ADBA:Approximation Decision Boundary Approach for Black-Box Adversarial Attacks

Feiyang Wang[1,2], **Xingquan Zuo**[1,2*], **Hai Huang**[1,2], **Gang Chen**[3]

[1]School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China
[2]Key Laboratory of Trustworthy Distributed Computing and Services, Ministry of Education, Beijing, China
[3]School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand
{flyingw, zuoxq, hhuang}@bupt.edu.cn, aaron.chen@vuw.ac.nz

Abstract

Many machine learning models are susceptible to adversarial attacks, with decision-based black-box attacks representing the most critical threat in real-world applications. These attacks are extremely stealthy, generating adversarial examples using hard labels obtained from the target machine learning model. This is typically realized by optimizing *perturbation directions*, guided by decision boundaries identified through query-intensive exact search, significantly limiting the attack success rate. This paper introduces a novel approach using the Approximation Decision Boundary (ADB) to efficiently and accurately compare perturbation directions without precisely determining decision boundaries. The effectiveness of our ADB approach (ADBA) hinges on promptly identifying suitable ADB, ensuring reliable differentiation of all perturbation directions. For this purpose, we analyze the probability distribution of decision boundaries, confirming that using the distribution's median value as ADB can effectively distinguish different perturbation directions, giving rise to the development of the ADBA-md algorithm. ADBA-md only requires four queries on average to differentiate any pair of perturbation directions, which is highly query-efficient. Extensive experiments on six well-known image classifiers clearly demonstrate the superiority of ADBA and ADBA-md over multiple state-of-the-art black-box attacks.

Code — https://github.com/BUPTAIOC/ADBA

Introduction

Background and motivation. It is widely acknowledged that machine learning methods such as *Deep Neural Networks* (DNNs) are vulnerable to adversarial attacks (Xie et al. 2017). Particularly, for image classification tasks, tiny additive perturbations in the input images can significantly affect the classification accuracy of a pre-trained model (Xie et al. 2019). The impact of these intentionally designed perturbations in real-world scenarios (Eykholt et al. 2018; Lin et al. 2023) has heightened security worries for critical applications of deep neural networks in many domains (Eykholt et al. 2018; Li et al. 2023a,b; Li and Deng 2020), which is detailed in Appendix A.

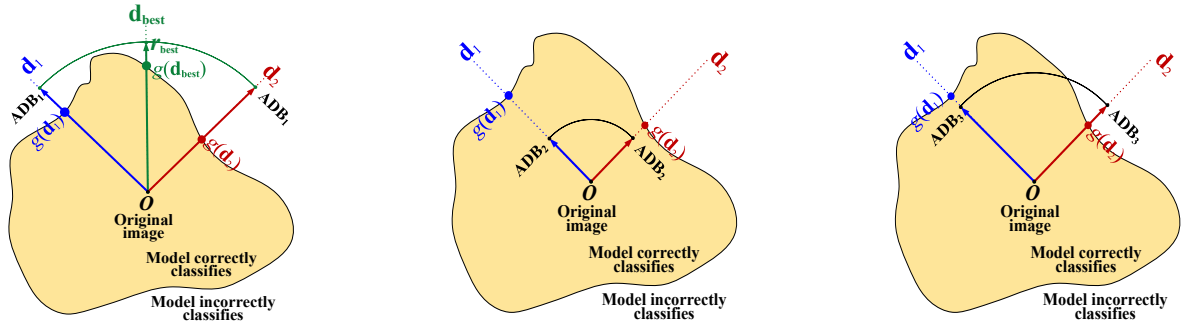
*Corresponding author.

Adversarial attacks can be categorized into *white-box* attacks and *black-box* attacks (Long et al. 2022). White-box attacks (Wang and He 2021) require the attacker to have comprehensive knowledge of the target machine learning model, rendering them impractical in many real-world scenarios. In comparison, black-box attacks (Bai, Wang, and Zeng 2023; Brendel, Rauber, and Bethge 2018; Li et al. 2021) are more realistic since they do not required detailed knowledge of the target model.

Black-box attacks can be divided into *transfer-based attacks*, *score-based attacks* (or *soft-label attacks, gray-box attacks*), and *decision-based attacks* (*hard-label attacks*) (Li et al. 2021). More detailed discussion of these attacks can be found in Appendix B. Among them, decision-based attacks (Li et al. 2021; Shi et al. 2022; Chen, Jordan, and Wainwright 2020; Cheng et al. 2020) are extremely stealthy since they rely solely on the hard label from the target model to create adversarial examples.

This paper studies the decision-based attacks due to their general applicability and effectiveness in real-world adversarial situations. These attacks aim to deceive the target model while adhering to two constraints (Ilyas et al. 2018): 1) they must generate adversarial examples with as few queries as possible (i.e., **query-efficient**) and cannot exceed a predetermined number of queries (i.e., **query budget**), and 2) the strength of the adversarial perturbations must remain within a predefined threshold ϵ. Violating those constrains results in the attack being easily detected or deemed unsuccessful. These constraints bring huge challenges to attackers (Ilyas et al. 2018). Specifically, lacking detailed knowledge of the target model and its output scores poses tremendous difficulty for attackers to determine the *decision boundary* (i.e., the minimum perturbation strength required to deceive the model) with respect to any perturbation direction (Cheng et al. 2020). Thus, decision-based attacks often require a large number of queries to identify the decision boundary and optimize the perturbation direction, increasing the likelihood of being detected and hurting the attack success rate. Therefore, enhancing attack efficiency and minimizing the number of queries are essential for decision-based attacks.

Proposed method. Existing decision-based attacks can be divided into *random search attacks* (Brendel, Rauber, and Bethge 2018; Li et al. 2021; Cheng et al. 2020; Brunner et al. 2019; Chen and Gu 2020; Cheng et al. 2018)(reviewed in



(a) Too large ADB fails to compare $g(\boldsymbol{d_1}), g(\boldsymbol{d_2})$

(b) Too small ADB fails to compare $g(\boldsymbol{d_1}), g(\boldsymbol{d_2})$

(c) ADB succeed in comparing $g(\boldsymbol{d_1}), g(\boldsymbol{d_2})$

Figure 1: Compare directions $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ using ADB. In (a), comparison of $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ fails because ADB_1 is too large ($g(\boldsymbol{d_1}) \leq \text{ADB}_1$ and $g(\boldsymbol{d_2}) \leq \text{ADB}_1$). In (b), comparison of $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ fails because ADB_2 is too small ($g(\boldsymbol{d_1}) > \text{ADB}_2$ and $g(\boldsymbol{d_2}) > \text{ADB}_2$). In (c), $\boldsymbol{d_2}$ is superior to $\boldsymbol{d_1}$ and $g(\boldsymbol{d_2}) \leq \text{ADB}_3 < g(\boldsymbol{d_1})$ because the attack in direction $\boldsymbol{d_1}$ fails but the attack in direction $\boldsymbol{d_1}$ succeeds. ADB_3 is hence the ADB of $\boldsymbol{d_2}$.

detail in Related Work), *gradient estimation attacks* (Chen, Jordan, and Wainwright 2020; Li et al. 2020; Liu, Moosavi-Dezfooli, and Frossard 2019; Rahmati et al. 2020), and *geometric modeling attacks* (Wang et al. 2022; Reza et al. 2023; Maho, Furon, and Le Merrer 2021)(reviewed in detail in Appendix B). This paper focuses on random search attacks, aiming to find the optimal perturbation direction with the smallest decision boundary. For this purpose, query-intensive exact search techniques such as binary search are typically utilized to identify the decision boundaries of different perturbation directions (Cheng et al. 2020; Chen and Gu 2020). For two candidate perturbation directions $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$. Binary search is used to calculate their decision boundaries $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$, and determine which direction has the smaller decision boundary (Chen and Gu 2020). However, binary search demands for a large number of queries, resulting in poor query efficiency.

In this study, we show that different perturbation directions can be compared without knowing their precise decision boundaries, avoiding costly exact search methods, as demonstrated in Figure 1(c). Suppose that an approximation decision boundary (ADB) enables direction $\boldsymbol{d_2}$ to deceive the target model but direction $\boldsymbol{d_1}$ fails to do the same, we can conclude that $g(\boldsymbol{d_2}) \leq \text{ADB} < g(\boldsymbol{d_1})$ and $\boldsymbol{d_2}$ is deemed superior for the black-box attack. Here, $g(\boldsymbol{d_1})$and $g(\boldsymbol{d_2})$ refer to the actual decision boundaries of $\boldsymbol{d_1}$and $\boldsymbol{d_2}$. On the contrary, $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$ cannot be clearly differentiated upon using ADB that is either too large or too small, as shown in Figure 1(a) and Figure 1(b). Hence, a key issue is to quickly find an appropriate ADB with few queries. To tackle this issue, we analyze the probability distribution of decision boundaries and discover that it is highly likely for any pair of perturbation directions to be successfully differentiated upon using this distribution's median value as ADB. In fact, only four queries on average are required for this purpose. Based on this idea, we propose innovative *Approximation Decision Boundary Approach* (ADBA) and *Median-Search Based ADBA* (ADBA-md) for random search attacks.

Comprehensive experiments. We perform comprehensive experiments for decision-based black-box attacks using a variety of image classification models. We select 6 well-known deep models that span a wide range of architectures. We compare our methods with four state-of-the-art decision-based adversarial attack approaches. The results show that our methods can generate adversarial examples with a high attack success rate (i.e., fooling rate) while using a small number of queries.

This paper makes three main contributions. (1) We propose a novel Approximation Decision Boundary Approach (ADBA) for query-efficient decision-based attacks. (2) We analyze the distribution of decision boundaries and discover that using the distribution's median value as ADB can differentiate any pair of perturbation directions with high query efficiency, giving rise to the development of ADBA-md. (3) We conduct comprehensive experiments, demonstrating that ADBA and ADBA-md can significantly outperform four leading decision-based attack approaches across six modern deep models on the ImageNet dataset.

Related Work

Random search attacks adopt a random search framework to generate candidate perturbation directions and adjust them according to the sizes of decision boundaries (Brendel, Rauber, and Bethge 2018; Brunner et al. 2019; Cheng et al. 2018, 2020; Chen and Gu 2020). These boundaries are precisely identified through methods such as binary search, which requires a large number of queries. For example, Boundary Attack (Brendel, Rauber, and Bethge 2018), Biased Boundary Attack (Brunner et al. 2019), and AHA (Li et al. 2021) progressively alter the current perturbation direction through a random walk along the decision boundary, which is query-intensive. To reduce the number of queries in Boundary Attack, Biased Boundary Attack (Brunner et al. 2019) proposes a biased sampling framework to accelerate the search for perturbation directions. Meanwhile, AHA in (Li et al. 2021)

gathers information from all historical queries as a prior for current sampling decisions to improve the efficiency of random walk. Additionally, OPT and Sign-OPT developed respectively in (Cheng et al. 2018) and (Cheng et al. 2020) transform the hard-label black-box attack to a continuous optimization problem, solvable through zeroth-order optimization. However, they utilize binary search to identify the decision boundary of each perturbation direction, demanding for a large number of queries. RayS in (Chen and Gu 2020) is a state-of-the-art decision-based attack based on the l_∞ norm. It transforms the task of optimizing the perturbation direction from the continuous domain to the discrete domain. It generates perturbation directions by progressively dividing and inverting direction vectors, with each division resulting in smaller blocks as the optimization advances, noticeably enhancing the efficiency in finding good directions. However, RayS optimizes perturbation directions based on precise estimation of the decision boundary through a query-intensive binary search process. More discussion of related works, including gradient estimation attacks and geometric modeling attacks, can be found in Appendix B.

Method

Preliminaries

Let $F\colon \mathbb{X}^N \to \{1,\dots,K\}$ denote a target image classification model that assigns images to one of K distinct classes. The model's input is a normalized RGB image $\boldsymbol{x} \in \mathbb{X}^N$, where N = Width $\times$ Height $\times$ Channels represents the dimensionality of the image, encompassing all pixels across all channels. In this context, $\mathbb{X}^N$ is the input space of all images to be classified. The channel value of each pixel of any image $\boldsymbol{x} \in \mathbb{X}^N$ ranges between 0 and 1. Meanwhile, $y(\boldsymbol{x}) \in \{1,\dots,K\}$ denotes the true label of image $\boldsymbol{x}$, and $F(\boldsymbol{x})$ is the label produced by the classification model F for image $\boldsymbol{x}$. The goal of an adversarial attack is to find an adversarial example $\tilde{\boldsymbol{x}} \in \mathbb{X}^N$ such that $F(\tilde{\boldsymbol{x}}) \neq y(\boldsymbol{x})$ (untargeted attack) or $F(\tilde{\boldsymbol{x}}) = t$ (targeted attack, where t is a given target label, and $t \neq y(\boldsymbol{x})$), subject to the condition that $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_v \leq \epsilon$, where $\boldsymbol{x}$ is a correctly classified image. Here, ϵ refers to the allowed maximum perturbation strength and v refers to the norm used to measure the perturbation strength, such as l_1, l_2, and l_∞ norms (Long et al. 2022). In this paper, we adopt the l_∞ norm, following RayS (Chen and Gu 2020). This paper focuses primarily on untargeted attacks, aiming to force the target image classification models to produce arbitrary incorrect class labels for any image $\boldsymbol{x}$. This goal can be expressed as follows:

$$\max_{\tilde{\boldsymbol{x}}} f(\boldsymbol{x}, y(\boldsymbol{x}), \tilde{\boldsymbol{x}}) \quad \text{subject to} \quad \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\infty \leq \epsilon \tag{1}$$

$$\text{where} \quad f(\boldsymbol{x}, y(\boldsymbol{x}), \tilde{\boldsymbol{x}}) = \begin{cases} 0 & \text{if } F(\tilde{\boldsymbol{x}}) \neq y(\boldsymbol{x}) \\ -1 & \text{if } F(\tilde{\boldsymbol{x}}) = y(\boldsymbol{x}) \end{cases}$$

Let the adversarial perturbation $\tilde{\boldsymbol{x}} - \boldsymbol{x} = r \cdot \boldsymbol{d}$, where $\boldsymbol{d} \in [-1,1]^N$ represents the perturbation direction and $r \in [0,\epsilon]$ represents the perturbation strength. Prior works (Ilyas et al. 2018; Chen and Gu 2020; Cheng et al. 2018; Moon, An, and Song 2019; Chen et al. 2020), proved that the optimal solution to question (1) is at the extremities of the ℓ_∞ norm ball, i.e., $\boldsymbol{d} \in \{-1,1\}^N$. In this case, $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\infty = \|r \cdot \boldsymbol{d}\|_\infty = r \cdot \|\boldsymbol{d}\|_\infty = r$. Hence, the continuous optimization problem in question (1) is transformed into a mixed-integer optimization problem according to (Chen and Gu 2020):

$$\max_{r\in[0,\epsilon], \boldsymbol{d}\in\{-1,1\}^N} h(\boldsymbol{x}, y(\boldsymbol{x}), r, \boldsymbol{d}) \tag{2}$$

$$\text{where} \quad h(\boldsymbol{x}, y(\boldsymbol{x}), r, \boldsymbol{d}) = \begin{cases} 0 & \text{if } F(\boldsymbol{x} + r \cdot \boldsymbol{d}) \neq y(\boldsymbol{x}) \\ -1 & \text{if } F(\boldsymbol{x} + r \cdot \boldsymbol{d}) = y(\boldsymbol{x}) \end{cases}$$

Previous works (Cheng et al. 2020; Ilyas et al. 2018; Chen and Gu 2020; Cheng et al. 2018) also proved that the solution space of question (2) is continuous, and exploited the concept of the decision boundary to develop effective adversarial attacks. For any direction $\boldsymbol{d}$, there exists a *decision boundary* $g(\boldsymbol{d})$ such that $F(\boldsymbol{x} + g(\boldsymbol{d}) \cdot \boldsymbol{d}) \neq y(\boldsymbol{x})$ and $F(\boldsymbol{x} + (g(\boldsymbol{d}) - \Delta) \cdot \boldsymbol{d}) = y(\boldsymbol{x}), \exists \Delta \in (0, \tau]$. $g(\boldsymbol{d})$ is further defined in question (3) below. As shown in Figure 1(a), $g(\boldsymbol{d}_{\text{best}})$, $g(\boldsymbol{d_1})$, and $g(\boldsymbol{d_2})$ are the decision boundaries of the previous best direction $\boldsymbol{d}_{\text{best}}$ and two new candidate directions $\boldsymbol{d_1}$, $\boldsymbol{d_2}$, respectively. According to the definition of current decision boundary in (Chen and Gu 2020), the decision-based attacks have the goal to find the optimal perturbation direction $\boldsymbol{d}_{\text{best}}$:

$$\boldsymbol{d}_{\text{best}} = \operatorname*{argmin}_{\boldsymbol{d}\in\{-1,1\}^N} g(\boldsymbol{d}) \tag{3}$$
$$\text{with} \quad g(\boldsymbol{d}) = \inf\{F(\boldsymbol{x} + r \cdot \boldsymbol{d}) \neq y(\boldsymbol{x})\}$$

According to the above equation, the direction $\boldsymbol{d_2}$ in Figure 1(a) has the smallest decision boundary $g(\boldsymbol{d_2})$ among $g(\boldsymbol{d}_{\text{best}})$, $g(\boldsymbol{d_1})$, and $g(\boldsymbol{d_2})$.

Approximation Decision Boundary Approach

To compare the decision boundary $g(\boldsymbol{d})$ of different directions, existing methods typically employ exact search techniques such as binary search (Cheng et al. 2020; Ilyas et al. 2018; Chen and Gu 2020) to identify $g(\boldsymbol{d})$ with high accuracy (e.g., up to 0.001 (Chen and Gu 2020)). Obviously, this process requires a large number of queries. We notice that, in the early stages of optimization, directions are often far away from optima and their decision boundaries often exceed the threshold ϵ on the perturbation strength. Thus it is unnecessary to precisely identify the decision boundary. However, existing works (Cheng et al. 2018, 2020; Chen and Gu 2020) perform numerous queries to determine the accurate decision boundary, even in the early stages, resulting in poor query efficiency. Different from these works, in this paper, we demonstrate that perturbation directions can be reliably compared and optimized without knowing the precise decision boundary. Driven by this idea, we propose ADBA, as summarized in Algorithm 1.

In Algorithm 1, ADBA iteractively searches for the optimal perturbation direction. It keeps track of the current best direction $\boldsymbol{d}_{\text{best}}$ along with its ADB r_{best} , and update $\boldsymbol{d}_{\text{best}}$ and r_{best} in each iteration. The initial perturbation direction $\boldsymbol{d}_{\text{best}}$ is flattened into a one-dimensional vector $(1,\dots,1)$ with N = Width $\times$ Height $\times$ Channels dimensions. The current best perturbation strength r_{best} that represents the ADB of $\boldsymbol{d}_{\text{best}}$ is set to 1 initially to make sure that

Algorithm 1: Approximate Decision Boundary Approach

Input: Model F, the original image $\boldsymbol{x}$ and its label $y(\boldsymbol{x})$, and maximum perturbation strength ϵ;
Output: Optimal direction $\boldsymbol{d}_{\text{best}}$ with approximation decision boundary r_{best};
Initialization: Initialize current best direction $\boldsymbol{d}_{\text{best}} \leftarrow (1, \ldots, 1)$, and set current best perturbation strength $r_{\text{best}} \leftarrow 1$, block variable $s \leftarrow 0$ and block index $k \leftarrow 0$;

1: **while** remaining query budget > 0 and $r_{\text{best}} > \epsilon$ **do**
2: $L = N/2^{(s+1)}$;
3: $\boldsymbol{d_1} \leftarrow \boldsymbol{d}_{\text{best}}$, $\boldsymbol{d_2} \leftarrow \boldsymbol{d}_{\text{best}}$;
4: **for** $i \in [k \cdot L, (k+1) \cdot L)$ **do**
5: $\boldsymbol{d_1}[i] \leftarrow -\boldsymbol{d_1}[i]$
6: **end for**
7: **for** $i \in [(k+1) \cdot L, (k+2) \cdot L)$ **do**
8: $\boldsymbol{d_2}[i] \leftarrow -\boldsymbol{d_2}[i]$
9: **end for**
10: $\boldsymbol{d}_{\text{best}}, r_{\text{best}} \leftarrow$ Algorithm 2$(F, \{\boldsymbol{x}, y(\boldsymbol{x})\}, \boldsymbol{d}_{\text{best}}, r_{\text{best}},$
11: $\boldsymbol{d_1}, \boldsymbol{d_2})$;
12: $k \leftarrow k + 2$;
13: **if** $k = 2^{(s+1)}$ **then**
14: $s \leftarrow s + 1$
15: $k \leftarrow 0$
16: **end if**
17: **end while**
18: **return** $\boldsymbol{d}_{\text{best}}, r_{\text{best}}$

$F(\boldsymbol{x} + r_{\text{best}} \cdot \boldsymbol{d}_{\text{best}}) \neq y(\boldsymbol{x})$. In line with RayS (Chen and Gu 2020), we use the block variable s to control the block size. In line 2, the current best direction vector $\boldsymbol{d}_{\text{best}}$ is divided into $2^{(s+1)}$ smaller blocks. In lines 3-9 of each algorithm iteration, we choose two blocks of $\boldsymbol{d}_{\text{best}}$ in sequence and reverse the sign of each block respectively to create two new directions, $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$. In line 10, different from (Chen and Gu 2020) that uses binary search to accurately identify $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$, we devise Algorithm 2 (to be detailed in the next section) to update the optimal direction $\boldsymbol{d}_{\text{best}}$ based on its approximation decision boundary r_{best}. In lines 11-15, if the block index k reaches $2^{(s+1)}$, i.e., all current blocks have been utilized for reversal, then $2^{(s+2)}$ more directions are produced in subsequent iterations by $s \leftarrow s + 1$. If the ADB r_{best} of $\boldsymbol{d}_{\text{best}}$ is within the allowed strength ϵ or the number of queries exceeds the budget, Algorithm 1 returns the current best direction together with its perturbation strength. The former scenario indicates a successful attack, while the latter signifies a failed attack.

In

Comparing Perturbation Directions Using ADB In this section, we propose a query-efficient method by comparing perturbation directions using ADB, rather than using binary search in (Chen and Gu 2020). Our idea is described in more details below. First, we determine whether $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ are superior to $\boldsymbol{d}_{\text{best}}$; if not, $\boldsymbol{d}_{\text{best}}$ remains the current best, and there is no need to compare $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$. We denote the ADB r_{best} of $\boldsymbol{d}_{\text{best}}$ as ADB_1 and check whether both $F(\boldsymbol{x} + \text{ADB}_1 \cdot \boldsymbol{d_1})$ and $F(\boldsymbol{x} + \text{ADB}_1 \cdot \boldsymbol{d_2})$ produce

Algorithm 2: Compare Directions Using ADB

Input: Model F, original image $\boldsymbol{x}$ and its label $y(\boldsymbol{x})$, current best direction $\boldsymbol{d}_{\text{best}}$ with approximation decision boundary r_{best}, two new directions $\boldsymbol{d_1}$, $\boldsymbol{d_2}$, and search tolerance τ;
Output: New best direction $\boldsymbol{d}_{\text{best}}$ with approximation decision boundary r_{best}
Initialization: Set ADB $\leftarrow r_{\text{best}}$, *start* $\leftarrow 0$, *end* $\leftarrow r_{\text{best}}$

1: **if** $F(\boldsymbol{x} + r_{\text{best}} \cdot \boldsymbol{d_1}) = y(\boldsymbol{x})$ and $F(\boldsymbol{x} + r_{\text{best}} \cdot \boldsymbol{d_2}) = y(\boldsymbol{x})$ **then**
2: **return** $\boldsymbol{d}_{\text{best}}, r_{\text{best}}$
3: **end if**
4: **while** *end* - *start* $> \tau$ **do**
5: **if** $F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_1}) \neq y(\boldsymbol{x})$ and $F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_2}) \neq y(\boldsymbol{x})$ **then**
6: ADB $\leftarrow$ (*start* + *end*)/2
7: *end* $\leftarrow$ ADB
8: **else if** $F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_1}) = y(\boldsymbol{x})$ and $F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_2}) = y(\boldsymbol{x})$ **then**
9: ADB $\leftarrow$ (*start* + *end*)/2
10: *start* $\leftarrow$ ADB
11: **else if** $F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_1}) \neq y(\boldsymbol{x})$ and $F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_2}) = y(\boldsymbol{x})$ **then**
12: **return** $\boldsymbol{d_1}$, ADB
13: **else if** $F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_1}) = y(\boldsymbol{x})$ and $F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_2}) \neq y(\boldsymbol{x})$ **then**
14: **return** $\boldsymbol{d_2}$, ADB
15: **end if**
16: **end while**
17: **return** $\boldsymbol{d_1}$, ADB

wrong class labels (attacks are successful). If ADB_1 is too large and both attacks succeed, as shown in Figure 1(a), then $g(\boldsymbol{d_1}) \leq \text{ADB}_1$ and $g(\boldsymbol{d_2}) \leq \text{ADB}_1$. In this case, we cannot differentiate $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$. Hence, ADB is decreased to $\text{ADB}_2 = (0 + \text{ADB}_1)/2$. Subsequently, we check whether $F(\boldsymbol{x} + \text{ADB}_2 \cdot \boldsymbol{d_1})$ and $F(\boldsymbol{x} + \text{ADB}_2 \cdot \boldsymbol{d_2})$ can lead to incorrect classification. If ADB_2 is too small and both attacks fail, as demonstrated in Figure 1(b), $g(\boldsymbol{d_1}) > \text{ADB}_2$ and $g(\boldsymbol{d_2}) > \text{ADB}_2$. In this case, we still cannot differentiate $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$. We instead increase ADB to $\text{ADB}_3 = (\text{ADB}_2 + \text{ADB}_1)/2$, and check whether $F(\boldsymbol{x} + \text{ADB}_3 \cdot \boldsymbol{d_1})$ and $F(\boldsymbol{x} + \text{ADB}_3 \cdot \boldsymbol{d_2})$ can induce wrong classification. This time, in Figure 1(c), the attack along the direction $\boldsymbol{d_2}$ succeeds, and the attack along the direction $\boldsymbol{d_1}$ fails, indicating $g(\boldsymbol{d_2}) \leq \text{ADB}_3 < g(\boldsymbol{d_1})$; i.e., $\boldsymbol{d_2}$ is superior to $\boldsymbol{d_1}$. Subsequently, we update $\boldsymbol{d}_{\text{best}} = \boldsymbol{d_2}$, $r_{\text{best}} = \text{ADB}_3$.

The above process is summarized in Algorithm 2. Notably, for the current best direction $\boldsymbol{d}_{\text{best}}$ in Figure 1(a), because its real decision boundary $g(\boldsymbol{d}_{\text{best}}) \leq r_{\text{best}}$, we cannot compare $g(\boldsymbol{d}_{\text{best}})$ and $g(\boldsymbol{d_2})$. However, further comparison of $\boldsymbol{d}_{\text{best}}$ and $\boldsymbol{d_2}$ is unnecessary, since this requires twice as many queries with minimal improvement of $g(\boldsymbol{d}_{\text{best}})$, as evidenced by our preliminary experiments in Appendix C.

In Algorithm 2, ADB is set initially to the ADB of the current best direction $\boldsymbol{d}_{\text{best}}$. In lines 1 and 2, if $g(\boldsymbol{d}_{\text{best}}) \leq r_{\text{best}} < g(\boldsymbol{d_1})$ and $g(\boldsymbol{d}_{\text{best}}) \leq r_{\text{best}} < g(\boldsymbol{d_2})$ (i.e., $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ are worse than $\boldsymbol{d}_{\text{best}}$), it is worthless to perform further

comparisons. Algorithm 2 hence returns $\boldsymbol{d}_{\text{best}}$ as its output. In lines 5 to 10, if the decision boundaries of $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ are both either smaller or larger than ADB, then Algorithm 2 updates the *start* or *end* to narrow the search range for ADB. In lines 11 to 14, if the current ADB leads to a successful attack in one direction but not the other direction, Algorithm 2 reports the successful direction and the corresponding ADB. In line 17, if the search range *end* - *start* is less than a search tolerance threshold τ, indicating that $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$ are closely matched, the algorithm returns $\boldsymbol{d_1}$ along with the current ADB. Meanwhile, according to Appendix C, it is unnecessary to compare $\boldsymbol{d_1}$ with $\boldsymbol{d}_{\text{best}}$ because this comparison requires an excessive number of queries while yielding minimal improvement to $g(\boldsymbol{d}_{\text{best}})$.

Optimization of ADB The practical efficiency of Algorithm 2 depends on the swift identification of ADB that can differentiate any two directions $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ accurately. In lines 6 and 9 of Algorithm 2, if the current ADB fails to differentiate the two directions, the next search point is chosen as the middle point of the search range $[start, end]$, i.e., ADB $\leftarrow (start + end)/2$. However, this simple method cannot achieve the desired query efficiency. In this paper, by considering the statistical distribution of decision boundaries, we can identify a more suitable ADB to improve the likelihood of differentiating $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$. Based on this idea, we propose ADBA-md in this section.

We define the random events A, B, C and D as follows:

$$\begin{cases} A = \{F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_1}) \neq y\} = \{g(\boldsymbol{d_1}) \leq \text{ADB}\} \\ B = \{F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_2}) \neq y\} = \{g(\boldsymbol{d_2}) \leq \text{ADB}\} \\ C = \{F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_1}) = y\} = \{g(\boldsymbol{d_1}) > \text{ADB}\} \\ D = \{F(\boldsymbol{x} + \text{ADB} \cdot \boldsymbol{d_2}) = y\} = \{g(\boldsymbol{d_2}) > \text{ADB}\} \end{cases} \tag{4}$$

We assume that $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$ are independent random variables that follow an identical probability distribution $\rho(r)$. The eligibility of adopting this assumption is clarified in Appendix D. The probability $P(A), P(B), P(C), P(D)$ for any ADB $\in [start, end]$ can be calculated respectively by:

$$\begin{cases} P(A) = P(B) = a = \int_{start}^{\text{ADB}} \rho(r)\, dr \\ P(C) = P(D) = b = \int_{\text{ADB}}^{end} \rho(r)\, dr \end{cases} \tag{5}$$

Accordingly, $P(B \cap C) = P(B) \cdot P(C \mid B)$ gives the probability of $g(\boldsymbol{d_2}) \leq \text{ADB} < g(\boldsymbol{d_1})$, as illustrated in Figure 1(c). To avoid the complexity of determining $P(C \mid B)$, which requires extra queries, $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$ are assumed to be independent. Therefore,

$$\begin{cases} P(B \cap C) = P(B)P(C \mid B) = P(B)P(C) = ab \\ P(A \cap D) = P(A)P(D \mid A) = P(A)P(D) = ab \\ P(A \cap B) = P(A)P(B \mid A) = P(A)P(B) = a^2 \\ P(C \cap D) = P(C)P(D \mid C) = P(C)P(D) = b^2 \end{cases} \tag{6}$$

Consequently, in each comparison attempt, the probability $P(SUCC)$ for ADB to successfully differentiate a pair of directions $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ and the probability $P(FAIL)$ for ADB to fail to do so are determined respectively by:

$$\begin{cases} P(SUCC) = P(B \cap C) + P(A \cap D) = 2ab, \\ P(FAIL) = P(A \cap B) + P(C \cap D) = a^2 + b^2 \end{cases} \tag{7}$$

$$\text{with } P(SUCC) + P(FAIL) = 1 \text{ and } a + b = 1$$

$P(SUCC)$ reaches its maximum by setting $a = b = 0.5$, resulting in $P(SUCC) = P(FAIL) = 1/2$. Hence, according to Eqn. (5), ADB should be set to divide the probability density function $\rho(r)$ into two parts of equal size, instead of setting ADB to $(start + end)/2$ at lines 6 and 9 in Algorithm 2. Specifically, we can set ADB to satisfy the following equation, without querying the target model:

$$\int_{start}^{\text{ADB}} \rho(r)\, dr = \int_{\text{ADB}}^{end} \rho(r)\, dr = \frac{1}{2} \cdot \int_{start}^{end} \rho(r)\, dr \tag{8}$$

Especially, if the probability distribution is uniformly identical, then ADB should be set to the mid point between $start$ and end. The expected number of queries required for differentiating any pair of perturbation directions can be kept small. In fact, it is straightforward to show that $P(SUCC) = 1/2$. If differentiation fails in one comparison attempt, in the next attempt, Algorithm 2 updates $start$ and end according to lines 7 and 10, and then sets ADB to satisfy Eqn. (8). Hence, $P(SUCC)$ remains to be $1/2$. Assuming that differentiation succeeds after C comparison attempts, which means that the first $C-1$ comparison attempts fail and the C-th comparison attempt succeeds, the probability that C equals any given n is: $P(C = n) = (P(FAIL))^{n-1} \cdot P(SUCC) = a^{n-1} \cdot b = \left(\frac{1}{2}\right)^n$. Hence, the expected number of comparison attempts $\bar{C}$ can be determined as $\bar{C} = 1 \cdot \left(\frac{1}{2}\right)^1 + 2 \cdot \left(\frac{1}{2}\right)^2 \cdots + n \cdot \left(\frac{1}{2}\right)^n = \sum_{n=1}^{\infty} n \cdot \left(\frac{1}{2}\right)^n = 2$. Therefore, 2 consecutive guesses of ADB on average are required to differentiate a pair of directions. Meanwhile, for each comparison attempt, a total of 2 queries are performed using the perturbations ADB$\cdot\boldsymbol{d_1}$ and ADB$\cdot\boldsymbol{d_2}$ respectively in line 1, 5, 8, 11, and 13 of Algorithm 2. In total, the average number of queries required to differentiate any two directions is $\bar{Q} = 2 \times \bar{C} = 2 \times 2 = 4$.

Since Eqn. (6) is obtained by assuming that $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$ are independent, the actual number of queries required to differentiate any two directions may not be 4. However, the experiment results in Table 2 show that the number of queries achieved by ADBA-md is very close to 4 in practice. Compared with RayS that requires on average 10 queries to reach an accuracy level of $0.001 \approx 2^{-10}$ through binary search in our experiments, ADBA can noticeably reduce the required number of queries by approximately 2.5 times.

In Appendix D, in order to estimate the probability density function ρ, we conduct a statistical analysis of the distribution of the actual decision boundaries r_{best}, $g(\boldsymbol{d_1})$ and $g(\boldsymbol{d_2})$ under various settings of $\boldsymbol{d}_{\text{best}}$, $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$. Subsequently, we utilize an inverse proportional function, i.e., $\rho(r) = \frac{a}{(|-r/r_{\text{best}}+d|)^b+c}$, $r \in [0, r_{\text{best}}]$, to estimate the true distribution of the decision boundary, where a, b, c, and d are the estimation parameters, r_{best} is the ADB of the current best direction $\boldsymbol{d}_{\text{best}}$. The efficiency of ADBA-md is insensitive to the accuracy of $\rho(r)$. Hence, ADBA-md can be easily applied to other datasets, as further supported by additional experimental results reported in Appendix E, as well as several adversarial training-based defense models (Wang et al. 2023) discussed in Appendix F. ADBA and ADBA-md do not incorporate random factors. Therefore, for the same original image,

Model / Attack	VGG -19	ResNet -50	Inception -V3	ViT -B32	DenseNet -161	EfficientNet -B0
OPT	1473.0(1191)	2096.8(1387)	2213.6(1419)	3291.7(2158)	3373.6(2350)	2825.4(1813)
	25.1%	9.2%	18.1%	16.2%	12.9%	10.8%
SignOPT	2575.2(1765)	3042.3(1727)	2253.3(1337)	3499.4(1969)	3943.7(3051)	3031.9(1897)
	42.5%	22.5%	35.0%	30.8%	25.2%	17.6%
HSJA	589.1(324)	1220.0(401)	733.6(358)	996.9(422)	1891.0(812)	1728.3(559)
	29.5%	12.7%	21.9%	18.8%	16.1%	13.9%
RayS	470.1(295)	1243.5(721)	672.2(339)	1005.1(414)	733.6(358)	669.7(321)
	100%	98.8%	98.8%	98.8%	99.1%	**99.8%**
ADBA	237.3(119)	793.8(332)	449.0(149)	693.7(202)	485.7(158)	360.5(**132**)
	100%	99.1%	**99.8%**	98.9%	99.2%	99.7%
ADBA-md	**197.6(115)**	**685.6(273)**	**377.0(148)**	**629.1(188)**	**457.8(149)**	**310.9(132)**
	100%	**99.5%**	**99.8%**	**99.7%**	**99.7%**	**99.8%**

Table 1: Comparison of decision-based untargeted attacks on the ImageNet dataset with 10,000 query budgets and maximum l_∞ perturbation strengths $\epsilon = 0.05$.

		VGG -19	ResNet -50	Inception -V3	ViT -B32	DenseNet -161	EfficientNet -B0
ADBA	**I**	49.90	168.98	103.25	170.42	112.87	94.34
	Q/I	4.755	4.552	4.468	4.509	4.303	4.540
	Q	237.3	769.3	461.3	768.4	485.7	428.3
ADBA -md	**I**	50.92(↑2.0%)	197.5(↑16.9%)	118.2(↑14.5%)	184.3(↑8.2%)	133.2(↑18.0%)	95.39(↑1.1%)
	Q/I	3.880(↓18.4%)	3.472(↓23.7%)	3.538(↓20.8%)	3.537(↓21.6%)	3.436(↓20.1%)	3.598(↓20.7%)
	Q	197.6(↓16.7%)	685.6(↓10.9%)	418.3(↓9.32%)	651.9(↓15.2%)	457.8(↓5.74%)	343.2(↓19.9%)

Table 2: The Average number of iterations (I), queries for each iteration (Q/I), and total queries (Q) performed by ADBA and ADBA-md on six contemporary image classification models.

ADBA and ADBA-md will consistently generate the same adversarial examples across multiple repeated experiments.

Experiments

Experiment Settings

In this section, we conduct comprehensive experiments to compare ADBA and ADBA-md against several state-of-the-art decision-based attack approaches, including OPT (Cheng et al. 2018), SignOPT (Cheng et al. 2020), HSJA (Chen, Jordan, and Wainwright 2020), and RayS (Chen and Gu 2020). Our experiments are conducted on a server with an Intel Xeon Gold 6330 CPU, NVIDIA 4090 GPUs using PyTorch 2.3.0, Torchvision 0.18.0 on the Python 3.11.5 platform.

Datasets and target models. We evaluate the performance of ADBA and ADBA-md on the ImageNet dataset (Deng et al. 2009). Our evaluation includes six contemporary machine learning models that are commonly used as attack target models (Chen and Gu 2020; Cheng et al. 2020; Li et al. 2021; Reza et al. 2023): VGG19 (Simonyan and Zisserman 2015), ResNet50 (He et al. 2016), Inception-V3 (Szegedy et al. 2016), ViT-B32 (Dosovitskiy et al. 2021), DenseNet161 (Huang et al. 2017), and EfficientNet-B0 (Tan and Le 2019).

Attack settings. Following cutting-edge research on black-box adversarial attacks (Chen and Gu 2020; Moon, An, and Song 2019), we adopt the l_∞ norm and set the perturbation strength threshold $\epsilon = 0.05$. Meanwhile, the query budget is set to 10000 for each model (Chen and Gu 2020). The parameters in the distribution function $\rho(r) = \frac{a}{(|-r/r_{\text{best}}+d|)^b+c}$, $r \in [0, r_{\text{best}}]$ are set to be $a = 0.0313$, $b = 3.066$, $c = 0.168$, and $d = 1.134$ according to Appendix D. The search tolerance threshold $\tau = 10^{-5}$.

Experiment Results

For six pre-trained models, we randomly select 1,000 correctly classified images from the ImageNet test set for each model. Then we produce adversarial examples for each image by using six hard-label attacks respectively to determine the attack success rate, which is calculated by $\frac{|E_{\text{successful}}|}{|E|}$. Here, E refers to the set of all test cases ($|E| = 1000$) and $E_{\text{successful}}$ is the set of successful attacks that meet the requirements for the query budget (i.e., 10000) and the perturbation strength threshold (i.e., $\epsilon = 0.05$). In Table 1, we summarize the attack success rate and the average (median) number of queries for six attack methods across six target models. ADBA-md provides the highest attack success rate, exceeding 99.5% across all target models, and requires the fewest average (median) number of queries across six black-box attacks. Compared to other attack approaches, our methods significantly reduce the average number of queries to below 800 and the median number of queries to below 400.

Under similar attack success rates, both ADBA and ADBA-md can significantly reduce the average and median number of queries compared to RayS. Specifically, the average number of queries is approximately 60-70% of those required by

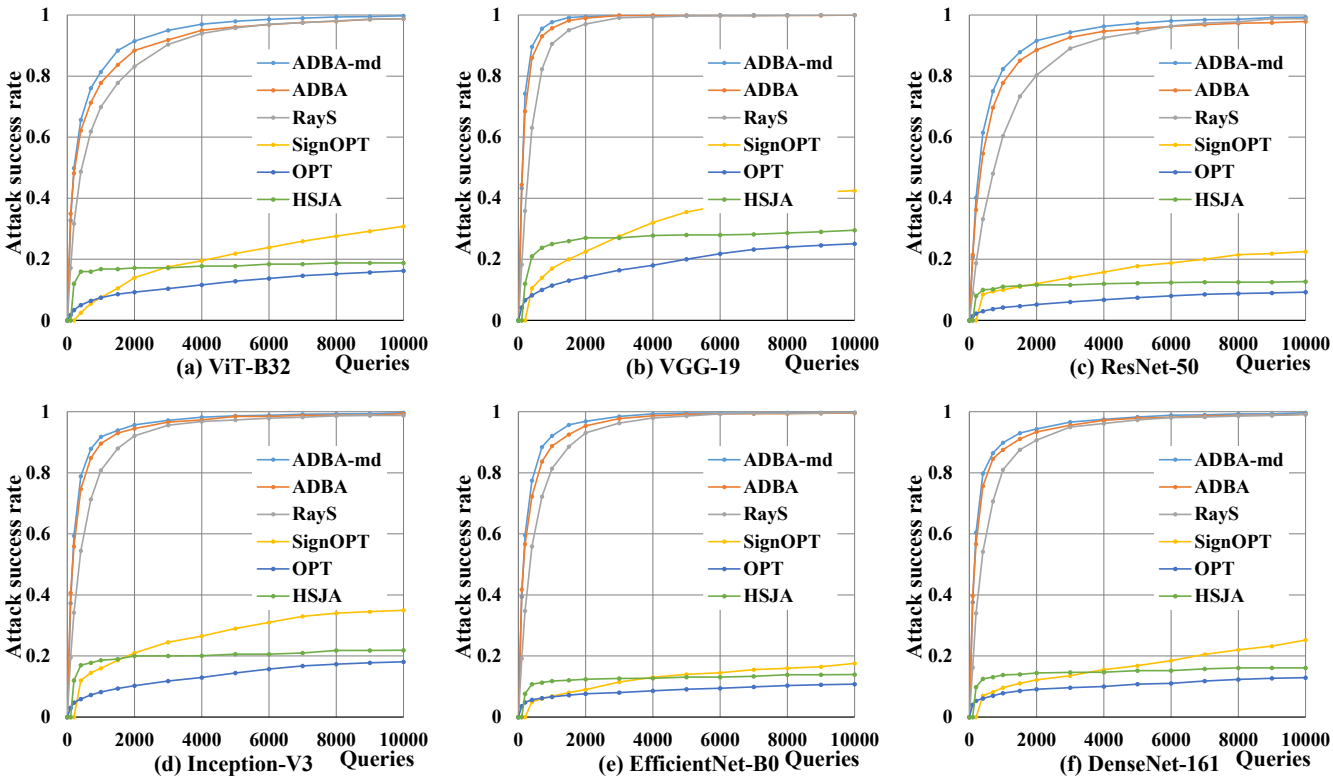


Figure 2: Attack success rates versus number of queries for six hard-label attacks on ImageNet.

RayS, and the median is reduced by 50%. Our results indicate that it is unnecessary to perform precise binary search as in RayS. Instead, it is much more efficient to optimize perturbation directions by using approximation decision boundaries.

Figure 2 presents a comparative performance analysis of six black-box methods on six target models, focusing on the attack success rate relative to the number of queries. ADBA and ADBA-md consistently achieve the highest attack success rates compared to other approaches. Notably, even with a very low query budget (1000 queries), both ADBA and ADBA-md attain an attack success rate of approximately 90%, whereas RayS achieves around 80%. The remaining three attack methods fail to obtain an attack success rate above 30%. This result confirms that our methods can achieve substantially higher success rate, making them highly effective under restricted query budgets.

In Table 2, we analyze the impact of the median search method on the efficiency of ADBA-md. For all six models, ADBA-md requires a higher average number of search iterations (indicated by the upward arrows in %Change of Iteration I), yet it has less number of queries per iteration (shown with downward arrows in %Change of Q/I). Consequently, ADBA-md requires less number queries in total (as denoted by the downward arrows in %Change of Queries Q). The increase in iterations for ADBA-md is attributed to the reduced number of queries in each iteration, leading to a smaller improvement in the ADBs per iteration. Therefore, more iterations are required to reach the perturbation strength threshold ϵ.

Conclusions

Decision-based black-box attacks are particularly relevant in practice as they rely solely on the hard label to generate adversarial examples. In this paper, we proposed a new Approximation Decision Boundary Approach (ADBA) for query-efficient decision-based attacks. Our approach introduced a novel insight: it is feasible to compare and optimize perturbation directions even without exact knowledge of the decision boundaries. Specifically, given any Approximation Decision Boundary (ADB), if one perturbation direction causes the model to fail while the other direction does not, the former direction is deemed superior. Driven by this insight that has been consistently overlooked in the past research, we substantially improved the direction search framework by replacing the binary search for precise decision boundaries with a query-efficient method that searches for ADBs. Subsequently, we analyzed the distribution of decision boundaries and developed ADBA-md to noticeably improve the efficiency of approximating decision boundaries. Our experiments on six image classification models clearly validated the effectiveness of our ADBA and ADBA-md approaches in enhancing the attack success rate. In the future, we plan to study a wide range of adversarial defense strategies, such as Barrage of Random Transforms (Raff et al. 2019) and Barrier Zones (Mahmood et al. 2022).

Acknowledgments

This work was partially supported by BUPT Excellent Ph.D. Students Foundation, foundation number is CX20241003. The authors express their gratitude to the anonymous reviewers for their insightful and constructive feedback on the initial version of this paper.

References

Bai, Y.; Wang, Y.; and Zeng, Y. 2023. Query efficient black-box adversarial attack on deep neural networks. *Pattern Recognition*, 133: 109037. Publisher: Elsevier.

Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. ArXiv:1712.04248 [cs, stat].

Brunner, T.; Diehl, F.; Le, M. T.; and Knoll, A. 2019. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4958–4966.

Chen, J.; and Gu, Q. 2020. RayS: A Ray Searching Method for Hard-label Adversarial Attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1739–1747. Virtual Event CA USA: ACM. ISBN 978-1-4503-7998-4.

Chen, J.; Jordan, M. I.; and Wainwright, M. J. 2020. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, 1277–1294. IEEE.

Chen, J.; Zhou, D.; Yi, J.; and Gu, Q. 2020. A frank-wolfe framework for efficient and effective adversarial attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 3486–3494. Issue: 04.

Cheng, M.; Le, T.; Chen, P.-Y.; Zhang, H.; Yi, J.; and Hsieh, C.-J. 2018. Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach. In *International Conference on Learning Representations*.

Cheng, M.; Singh, S.; Chen, P.; Chen, P.-Y.; Liu, S.; and Hsieh, C.-J. 2020. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack. ArXiv:1909.10773 [cs, stat].

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. ISSN: 1063-6919.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ArXiv:2010.11929 [cs].

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1625–1634.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, 2137–2146. PMLR.

Li, H.; Xu, X.; Zhang, X.; Yang, S.; and Li, B. 2020. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1221–1230.

Li, J.; Ji, R.; Chen, P.; Zhang, B.; Hong, X.; Zhang, R.; Li, S.; Li, J.; Huang, F.; and Wu, Y. 2021. Aha! adaptive history-driven attack for decision-based black-box models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16168–16177.

Li, S.; and Deng, W. 2020. Deep facial expression recognition: A survey. *IEEE transactions on affective computing*, 13(3): 1195–1215.

Li, S.; Zhang, S.; Chen, G.; Wang, D.; Feng, P.; Wang, J.; Liu, A.; Yi, X.; and Liu, X. 2023a. Towards benchmarking and assessing visual naturalness of physical world adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12324–12333.

Li, Y.; Li, Y.; Dai, X.; Guo, S.; and Xiao, B. 2023b. Physical-world optical adversarial attacks on 3d face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24699–24708.

Lin, C.; Han, S.; Zhu, J.; Li, Q.; Shen, C.; Zhang, Y.; and Guan, X. 2023. Sensitive region-aware black-box adversarial attacks. *Information Sciences*, 637: 118929.

Liu, Y.; Moosavi-Dezfooli, S.-M.; and Frossard, P. 2019. A geometry-inspired decision-based attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4890–4898.

Long, T.; Gao, Q.; Xu, L.; and Zhou, Z. 2022. A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions. *Computers & Security*, 121: 102847.

Mahmood, K.; Nguyen, P. H.; Nguyen, L. M.; Nguyen, T.; and Van Dijk, M. 2022. Besting the Black-Box: Barrier Zones for Adversarial Example Defense. *IEEE Access*, 10: 1451–1474.

Maho, T.; Furon, T.; and Le Merrer, E. 2021. Surfree: a fast surrogate-free black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10430–10439.

Moon, S.; An, G.; and Song, H. O. 2019. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *International conference on machine learning*, 4636–4645. PMLR.

Raff, E.; Sylvester, J.; Forsyth, S.; and McLean, M. 2019. Barrage of Random Transforms for Adversarially Robust Defense. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6521–6530.

Rahmati, A.; Moosavi-Dezfooli, S.-M.; Frossard, P.; and Dai, H. 2020. Geoda: a geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8446–8455.

Reza, M. F.; Rahmati, A.; Wu, T.; and Dai, H. 2023. CGBA: Curvature-aware geometric black-box attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 124–133.

Shi, Y.; Han, Y.; Tan, Y.-a.; and Kuang, X. 2022. Decision-based black-box attack against vision transformers via patch-wise adversarial removal. *Advances in Neural Information Processing Systems*, 35: 12921–12933.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations (ICLR 2015)*. Publisher: Computational and Biological Learning Society.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tan, M.; and Le, Q. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, 6105–6114. PMLR. ISSN: 2640-3498.

Wang, X.; and He, K. 2021. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1924–1933.

Wang, X.; Zhang, Z.; Tong, K.; Gong, D.; He, K.; Li, Z.; and Liu, W. 2022. Triangle Attack: A Query-Efficient Decision-Based Adversarial Attack. In Avidan, S.; Brostow, G.; and Cissé, M., eds., *Computer Vision – ECCV 2022*, volume 13665, 156–174. Cham: Springer Nature Switzerland. Series Title: Lecture Notes in Computer Science.

Wang, Z.; Pang, T.; Du, C.; Lin, M.; Liu, W.; and Yan, S. 2023. Better diffusion models further improve adversarial training. In *International Conference on Machine Learning*, 36246–36263. PMLR.

Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. 2017. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, 1369–1378.

Xie, C.; Wu, Y.; Maaten, L. v. d.; Yuille, A. L.; and He, K. 2019. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 501–509.