] "Google Cloud Vision". [Online]. Available: https://cloud.google.com/vision

[4] "IBM Watson Machine Learning". [Online]. Available: https://www.ibm.com/cloud/machine-learning

[5] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *International Conference on Machine Learning (ICML)*, 2018.

[6] W. Brendel, J. Rauber, and Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *International Conference on Learning Recognition (ICLR)*, 2018.

[7] T. Brunner, F. Diehl, M. Le, and A. Knoll, "Guessing smart: Biased sampling for efficient black-box adversarial attacks," *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[8] X. Cao and N. Z. Gong, "Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification," *Annual Computer Security Applications Conference (ACSAC)*, 2017.

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," *IEEE Symposium on Security and Privacy (S&P)*, 2017.

[10] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving," *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[11] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A Query-Efficient Decision-Based Attack," *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[12] P. Chen, H. Zhang, Y. Sharma, J. Yi, and C. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," *ACM Workshop on Artificial Intelligence and Security (AISec)*, pp. 15–26, 2017.

[13] M. Cheng, T. Le, P. Chen, H. Zhang, C. Hsieh, and J. Yi, "Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach," *International Conference on Learning Recognition (ICLR)*, 2019.

[14] M. Cheng, S. Singh, P. Chen, P.-Y. Chen, H. Yi, J. Zhang, and C.-J. Hsieh, "Sign-OPT: A Query-Efficient Hard-label Adversarial Attack," *International Conference on Learning Recognition (ICLR)*, 2020.

[15] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving Black-box Adversarial Attacks with a Transfer-based Prior," *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[16] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[17] K. He, X. Zhang, S. Ren, and J. Sun, " Deep residual learning for image recognition," *Computer Vision and Pattern Recognition (CVPR)*, p. 770–778, 2016.

[18] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," *International Conference on Machine Learning (ICML)*, 2018.

[19] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research). [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[20] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *International Conference on Learning Recognition (ICLR)*, 2017.

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *International Conference on Learning Recognition (ICLR)*, 2018.
[22] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," *Proceedings of the 18th ACM International Conference on Multimedia*, p. 1485–1488, 2010.
[23] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. Celik, and A. Swami, "Practical black-box attacks against machine learning," *ACM on Asia Conference on Computer and Communications Security (ASIA CCS)*, pp. 506–519, 2017a.
[24] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *IEEE European Symposium on Security and Privacy (EuroS&P*, pp. 372–387, 2016.
[25] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," *IEEE Symposium on Security and Privacy (S&P)*, 2016.
[26] R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," *The IEEE International Conference on Computer Vision (ICCV)*, 2017.
[27] F. Suya, J. Chi, D. Evans, and Y. Tian, "Hybrid Batch Attacks: Finding Black-box Adversarial Examples with Limited Queries," *USENIX Security Symposium*, 2020.
[28] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *International Conference on Learning Recognition (ICLR)*, 2014.
[29] F. Tramer, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *International Conference on Learning Recognition (ICLR)*, 2018.
[30] K. Xu, S. Liu, P. Zhao, P. Chen, H. Zhang, Q. Fan, D. Erdogmus, Y. Wang, and X. Lin, "Structured Adversarial Attack: Towards General Implementation and Better Interpretability," *International Conference on Learning Recognition (ICLR)*, 2019.

APPENDIX A
HYPER-PARAMETERS AND IMPACTS

Gradient Estimation: The main hyper-parameter n_t used in gradient estimation method is to control when the first component terminates and switches to BlockDescent. In practice, we keep track of query numbers executed and distortion between the source image and a crafted sample per iteration. This information is then used to determine distortion reduction rate Δ over T queries. On `CIFAR10`, if applying HopSkipJump or Sign-OPT to the first component, $T = 500$ *or* 400, respectively while on `ImageNet`, $T = 2000$ *or* 1000, respectively.

BlockDescent: The hyper-parameters used are $n = 1$, initial $\delta = P_i(|\boldsymbol{x} - \boldsymbol{x}_s|)$, $m = 1, \lambda = 1.2, \epsilon_r = 0.01, \epsilon_s = 0.01$ for GradEstimation, $\epsilon_s = 0.01$ for BlockDescent, $T = 500$ and $P_i = P_{100}$. For the *larger* dataset, `ImageNet`, the changes are: $m = 16, \lambda = 2, \epsilon_r = 0.1, \epsilon_s = 1$ for GradEstimation, $\epsilon_s = 0.1$ for BlockDescent, $T = 1000$ and $P_i = P_{50}$.

The impact of parameter λ: The key parameter that may influence BlockDescent is λ because it controls the step size (or perturbation magnitude δ) for each cycle (see line 28 in Algorithm 3). For example, λ is used to determine the step from $x^{(4)}$ to $x^{(5)}$ in Fig. 6. If λ is small, δ reduces slightly and thus remains relatively large after each cycle. Consequently BlockDescent takes large movements that are likely to yield large magnitude adversarial examples and/or miss the optimal solution. Alternatively, it may cross the decision boundary into an undesired class (source image class in a targeted attack).

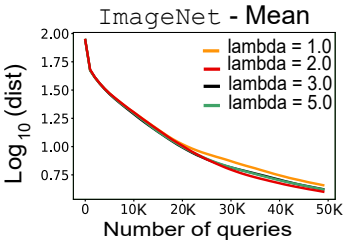

Fig. 17. A comparison between RamBoAttack with different values of λ on 100 source and target class sample pairs selected from `ImageNet`.

In contrast, if λ is large, BlockDescent takes finer steps to yield adversarial samples whilst moving towards the source image and likely stay in the desired class (target class in a targeted attack) Nevertheless, the empirical result with 100 pairs of source and target class images on `ImageNet` shown in Fig. 17 illustrates that the overall performance of RamBoAttack is not greatly affected by λ and at $\lambda = 2$, RamBoAttack achieves the best performance.

APPENDIX B
PROPOSED ROBUSTNESS EVALUATION PROTOCOL

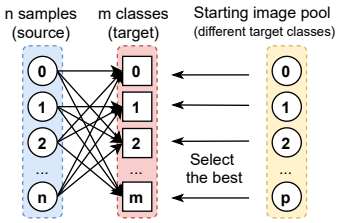

Fig. 18. The proposed evaluation protocol for assessing robustness under an exhaustive evaluation setting. In this mode, each sample from a dataset with size of n is evaluated to obtain an adversarial example for that sample capable of flipping its predicted label to m different target classes from that dataset. For each attack, a starting image is selected from a pool of p starting images.

An attack method is mounted to change the true prediction of the DNN from its ground truth label for a given source sample image to each of the different different target classes. For `CIFAR10` with ten classes, an attack method selects each of the 1000 test set samples for a given class as a source image and attempts to find an adversarial example for each of the other target classes (of which there are 9). Consequently, we evaluate 90,000 pairs of source and starting images. Since there is no effective method to choose a starting image from a target class, for a fair evaluation, we apply the same protocol used in [13], [14] to initialize an attack for each method. We execute each attack with a query budget of 50K queries. Then we identify *hard* cases of each attack method against the victim model (detailed in Section IV-A). This protocol can be generalized to other datasets by choosing n samples and m different target classes from that dataset where each target class has its own starting image as shown in Fig. 18.

APPENDIX C
PROPOSED VALIDATION PROTOCOL FOR BALANCED SETS AND RESULTS ON NON-HARD SETS

Evaluation protocol. The second research question highlights a need to evaluate the overall performance of various blackbox attacks under decision-based settings reliably. On `CIFAR10`, most previous works propose to choose a random evaluation set with randomly sampled images with label y and select a random target label $\tilde{y}$ [14] or set $\tilde{y} = (y + 1)$ mod

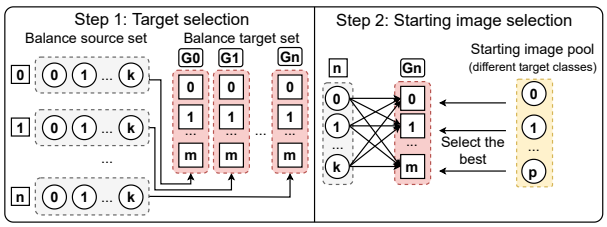

Fig. 19. The proposed evaluation protocol requires a balance dataset including n source classes and a balance target set comprising of n corresponding groups. On balance source set, all source classes have an equal number of samples (k) while all n corresponding groups have an equal number of target classes (m). These target classes are different within a group but can be repeated in other groups. From these groups G_n, a starting image is selected from a pool of p starting images.

10 [6], [7], [13]. Nonetheless, these selection schemes may lead to an imbalanced dataset that is insufficient to evaluate the effectiveness of the attack since it may lack the so called *hard* cases that occur more frequently with specific pairs of classes. As a result, it may lead to a bias in evaluation results and fail to highlight potential weaknesses of an attack. Consequently, were were motivated to propose a more robust and reliable evaluation protocol and illustrate it in Fig. 19.

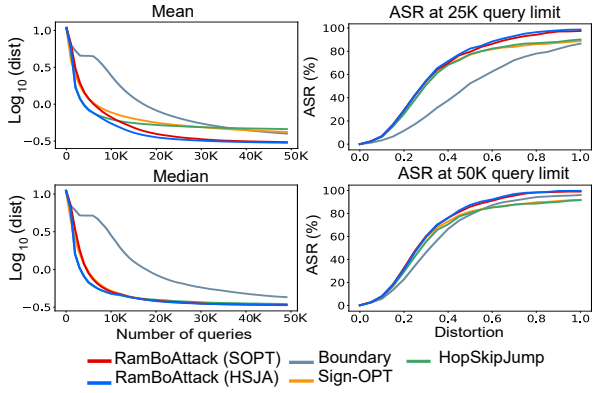

Fig. 20. A comparison between three current state-of-the-art attacks and RamBoAttacks on a balance set selected from `CIFAR10`.

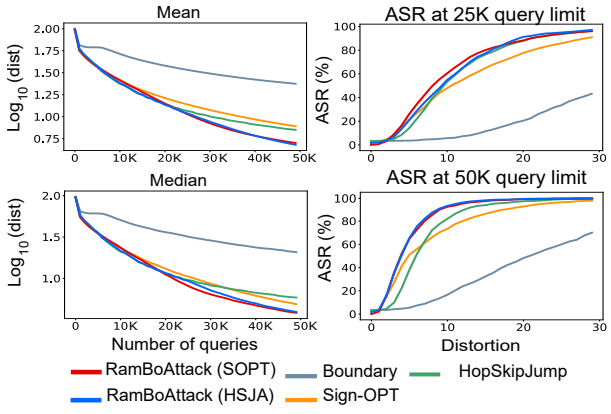

Fig. 21. A comparison between three current state-of-the-art attacks and RamBoAttacks on a large scale balance set selected from `ImageNet`.

TABLE I
SUMMARY COMPARISON AMONG ATTACKS WITH RAMBOATTACK ON SMALL AND LARGE SCALE BALANCE DATASETS.

Query budget	Methods	CIFAR10 Mean	CIFAR10 Std	CIFAR10 Median	CIFAR10 ASR(ϵ=0.3)	ImageNet Mean	ImageNet Std	ImageNet Median	ImageNet ASR
25K	Boundary	0.674	0.654	0.499	22.6%	31.80	18.43	32.88	5.5%
	HopSkipJump	0.507	0.748	0.296	50.8%	11.91	8.39	10.87	51.4%
	Sign-OPT	0.526	0.754	0.286	53.6%	14.21	11.52	9.81	46.3%
	RamBo. (HSJA)	**0.336**	**0.218**	0.283	54.0%	11.33	**8.0**	**8.62**	53.1%
	RamBo. (SOPT)	0.363	0.359	**0.282**	**54.1**%	**11.25**	9.47	9.62	**57.5**%
50K	Boundary	0.399	0.404	0.319	45.2%	23.73	15.65	20.71	16.6%
	HopSkipJump	0.460	0.683	0.273	55.3%	7.09	5.11	4.87	82.0%
	Sign-OPT	0.420	0.562	0.267	59.1%	7.79	7.84	5.87	73.3%
	RamBo. (HSJA)	**0.300**	**0.178**	**0.260**	59.9%	**4.80**	**3.70**	3.92	**93.1**%
	Rambo. (SOPT)	0.306	0.193	0.261	**60.11**%	5.02	4.57	**3.84**	92.3%

On balance sets: A balance set comprises of a balanced source set and a balanced target set. Both sets are composed of N different source classes and N corresponding groups. Each group is composed of m different target classes and all source and target classes are randomly chosen from all classes of a test set. In addition, all target classes are different within a group but can be repeated in other groups. Each source class has n samples selected randomly from a test set. Adversaries may have one or several images from each target class and select one to initialize an attack. Each attack method aims to craft an adversarial example for every selected sample from each source class and flip its true prediction towards every target class given in the corresponding group of balanced target set. The total number of evaluation pairs is $N \times n \times m$. For instance, every sample of source class i (img: $i_1, i_2, \cdots, i_n$) is flipped towards each target class (class: $i_1, i_2, \cdots, i_m$) in the corresponding group i (see Fig. 19).

Balanced Set with `CIFAR10`. It is simple to carry out a comprehensive evaluation over all classes, so we choose N=10, n=10 and m=9. In addition, to demonstrate the query efficiency and effectiveness of each attack, we employ a query budget of 25K and 50K across all experiments. RamBoAttack obtain slightly better median and mean distortion than HopSkipJump and Sign-OPT at 25K and 50K, as shown in Table I. On the standard deviation metric used to measure distortion variance across an evaluation set, our RamBoAttack outperform Boundary, HopSkipJump and Sign-OPT at query limit of 25K and 50K. In order words, our attack performs robustly across the evaluation set.

Balanced Set with `ImageNet`. `ImageNet` has 1000 distinct classes, hence carrying out a comprehensive evaluation like on `CIFAR10` requires huge computing resources and time. Therefore, we choose N=200, n=1 m=5 and limit the query budget to 25K and 50K. The average distortion (on a $\log_{10}$ scale) against the queries and attack success rate (ASR) at 25K and 50K query budgets achieved by RamBoAttack is better than Boundary, Sign-OPT and HopSkipJump attacks as seen in shown in Fig. 21. As shown in Table I, on average distortion metric, RamBoAttacks obtain better result and achieve significantly smaller standard deviation of distortion overall.

On *non-hard* sets: In this section, we evaluate the performance of SignOPT, HopSkipJump and our RamBoAttacks on both `CIFAR10` and `ImageNet` *non-hard* set. The common

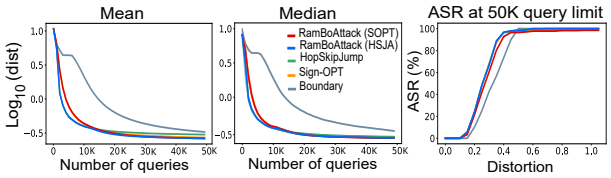

Fig. 22. A comparison between three current state-of-the-art attacks and RamBoAttacks on a *non-hard* set C selected from `CIFAR10`. In *non-hard* cases, we perform comparably.

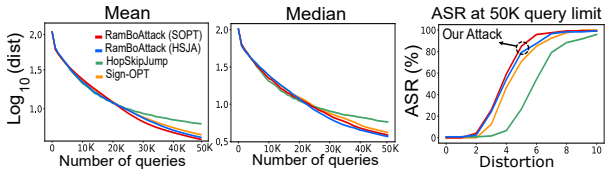

Fig. 23. A comparison between three current state-of-the-art attacks and RamBoAttacks on a *non-hard* set selected from `ImageNet`. In *non-hard* cases, RamBoAttacks improve attack performance by yielding more effective adversarial examples notable in ASR results.

non-hard set C drawn from `CIFAR10` for all methods is composed of 400 *non-hard* sample pairs. They are selected such that a distortion between a source image and its adversarial example found after 50K is smaller or equal 0.6. Likewise, a *non-hard set* from `ImageNet` is composed of 120 *non-hard* sample pairs and the distortion threshold to select these is is 7. Fig. 22 and 23 show that our attack has comparable performance to SignOPT and HopSkipJump on `CIFAR10` *non-hard* subsets whilst demonstrating improved attack performance by yielding more effective adversarial examples, especially with a 50K query budget, as seen in the higher attack success rates obtained by RamBoAttacks.

Appendix D
Untageted Attack Validation

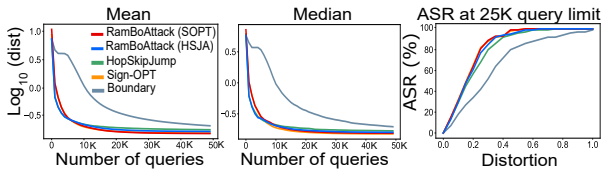

Fig. 24. Comparing between three state-of-the-art attacks and RamBoAttacks on the balance set selected from `CIFAR10` under an untargeted setting.

Here, we evaluate our RamBoAttack and other state-of-the-art attacks on two different balanced sets from `CIFAR10` and `ImageNet` as described in Appendix C under an untargeted scenario, for completeness. First, on the balance set from `CIFAR10`, our attacks can achieve comparable performance with Sign-OPT and HopSkipJump and obtain approximately 97% success rate at a distortion of 0.5 on a 25K query budget (see Fig. 24); notably, our attack method outperforms Boundary attack. In contrast, on the balance set selected from `ImageNet`, we observe that our methods can achieve comparable performance with Sign-OPT but outperform HopSkipJump and Bourndary attacks as shown in Fig. 25.

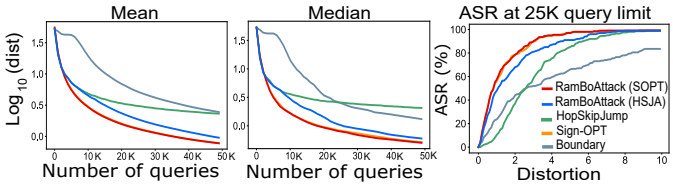

Fig. 25. Comparing between three current state-of-the-art attacks and RamBoAttacks on the balance set from `ImageNet` under untargeted setting.

Appendix E
Impact of Starting Images

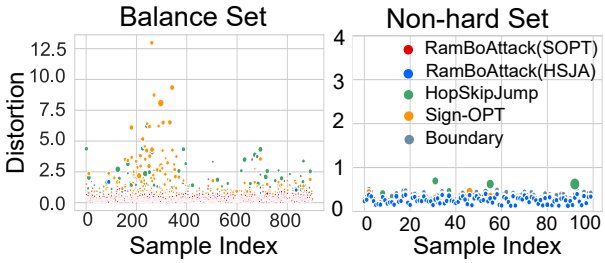

Fig. 26. An illustration of the sensitivity of different attacks to various chosen starting images. Size of each circle denotes standard deviation and y-axis indicates the mean distortion. The results are from the `CIFAR10` balance set and a *non-hard* subset from *non-hard* set C. Compared with Boundary, Sign-OPT and HopSkipJump attacks, our RamBoAttacks are ***much less sensitive to the choice of starting image*** in general. On *non-hard* cases, all of attacks can achieve comparable results. Hence our attack is demonstrably more robust.

In this section, we first compose a *non-hard* subset with 100 random *non-hard* sample pairs selected from *non-hard* set C. We also compose a balance subset from the balance set described in Appendix C. We then evaluate our RamBoAttack, Sign-OPT, HopSkipJump and Boundary attack on these subsets. To conduct this experiment, for every source image, a target class image is selected from 10 different randomly selected starting images and these attacks are executed with a query budget of 50K. We calculate the mean and standard deviation of distortion for each sample to measure the robustness of each attack to yield adversarial examples for each source image and 10 target class pairs.

In Fig. 26, size of each bubble denotes the standard deviation while the y-axis indicates mean distortion value. We can see that, on the *non-hard* subset, the RamBoAttacks are able to achieve comparable result to all of the state-of-the-art methods. On the balance subset, our RamBoAttacks can achieve significantly less variance (smaller bubbles) at lower distortions while most results achieved by Sign-OPT, HopSkipJump and Boundary indicate larger variance (larger bubbles) and higher distortions. Consequently, our RamBoAttacks are more robust than Sign-OPT and HopSkipJump and less sensitive to the chosen starting image.

Appendix F
Attacks Against Defended Models

In this section, we illustrate the results that we briefly mention in Section IV-F. Fig. 27 shows that the average

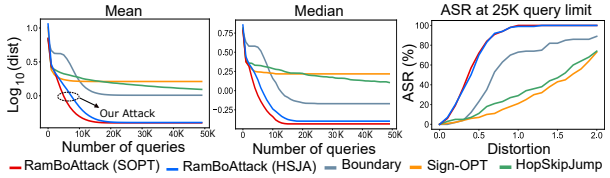

Fig. 27. Performance comparisons between different state-of-the-art attacks and RamBoAttacks against a region-based classifier on `CIFAR10`. RamBoAttacks outperforms other blackbox attacks and is able to craft *significantly more effective* adversarial examples of lower distortion against the defense method as seen by the higher ASR results against the defended models from RamBoAttacks across all of the evaluations.

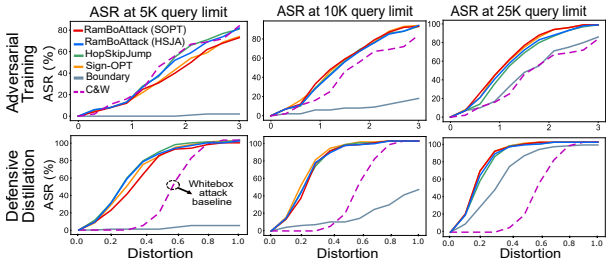

Fig. 28. ASR comparison between white-box (*employed as a baseline*) and current decision-based attacks versus our RamBoAttack against Adversarial Training model and Defensive Distillation on `CIFAR10` (using the balanced set). Interestingly, RamBoAttacks are more effective than the white-box attack method baseline, and are slightly more robust under different query settings when compared to other decision-based blackbox attacks.

and median distortion (on a $\log_{10}$ scale) achieved by RamBoAttacks are significantly lower than BA, Sign-OPT and HopSkipJump. In addition, our attack outperform others in terms of attack success rate (ASR) at 25K—i.e. achieves *higher* ASR on defended models under different query budgets and distortion thresholds. Based on these results, *we observe our attack is more robust than exiting attacks when mounting an attack against region-based classifiers*.

The reason for this is that existing attack methods need to follow the decision boundary where region-based classifiers are capable of correcting its prediction by uniformly generating a large amount of data points at random and returning the most frequent predicted label. This capability of region-based classifiers prevents binary search in Sign-OPT and HopSkipJump from specifying the boundary exactly and results in noisy and coarse boundary estimations that cause all attack methods aiming to walk along the boundary to fail to estimate a useful gradient direction. Nevertheless, our RamBoAttacks are able to break this defense mechanism because the core component, BlockDescent, is a derivative-free optimization method that does not need to determine the boundary and estimate a gradient to determine a direction to descend.

A. Results

Fig. 28 shows the attack success rate (ASR) at different distortion levels and query limits for various attack methods against an adversarially trained model and defensive distillation model. Particularly, for adversarial training, our RamBoAttacks can achieve comparable performance with Sign-OPT and HopSkipJump while outperforming Boundary attack within the query limits of 5K, 10K or 25K. In addition, we compare the performance of our attack at different query budgets with the whitebox attack—C&W—*used as a baseline for comparison*. Notably, we do not execute C&W attack at different query setting because it is a whitebox method and use the best result produced by this attack.

We observe that our attacks are able to obtain a comparable performance with the C&W attack at the 5K query budget. When the query limit is up to 10K and higher, our RamBoAttacks outperform the whitebox C&W baseline attack method. Nevertheless, Adversarial Training is still effective at reducing the ASR achieved by our method, even with a 25K query budget. Success falls from around 99% (see Fig. 24) to approximately 43% (see Fig. 28) at a distortion of 1.0 (l_2 norm). Similarly, at a distortion of 0.3, the ASR decreases from about 60% (see Fig. 24) to approximately 10% (see Fig. 28). However, what we can observe is that as the distortion increases, the attack is more effective. This is expected because the attack budget of the adversary is increased beyond the budget used for building the adversarially trained model.

Likewise, for defensive distillation, our RamBoAttacks can achieve comparable performance with Sign-OPT and HoSkipJump whilst outperforming Boundary attack and C&W whitebox baseline attack at different query budgets. These results confirm the results and findings presented in [11].

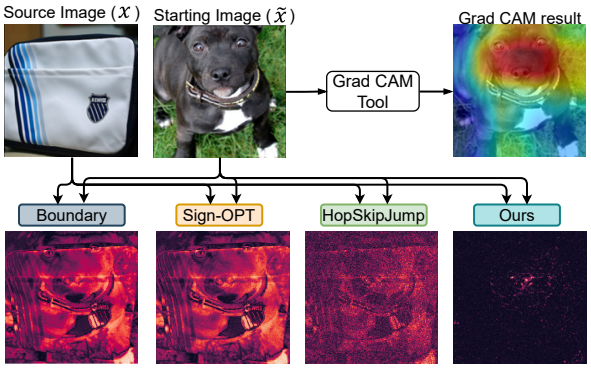

Fig. 29. Grad-CAM tool visualizes salient area of the starting image `Staffordshire bull terrier`. Perturbation heat map (PHM) visualizes the normalized perturbation magnitude at each pixel. It shows that the perturbation yielded by RamBoAttack is able to concentrate on salient areas illustrated by GRAD-CAM even though RamBoAttack does not exploit the knowledge of salient regions to perturb.

B. C&W Attack Configuration and Results Collection

For clarity, we firstly describe the configuration used for the C&W attack. We adopt the PyTorch implementation of the C&W method used in [13], [14]. In their implementation, they use a learning rate of 0.1 and 1000 iterations for all evaluations (see published code). To search for an adversarial example for an image, the method performs a binary search step to find a relevant constant c within a range from 0.01 to 1000 *until a successful attack is achieved*. With this configuration,

the C&W attack is run once to always yield an adversarial example for every instance. We record the distortion of the adversarial example found.

C&W Results Collection. To construct ASR vs. distortion results, at different distortion thresholds: i) we compute the number of source images in the evaluation set meeting a given distortion threshold (along the x-axis); ii) then divide this by the total number of images in the evaluation set to compute the ASR at each distortion value.

Blackbox Attack Results Collection. For the blackbox attacks, we perform a blackbox attack for each evaluation-set source image, using the set query budgets: 5K, 10K, and 25K. We record the distortion achieved by each source image with a set query budget. To construct ASR vs. distortion, at different distortion thresholds with a given query budget: i) we compute the number of source images in the evaluation set meeting a given distortion threshold (along the x-axis); and ii) then divide this by the total number of images in the evaluation set to compute the ASR at each distortion value.

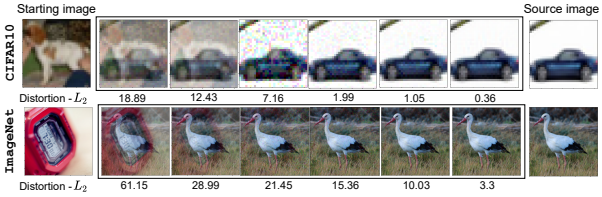

Fig. 30. Visualization of different distortions produced by RamBoAttack. The second example is from `ImageNet` with a starting image of a `digital watch` gradually perturbed until it is similar to the source image `white stork`—the final adversarial example crafted.

APPENDIX G
PERTURBATION REGIONS AND ATTACK INSIGHTS

In this section, we provide additional results on the connection between the adversarial perturbations yielded by RamBoAttack and salient regions visualized by the Grad-CAM tool. Effectively, all of the attack methods embedded the target features within the source image where the changes are effectively unnoticeable. However, Fig. 29 illustrates that a high density of adversarial perturbations yielded by our attack concentrates on a region that is matched to the salient features visualized by the Grad-CAM tool. This is possible because our attack methods employs localized changes to search for adversarial examples and is able to effectively find perturbations targeting salient features of the target class to apply to the input source class image to fool the classifier to classify the source image as the target class.

Further, to help visualize different level of l_2 distortions, we include Fig. 30. We illustrate two examples where we showcase the sample adversarial examples crafted by RamBoAttack during the progression of the attack.

APPENDIX H
ATTACK SUCCESS RATES VS QUERY BUDGETS

In this section, we show results at three different perturbation budgets ϵ = 0.4 and 0.6 for *hard-sets* A and B from

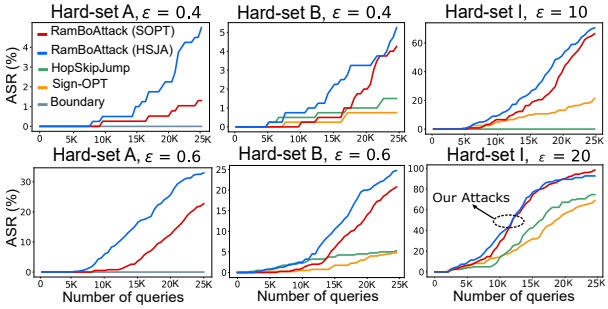

Fig. 31. The first and second columns illustrates ASR vs. queries for our RamBoAttacks with respect to Boundary attack on *hard-set* A and with respect to HopSkipJump and Sign-OPT on *hard-set* B. For a given query budget, as expected, our RamBoAttacks yield similar ASR to Sign-OPT and HSJA with very low query budgets and significantly higher ASR with budgets above 10K queries, where gradient estimation methods do not appear to improve the adversarial example found with increasing numbers of queries. Similarly, the third column illustrates ASR vs. queries for our RamBoAttacks with respect to HopSkipJump and Sign-OPT on the *hard-set* I. Our RamBoAttacks are more query efficient, more robust and are able to yield significantly higher ASR under low distortion settings.

`CIFAR10` and ϵ = 10 and 20 for the *hard-set* I selected from `ImageNet`. The results demonstrate that our attack is significantly more robust than other attacks within 4-11K query budgets. From 11K, RamBoAttacks outperforms others. The reason is that, around this region, the gradient estimation method switches to BlockDescent, resulting in much higher attack success rates compared to the baselines. Notably, on the high-resolution benchmark task `ImageNet`, RamBoAttacks achieve *significantly* better results compared to the baselines.

APPENDIX I
ROBUSTNESS OF RAMBOATTACK

Fig. 32 provides further detailed results on *hard* cases encountered by different attack methods at distortion threshold of 0.8, 0.9 and 1.0. Compared to Boundary, Sign-OPT and HopSkipJump attacks, our RamBoAttacks achieve much lower number of *hard* cases at all distortion thresholds.

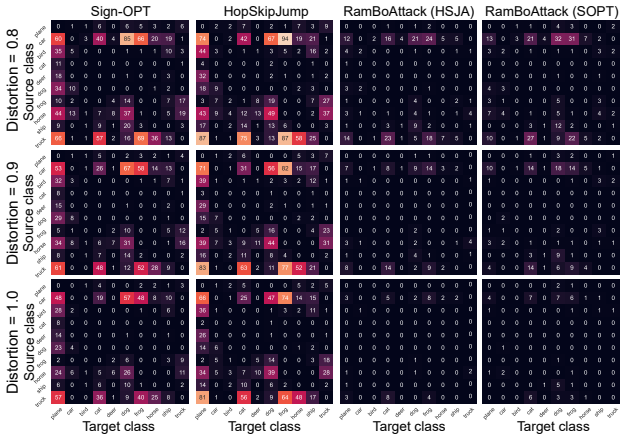

Fig. 32. The number of *hard* cases in `CIFAR10` obtained from different attack methods categorized by pairs of source and target classes (at distortion threshold = 0.8, 0.9 and 1.0). RamBoAttacks are seen to nearly overcome all of the *hard* cases encountered by other decision-based blackbox attack methods; thus, demonstrating the robustness of our proposed attack.