RayS: A Ray Searching Method for Hard-label Adversarial Attack

Jinghui Chen
University of California, Los Angeles
jhchen@cs.ucla.edu

Quanquan Gu
University of California, Los Angeles
qgu@cs.ucla.edu

ABSTRACT

Deep neural networks are vulnerable to adversarial attacks. Among different attack settings, the most challenging yet the most practical one is the hard-label setting where the attacker only has access to the hard-label output (prediction label) of the target model. Previous attempts are neither effective enough in terms of attack success rate nor efficient enough in terms of query complexity under the widely used L_∞ norm threat model. In this paper, we present the Ray Searching attack (RayS), which greatly improves the hard-label attack effectiveness as well as efficiency. Unlike previous works, we reformulate the continuous problem of finding the closest decision boundary into a discrete problem that does not require any zeroth-order gradient estimation. In the meantime, all unnecessary searches are eliminated via a fast check step. This significantly reduces the number of queries needed for our hard-label attack. Moreover, interestingly, we found that the proposed RayS attack can also be used as a sanity check for possible "falsely robust" models. On several recently proposed defenses that claim to achieve the state-of-the-art robust accuracy, our attack method demonstrates that the current white-box/black-box attacks could still give a false sense of security and the robust accuracy drop between the most popular PGD attack and RayS attack could be as large as 28%. We believe that our proposed RayS attack could help identify falsely robust models that beat most white-box/black-box attacks.

CCS CONCEPTS

• **Computing methodologies** → **Discrete space search**; **Object recognition**.

KEYWORDS

robustness, deep neural networks, hard-label attacks

ACM Reference Format:
Jinghui Chen and Quanquan Gu. 2020. RayS: A Ray Searching Method for Hard-label Adversarial Attack. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403225

1 INTRODUCTION

Deep neural networks (DNNs) have achieved remarkable success on many machine learning tasks such as computer vision [15, 36], and speech recognition [17] in the last decade. Despite the great success, recent studies have shown that DNNs are vulnerable to adversarial examples, i.e., even imperceptible (specially designed not random) perturbations could cause the state-of-the-art classifiers to make wrong predictions [13, 38]. This intriguing phenomenon has soon led to an arms race between adversarial attacks [3, 5, 7] that are trying to break the DNN models with such small perturbations and adversarial defenses methods [27, 33, 40, 41, 46] that tries to defend against existing attacks. During this arm race, many heuristic defenses [12, 14, 26, 33–35, 42] are later proved to be not effective under harder attacks. One exception is adversarial training [13, 27], which was demonstrated as an effective defense approach.

A large body of adversarial attacks has been proposed during this arm race. According to the different amounts of information the attacker could access, adversarial attacks can be generally divided into three categories: white-box attacks, black-box attacks, and hard-label attacks. White-box attacks [5, 27] refer to the case where the attacker has access to all information regarding the target model, including the model weights, structures, parameters, and possible defense mechanisms. Since white-box attackers could access all model details, it can efficiently perform back-propagation on the target model and compute gradients. In black-box attacks, the attacker only has access to the queried soft label output (logits or probability distribution of different classes) of the target model, and the other parts are treated as a black-box. The black-box setting is much more practical compared with the white-box case, however, in such a setting, the attacker cannot perform back-propagation and direct gradient computation. Therefore, many turn to transfer the gradient from a known model [30] or estimate the true gradient via zeroth-order optimization methods [1, 7, 19, 20].

Hard-label attacks, also known as decision-based attacks, on the other hand, only allow the attacker to query the target model and get hard-label output (prediction label). Obviously, the hard-label setting is the most challenging one, yet it is also the most practical one, as in reality, there is little chance that the attacker could know all the information about the target model in advance or get the probability prediction of all classes. The hard-label-only access also means that the attacker cannot tell the subtle changes in the target model's output when feeding a slightly perturbed input sample (assuming this slight perturbation will not change the model prediction). Therefore, the attacker can only find informative clues around the decision boundary of the target model where tiny perturbations could cause the model to have different prediction labels. Previous works [4, 6, 9, 10] mostly follow this idea to tackle the hard-label adversarial attack problem. However, [4, 6, 9, 10] are all originally proposed for L_2 norm threat model while L_∞ norm threat models [21, 27, 44–46] are currently the most popular and widely used. Even though [6, 9, 10] provide extensions to L_∞ norm case, none of them has been optimized for the L_∞ norm case and consequently, their attack performance falls largely behind traditional L_∞ norm based white-box and black-box attacks, making

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08.
https://doi.org/10.1145/3394486.3403225

them inapplicable in real world scenarios. This leads to a natural question that,

Can we design a hard-label attack that could greatly improve upon previous hard-label attacks and provide practical attacks for the most widely used L_∞ norm threat model?

In this paper, we answer this question affirmatively. We summarize our main contributions as follows

- We propose the Ray Searching attack, which only relies on the hard-label output of the target model. We show that the proposed hard-label attack is much more effective and efficient than previous hard-label attacks in the L_∞ norm threat model.
- Unlike previous works, most of which solve the hard-label attack problem via zeroth-order optimization methods, we reformulate the continuous optimization problem of finding the closest decision boundary into a discrete one and directly search for the closest decision boundary along a discrete set of ray directions. A fast check step is also utilized to skip unnecessary searches. This significantly saves the number of queries needed for the hard-label attack. Our proposed attack is also free of hyperparameter tuning such as step size or finite difference constant, making itself very stable and easy to apply.
- Moreover, our proposed RayS attack can also be used as a strong attack to detect possible "falsely robust" models. By evaluating several recently proposed defenses that claim to achieve the state-of-the-art robust accuracy with RayS attack, we show that the current white-box/black-box attacks can be deceived and give a false sense of security. Specifically, the RayS attack significantly decrease the robust accuracy of the most popular PGD attack on several robust models and the difference could be as large as 28%. We believe that our proposed RayS attack could help identify falsely robust models that deceive current white-box/black-box attacks.

The remainder of this paper is organized as follows: in Section 2, we briefly review existing literature on adversarial attacks. We present our proposed Ray Searching attack (RayS) in Section 3. In Section 4, we show the proposed RayS attack is more efficient than other hard-label attacks and can be used as a sanity check for detecting falsely robust models by evaluating several recently proposed defenses. Finally, we conclude this paper and provide discussions in Section 5.

Notation. For a d-dimensional vector $\mathbf{x} = [x_1, ..., x_d]^\top$, we use $\|\mathbf{x}\|_0 = \sum_i \mathbb{1}\{x_i \neq 0\}$ to denote its ℓ_0-norm, use $\|\mathbf{x}\|_2 = (\sum_{i=1}^d |x_i|^2)^{1/2}$ to denote its ℓ_2-norm and use $\|\mathbf{x}\|_\infty = \max_i |x_i|$ to denote its ℓ_∞-norm, where $\mathbb{1}(\cdot)$ denotes the indicator function.

2 RELATED WORK

There is a large body of works on evaluating model robustness and generating adversarial examples. In this section, we review the most relevant works with ours.

White-box attacks: Szegedy et al. [38] first brought up the concept of adversarial examples and adopt the L-BFGS algorithm for attacks. Goodfellow et al. [13] proposed the Fast Gradient Sign Method (FGSM) method via linearizing the network loss function. Kurakin et al. [23] proposed to iteratively perform FGSM and conduct projection afterward, which is equivalent to Projected Gradient Descent (PGD) [27]. Papernot et al. [32] proposed JSMA method based on the Jacobian saliency map and Moosavi-Dezfooli et al. [29] proposed DeepFool attack by projecting the data to the closest separating hyper-plane. Carlini and Wagner [5] introduced the CW attack with a margin-based loss function and show that defensive distillation [33] is not truly robust. Chen et al. [7] proposed a projection-free attack based on the Frank-Wolfe method with momentum. Athalye et al. [3] identified the effect of obfuscated gradients and proposed the BPDA attack for breaking those obfuscated gradient defenses.

Black-box attacks: Other than the aforementioned white-box attack algorithms, there also exists a large body of literature [7, 8, 18–20, 25, 30, 31] focusing on the black-box attack case where the information is limited to the logits output of the model rather than every detail of the model. Transfer-based black-box attacks [18, 30, 31] try to transfer the gradient from a known model to the black-box target model and then apply the same technique as in the white-box case. However, their attack effectiveness is often not quite satisfactory. Optimization-based black-box attacks aim to estimate the true gradient via zeroth-order optimization methods. Chen et al. [8] proposed to estimate the gradient via finite-difference on each dimension. Ilyas et al. [19] proposed to improve the query efficiency of [8] via Natural Evolutionary Strategies. Ilyas et al. [20] further improved upon Ilyas et al. [19] by exploiting gradient priors. Uesato et al. [39] proposed to use the SPSA method to build a gradient-free attack that can break vanishing gradient defenses. Al-Dujaili and O'Reilly [1] proposed to directly estimate the sign of the gradient instead of the true gradient itself. Moon et al. [28] reformulated the continuous optimization problem into a discrete one and proposed a combinatorial search based algorithm to make the attack more efficient. Andriushchenko et al. [2] proposed a randomized search scheme to iteratively patch small squares onto the test example.

Hard-label attacks: Brendel et al. [4] first studied the hard-label attack problem and proposed to solve it via random walks near the decision boundary. Ilyas et al. [19] demonstrated a way to transform the hard-label attack problem into a soft label attack problem. Cheng et al. [9] turned the adversarial optimization problem into the problem of finding the optimal direction that leads to the shortest L_2 distance to decision boundary and optimized the new problem via zeroth-order optimization methods. Cheng et al. [10] further improved the query complexity of [9] by estimating the sign of gradient instead of the true gradient. Chen et al. [6] also applied zeroth-order sign oracle to improve [4] by searching the step size and keeping the iterates along the decision boundary.

3 THE PROPOSED METHOD

In this section, we introduce our proposed *Ray Searching attack* (RayS). Before we go into details about our proposed method, we first take an overview of the previous adversarial attack problem formulations.

3.1 Overview of Previous Problem Formulations

We denote the DNN model by f and the test data example as $\{\mathbf{x}, y\}$. The goal of adversarial attack is to solve the following optimization

problem

$$\min_{\mathbf{x}'} \mathbb{1}\{f(\mathbf{x}') = y\} \ \text{ s.t., } \ \|\mathbf{x}' - \mathbf{x}\|_\infty \leq \epsilon, \tag{3.1}$$

where ϵ denotes the maximum allowed perturbation strength. The indicator function $\mathbb{1}\{f(\mathbf{x}') = y\}$ is hard to optimize, therefore, [1, 7, 19, 20, 27, 46] turn to relax (3.1) into

$$\max_{\mathbf{x}'} \ell(f(\mathbf{x}'), y) \ \text{ s.t., } \ \|\mathbf{x}' - \mathbf{x}\|_\infty \leq \epsilon, \tag{3.2}$$

where ℓ denotes the surrogate loss function such as CrossEntropy loss. On the other hand, traditional hard-label attacks [9, 10] reformulate (3.1) as

$$\min_{\mathbf{d}} g(\mathbf{d}) \ \text{ where } \ g(\mathbf{d}) = \operatorname*{argmin}_r \mathbb{1}\{f(\mathbf{x} + r\mathbf{d}/\|\mathbf{d}\|_2) \neq y\}. \tag{3.3}$$

Here $g(\mathbf{d})$ represents the decision boundary radius from original example $\mathbf{x}$ along ray direction $\mathbf{d}$ and the goal is to find the minimum decision boundary radius regarding the original example $\mathbf{x}$. Let $(\widehat{r}, \widehat{\mathbf{d}})$ denotes the minimum decision boundary radius and the corresponding ray direction. If the minimum decision boundary radius satisfies $\|\widehat{r}\widehat{\mathbf{d}}/\|\widehat{\mathbf{d}}\|_2\|_\infty \leq \epsilon$, it will be counted as a successful attack.

While prior works [9, 10] try to solve problem (3.3) in a continuous fashion by estimating the gradient of $g(\mathbf{d})$ via zeroth-order optimization methods, the hard-label-only access restriction imposes great challenges in solving (3.3). Specifically, estimating the the decision boundary radius $g(\mathbf{d})$ typically takes a binary search procedure and estimating an informative gradient of $g(\mathbf{d})$ via finite difference requires multiple rounds of $g(\mathbf{d})$ computation. Furthermore, due to the large variance in zeroth-order gradient estimating procedure, optimizing (3.3) typically takes a large number of gradient steps. These together, make solving (3.3) much less efficient and effective than black-box attacks, not to mention white-box attacks.

Given all the problems mentioned above, we turn to directly search for the closest decision boundary without estimating any gradients.

3.2 Ray Search Directions

With a finite number of queries, it is impossible to search through the whole continuous ray direction space. As a consequence, we need to restrict the search space to a discrete set of ray directions to make direct searches possible. Note that applying FGSM to (3.2) leads to an optimal solution at the vertex of the L_∞ norm ball [7, 28], suggesting that those vertices might provide possible solutions to (3.2). Empirical findings in [28] also suggest that the solution to (3.2) obtained from the PGD attack is mostly found on the vertices of L_∞ norm ball. Inspired by this, Moon et al. [28] restrict the feasible solution set as the vertex of the L_∞ norm ball. Following this idea, since our goal is to obtain the decision boundary radius, we consider the ray directions that point to the L_∞ norm ball vertices, i.e., $\mathbf{d} \in \{-1, 1\}^d$ where d denotes the dimension of original data example $\mathbf{x}$[1]. Therefore, instead of solving (3.3), we turn to solve a discrete problem

$$\min_{\mathbf{d} \in \{-1,1\}^d} g(\mathbf{d}) \ \text{ where } \ g(\mathbf{d}) = \operatorname*{argmin}_r \mathbb{1}\{f(\mathbf{x} + r\mathbf{d}/\|\mathbf{d}\|_2) \neq y\}. \tag{3.4}$$

[1]Without loss of generality, here we view $\mathbf{x}$ simply as a d-dimensional vector.

In problem (3.4), we reduce the search space from $\mathbb{R}^d$ to $\{-1, 1\}^d$, which contains 2^d possible search directions.

Now we begin to introduce our proposed Ray Searching attack. We first present the naive version of the Ray Searching attack, which is summarized in Algorithm 1. Specifically, given a model f and a test example $\{\mathbf{x}, y\}$, we first initialize the best search direction as an all-one vector and set the initial best radius as infinity. Then we iteratively change the sign of each dimension of the current best ray direction and test whether this modified ray direction leads to a better decision boundary radius by Algorithm 2 (will be described later). If it does, we update the best search direction and the best radius, otherwise, they remain unchanged. Algorithm 1 is a greedy search algorithm that finds the local optima of the decision boundary radius, where the local optima of the decision boundary radius are defined as follows.

Definition 3.1 (Local Optima of Decision Boundary Radius)**.** A ray direction $\mathbf{d} \in \{-1, 1\}^d$ is the local optima of the decision boundary radius regarding (3.4), if for all $\mathbf{d}' \in \{-1, 1\}^d$ satisfy $\|\mathbf{d}' - \mathbf{d}\|_0 \leq 1$, we have $g(\mathbf{d}) \leq g(\mathbf{d}')$.

Theorem 3.2. Given enough query budgets, let $(\widehat{r}, \widehat{\mathbf{d}})$ be the output of Algorithm 1, then $\widehat{\mathbf{d}}$ is the local optima of decision boundary radius problem (3.4).

Proof. We prove this by contradiction. Suppose $\widehat{\mathbf{d}}$ is not the local optima, there must exist some $\mathbf{d}'$ satisfying $\|\mathbf{d}' - \widehat{\mathbf{d}}\|_0 \leq 1$, i.e., $\mathbf{d}'$ differs from $\widehat{\mathbf{d}}$ by at most 1 dimension, that $g(\widehat{\mathbf{d}}) > g(\mathbf{d}')$. This means Algorithm 1 can still find better solution than $g(\widehat{\mathbf{d}})$ by going through all dimensions and thus $\widehat{\mathbf{d}}$ will not be the output of Algorithm 1. This leads to a contradiction. □

Next we introduce Algorithm 2, which performs decision boundary radius search. The main body of Algorithm 2 (from Line 7 to Line 12) is a binary search algorithm to locate the decision boundary radius with high precision. The steps before Line 7, on the other hand, focus on deciding the search range and whether we need to search it (this is the key to achieve efficient attacks). Specifically, we first normalize the search direction by its L_2 norm. And then in Line 3, we do a fast check at $x + r_{\text{best}} \cdot \mathbf{d}_n$[2] and decide whether we need to further perform a binary search for this direction. To help better understand the underlying mechanism, Figure 1 provides a two-dimensional sketch for the fast check step in Line 3 in Algorithm 2. Suppose we first change the sign of the current $\mathbf{d}_{\text{best}}$ at dimension 1, resulting a modified direction $\mathbf{d}_{\text{tmp1}}$. The fast check shows that it is a valid attack and it has the potential to further reduce the decision boundary radius. On the other hand, if we change the sign of $\mathbf{d}_{\text{best}}$ at dimension 2, resulting a modified direction $\mathbf{d}_{\text{tmp2}}$. The fast check shows that it is no longer a valid attack and the decision boundary radius of direction $\mathbf{d}_{\text{tmp2}}$ can only be worse than the current r_{best}. Therefore, we skip all unnecessary queries that aim to estimate a worse decision boundary radius. Note that in Cheng et al. [10], a similar check was also presented for slightly perturbed directions. However, they use it as the sign for gradient estimation while we simply drop all unsatisfied radius based on the check result and obtain better efficiency. Finally, we explain Line 6 in Algorithm 2.

[2]For applications such as image classification, there is an additional clipping to $[0, 1]$ operation to keep the image valid. We assume this is included in model f and do not write it explicitly in Algorithm 2.

Algorithm 1 Ray Searching Attack (Naive)

input: Model f, Original data example $\{\mathbf{x}, y\}$;
Initialize current best search direction $\mathbf{d}_{\text{best}} = (1, \ldots, 1)$
Initialize current best radius $r_{\text{best}} = \infty$
Initialize ray searching index $k = 1$
while remaining query budget > 0 **do**
  $\mathbf{d}_{\text{tmp}} = \mathbf{d}_{\text{best}}.copy()$
  $\mathbf{d}_{\text{tmp}}[k] = -\mathbf{d}_{\text{tmp}}[k]$
  $r_{\text{tmp}} = \text{DBR-Search}(f, \mathbf{x}, y, \mathbf{d}_{\text{tmp}}, r_{\text{best}})$
  if $r_{\text{tmp}} < r_{\text{best}}$ **then**
    $r_{\text{best}}, \mathbf{d}_{\text{best}} = r_{\text{tmp}}, \mathbf{d}_{\text{tmp}}$
  end if
  $k = k + 1$
  if $k == d$ **then**
    $k = 1$
  end if
end while
return $r_{\text{best}}, \mathbf{d}_{\text{best}}$

The choice of $\min(r_{\text{best}}, \|\mathbf{d}\|_2)$ is because initial r_{best} is ∞, in the case where the fast check passes, we should make sure the binary search range is finite.

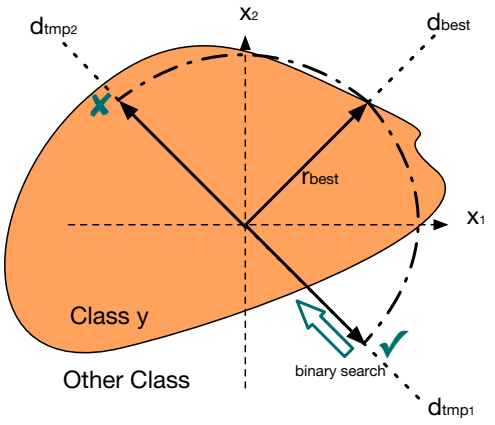


Figure 1: A two-dimensional sketch for the fast check step in Algorithm 2.

3.3 Hierarchical Search

Recent works on black-box attacks [20, 28] found that there exists some spatial correlation between different dimensions of the gradients, and exploiting this prior could help improve the efficiency of black-box attacks. Therefore, they added the same perturbation for small tiles or image blocks on the original data example to achieve better efficiency. Inspired by this finding, we also exploit these spatial correlations by designing a hierarchical search version of the Ray Searching attack, displayed in Algorithm 3. Specifically, we add a new stage variable s. At each stage, we cut the current search direction into 2^s small blocks, and for each iteration, change the sign of the entire block simultaneously as the modified ray search direction for decision boundary radius search. After iterating through

Algorithm 2 Decision Boundary Radius Search (DBR-Search)

input: Model f, Original data example $\{\mathbf{x}, y\}$, Search direction $\mathbf{d}$, Current best radius r_{best}, Binary search tolerance ϵ;
Normalized search direction $\mathbf{d}_n = \mathbf{d}/\|\mathbf{d}\|_2$
if $f(\mathbf{x} + r_{\text{best}} \cdot \mathbf{d}_n) == y$ **then**
  return ∞
end if
Set $start = 0, end = \min(r_{\text{best}}, \|\mathbf{d}\|_2)$
while $end - start > \epsilon$ **do**
  $mid = (start + end)/2$
  if $f(x + mid \cdot \mathbf{d}_n) == y$ **then**
    $end = mid$
  else
    $start = mid$
  end if
end while
return end

Algorithm 3 Ray Searching Attack (Hierarchical)

input: Model f, Original data example $\{\mathbf{x}, y\}$;
Initialize current best search direction $\mathbf{d}_{\text{best}} = (1, \ldots, 1)$
Initialize current best radius $r_{\text{best}} = \infty$
Initialize stage $s = 0$
Initialize block index $k = 1$
while remaining query budget > 0 **do**
  $\mathbf{d}_{\text{tmp}} = \mathbf{d}_{\text{best}}.copy()$
  Cut $\mathbf{d}_{\text{tmp}}$ into 2^s blocks and denote index set in the k-th block by $\mathcal{I}_k$
  $\mathbf{d}_{\text{tmp}}[\mathcal{I}_k] = -\mathbf{d}_{\text{tmp}}[\mathcal{I}_k]$
  $r_{\text{tmp}} = \text{DBR-Search}(f, \mathbf{x}, y, \mathbf{d}_{\text{tmp}}, r_{\text{best}})$
  if $r_{\text{tmp}} < r_{\text{best}}$ **then**
    $r_{\text{best}}, \mathbf{d}_{\text{best}} = r_{\text{tmp}}, \mathbf{d}_{\text{tmp}}$
  end if
  $k = k + 1$
  if $k == 2^s$ **then**
    $s = s + 1$
    $k = 1$
  end if
end while
return $r_{\text{best}}, \mathbf{d}_{\text{best}}$

all blocks we move to the next stage and repeat the search process. Empirically speaking, Algorithm 3 largely improves the search efficiency by exploiting the spatial correlation mentioned above. All our experiments in Section 4 are conducted using Algorithm 3. Note that if the query budget is large enough, Algorithm 3 will, in the end, get to the case where the block size[3] equals to 1 and reduce to Algorithm 1 eventually.

Note that all three algorithms (Algorithms 1, 2 and 3) do not involve any hyperparameters aside from the maximum number of queries, which is usually a predefined problem-related parameter. In sharp contrast, typical white-box attacks and zeroth-order

[3]For completeness, when 2^s is larger than data dimension d, Algorithm 3 will only partition the search direction vector $\mathbf{d}_{\text{tmp}}$ into d blocks to ensure each block contain at least one dimension.

optimization-based black-box attacks, need to tune quite a few hyperparameters in order to achieve good attack performance.

4 EXPERIMENTS

In this section, we present the experimental results of our proposed Ray Searching attack (RayS). We first test RayS attack with other hard-label attack baselines on naturally trained models and then apply RayS attack on recently proposed state-of-the-art robust training models to test their performances. All of our experiments are conducted with NVIDIA 2080 Ti GPUs using Pytorch 1.3.1 on Python 3.6.9 platform.

4.1 Datasets and Target Models

We compare the performance of all attack algorithms on MNIST [24], CIFAR-10 [22] and ImageNet [11] datasets. Following adversarial examples literature [1, 19, 28], we set $\epsilon = 0.3$ for MNIST dataset, $\epsilon = 0.031$ for CIFAR-10 dataset and $\epsilon = 0.05$ for ImageNet dataset. For naturally trained models, on the MNIST dataset, we attack two pre-trained 7-layer CNN: 4 convolutional layers followed by 3 fully connected layers with Max-pooling and RelU activation applied after each convolutional layer. The MNIST pre-trained model achieves 99.5% accuracy on the test set. On the CIFAR-10 dataset, we also use a 7-layer CNN structure with 4 convolutional layers and an additional 3 fully connected layers accompanied by Batch-norm and Max-pooling layers. The CIFAR-10 pre-trained model achieves 82.5% accuracy on the test set. For ImageNet experiments, we attack pre-trained ResNet-50 model [16] and Inception V3 model [37]. The pre-trained ResNet-50 model is reported to have a 76.2% top-1 accuracy. The pre-trained Inception V3 model is reported to have a 78.0% top-1 accuracy. For robust training models, we evaluate two well-recognized defenses: Adversarial Training (AdvTraining) [27] and TRADES [46]. In addition, we also test three other recently proposed defenses which claim to achieve the state-of-the-art robust accuracy: Sensible Adversarial Training (SENSE) [21], Feature Scattering-based Adversarial Training (FeatureScattering) [44], Adversarial Interpolation Training (AdvInterpTraining) [45]. Specifically, adversarial training [27] solves a min-max optimization problem to minimize the adversarial loss. Zhang et al. [46] studied the trade-off between robustness and accuracy in adversarial training and proposed an empirically more robust model. Kim and Wang [21] proposed to stop the attack generation when a valid attack has been found. Zhang and Wang [44] proposed an unsupervised feature-scattering scheme for attack generation. Zhang and Xu [45] proposed an adversarial interpolation scheme for generating adversarial examples as well as adversarial labels and trained on those example-label pairs.

4.2 Baseline Methods

We compare the proposed algorithm with several state-of-the-art attack algorithms. Specifically, for attacking naturally trained models, we compare the proposed RayS attack with other hard-label attack baselines (i) OPT attack [9], (ii) SignOPT attack [10], and (iii) HSJA attack [6]. We adopt the same hyperparameter settings in the original papers of OPT, SignOPT, and HSJA attack.

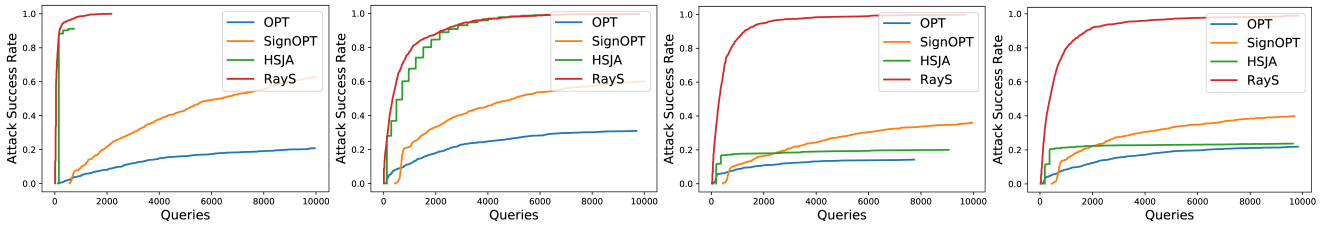
For attacking robust training models, we additionally compare with other state-of-the-art black-box attacks and even white-box attacks: (i) PGD attack [27] (white-box), (ii) CW attack [5] [4] (white-box), (iii) SignHunter [1] (black-box), and (iv) Square attack [2] (black-box). For PGD attack and CW attack, we set step size as 0.007 and provide attack results for 20 steps and also 100 steps. For SignHunter and Square attack, we adopt the same hyperparameter settings used in their original papers.

4.3 Comparison with hard-label Attack Baselines on Naturally Trained Models

In this subsection, we compare our Ray Searching attack with other hard-label attack baselines on naturally trained models. For each dataset (MNIST, CIFAR-10, and ImageNet), we randomly choose 1000 images from its test set that are verified to be correctly classified by the pre-trained model and test how many of them can be successfully attacked by the hard-label attacks. For each method, we restrict the maximum number of queries as 10000. For the sake of query efficiency, we stop the attack for certain test sample once it is successfully attacked, i.e., the L_∞ norm distance between adversarial examples and original examples is less than the pre-defined perturbation limit ϵ. Tables 1, 2, 3 and 4 present the performance comparison of all hard-label attacks on MNIST model, CIFAR-10 model, ResNet-50 Model and Inception V3 model respectively. For each experiment, we report the average and median of the number of queries needed for successful attacks for each attack, as well as the final attack success rate, i.e., the ratio of successful attacks against the total number of attack attempts. Specifically, on the MNIST dataset, we observe that our proposed RayS attack enjoys much better query efficiency in terms of average and median of the number of queries, and much higher attack success rate than OPT and SignOPT methods. Note that the average (median) number of queries of SignOPT is larger than that of OPT. However, this does not mean that SignOPT performs worse than OPT. This result is due to the fact that the attack success rate of OPT is very low and its average (median) queries number is calculated based on the successfully attacked examples, which in this case, are the most vulnerable examples. HSJA attack, though improving over SignOPT[5], still falls behind our RayS attack. For the CIFAR model, the RayS attack still achieves the highest attack success rate. Though the HSJA attack comes close to the RayS attack in terms of attack success rate, its query efficiency still falls behind. On ResNet-50 and Inception V3 models, only RayS attack maintains the high attack success rate while the other baselines largely fall behind. Note that HSJA attack achieves similar or even slightly better average (median) queries on ImageNet models, suggesting that HSJA is efficient for the most vulnerable examples but not very effective when dealing with hard-to-attack examples. Figure 2 shows the attack success rate against the number of queries plot for all baseline methods on different models. Again we can see that the RayS attack overall achieves the highest attack success rate and best query efficiency compared with other hard-label attack baselines.

[4] To be precise, here CW attack refers to PGD updates with CW loss [5]

[5] Note that the relatively weak performance of SignOPT is due to the fact that SignOPT is designed for L_2 norm attack while this experiment is under the L_∞ norm setting. So the result does not conflict with the result reported in the original paper of SignOPT [10].



(a) MNIST (b) CIFAR (c) ResNet-50 (d) Inception V3

Figure 2: Attack success rate against the number of queries plots for different hard-label attacks on MNIST, CIFAR-10 and ImageNet datasets.

Table 1: Comparison of L_∞ norm based hard-label attack on MNIST dataset ($\epsilon = 0.3$).

Methods	Avg. Queries	Med. Queries	ASR (%)
OPT	3260.9	2617.0	20.9
SignOPT	3784.3	3187.5	62.8
HSJA	161.6	154.0	91.2
RayS	**107.0**	**47.0**	**100.0**
PGD (white-box)	-	-	100.0

Table 2: Comparison of L_∞ norm based hard-label attack on CIFAR-10 dataset ($\epsilon = 0.031$).

Methods	Avg. Queries	Med. Queries	ASR (%)
OPT	2253.3	1531.0	31.0
SignOPT	2601.3	1649.0	60.1
HSJA	1021.6	714.0	99.7
RayS	**792.8**	**343.5**	**99.8**
PGD (white-box)	-	-	100.0

Table 3: Comparison of L_∞ norm based hard-label attack on ImageNet dataset for ResNet-50 model ($\epsilon = 0.05$).

Methods	Avg. Queries	Med. Queries	ASR (%)
OPT	1344.5	655.5	14.2
SignOPT	3103.5	2434.0	36.0
HSJA	749.6	**183.0**	19.9
RayS	**574.0**	296.0	**99.8**
PGD (white-box)	-	-	100.0

4.4 Evaluating the Robustness of State-of-the-art Robust Models

In this subsection, we further test our proposed Ray Searching attack by applying it to the state-of-the-art robust training models. Specifically, we selected five recently proposed open-sourced defenses on CIFAR-10 dataset and WideResNet [43] architecture. For the test examples, we randomly choose 1000 images from the CIFAR-10 test set. We set the maximum number of queries as 40000.

Table 4: Comparison of L_∞ norm based hard-label attack on ImageNet dataset for Inception V3 model ($\epsilon = 0.05$).

Methods	Avg. Queries	Med. Queries	ASR (%)
OPT	2375.6	1674.0	21.9
SignOPT	2624.8	1625.0	39.9
HSJA	**652.3**	**362.0**	23.7
RayS	748.2	370.0	**98.9**
PGD (white-box)	-	-	100.0

In terms of evaluation metrics, following the literature of robust training [27, 46], we report the natural accuracy and robust accuracy (classification accuracy under adversarial attacks) of the defense model. In addition, we report a new metric called *Average Decision Boundary Distance* (ADBD), which is defined as the average L_∞ norm distance between *all* test examples to their nearest decision boundaries. Note that ADBD is not valid for white-box and black-box attacks that follow formulation (3.1), since they cannot find the nearest decision boundaries for all test examples.

Here we want to emphasize the difference between ADBD and the average L_∞ distortion in the adversarial learning literature. Note that L_∞ distortion[6] usually refers to the L_∞ norm distance between successful adversarial attack examples and their corresponding original clean examples and therefore, is affected by the choice the maximum perturbation limit ϵ. For hard-label attacks, only considering the attacks with a radius less than ϵ loses too much information and cannot capture the whole picture of model robustness[7]. On the other hand, the ADBD metric, though only valid for hard-label attacks, provides a meaningful estimation on the average distance from the original clean examples to their decision boundaries.

Tables 5, 6, 7, 8 and 9 show the comparison of different adversarial attack methods on five selected robust models. Specifically, for two well recognized robust training models, Adversarial Training (in Table 5) and TRADES (in Table 6), we observe that white-box attacks are still the strongest attacks, where PGD attack and CW attack achieve very similar attack performances. For black-box attacks,

[6]For all white-box and black-box attacks tested in this experiment, their L_∞ distortions are very close to 0.031, which is the perturbation limit ϵ. Therefore, we do not report the L_∞ distortion in the tables as it does not provide much additional information.
[7]For hard-label attacks, the ADBD value is always larger than the L_∞ distortion.

the SignHunter attack and Square attack achieve similar attack performances as their white-box counterparts. In terms of hard-label attacks, our proposed RayS attack also achieves comparable attack performance as black-box or even white-box attacks given the most restricted access to the target model. When comparing with other hard-label attack baselines, it can be seen that our RayS attack achieves significant performance improvement in terms of both robust accuracy (over 20%) and the average decision boundary distance (reduced by 30%). The less effectiveness in attacking L_∞ norm threat model makes the SignOPT attack and HSJA attack less practical. For Sensible Adversarial Training model (in Table 7), it indeed achieves overall better robust accuracy under white-box attacks, compared with Adversarial Training and TRADES. For black-box attacks, the SignHunter attack achieves similar performance as PGD attack and Square attack achieves similar performance as CW attacks. Interestingly, we observe that for hard-label attacks, our proposed RayS attack achieves 42.5% robust accuracy, reducing 20% from PGD attack and 15% from CW attack, suggesting that the robustness of Sensible Adversarial Training is not truly better than TRADES and Adversarial Training, but just looks better under PGD attack and CW attack. For Feature Scattering-based Adversarial Training model (in Table 8), note that the CW attack is much more effective than the PGD attack. Also for black-box attacks, the performance of the Square attack is much better than SignHunter attack[8], suggesting that the CW loss is more effective than CrossEntropy loss in attacking Feature Scattering-based Adversarial Training model. Again, we can observe that our proposed RayS attack reduces the robust accuracy of PGD attack by 28% and CW attack by 10%. This also suggests that Feature Scattering-based Adversarial Training model does not really provide better robustness than Adversarial Training or TRADES. For Adversarial Interpolation Training model (in Table 9), under white-box attacks, it achieves surprisingly high robust accuracy of 75.3% (under PGD attack) and 68.9% (under CW attack), and similar results can be obtained under the corresponding black-box attacks. However, it is still not truly robust under our RayS attack, reducing the robust accuracy of PGD attack by 28% and CW attack by 22%. Note that in this experiment, the HSJA attack also achieves lower robust accuracy than PGD attack, suggesting that all hard-label attacks may have the potential to detect those falsely robust models that deceive current white-box/black-box attacks, but the low efficiency of HSJA restricts its power for greater use.

To obtain the overall comparison on the robustness of the five selected robust training models under our proposed RayS attack, we plot the Average Decision Boundary Distance (ADBD) against RayS attack iterations in Figure 3 and the robust accuracy against RayS attack iterations in Figure 4. First, it can be seen that the Average Decision Boundary Distance and robust accuracy indeed converge and remain stable after around 10000 RayS attack iterations. Figures 3 and 4 suggest that among the five selected robust training models, TRADES and Adversarial Training remain the most robust models while Sensible Adversarial Training, Feature Scattering-based Adversarial Training and Adversarial Interpolation Training, are not as robust as they appear under PGD attacked and CW attack.

[8]Square attack is based on CW loss while SignHunter attack is based on CrossEntropy loss.

Table 5: Comparison of different adversarial attack methods on Adversarial Training [27] for CIFAR-10 dataset (WideResNet, $\epsilon = 0.031$, natural accuracy: 87.4%).

Methods	Att. Type	ADBD	Rob. Acc (%)
SignOPT	hard-label	0.202	85.1
HSJA	hard-label	0.060	76.8
RayS	hard-label	**0.038**	**54.0**
SignHunter	black-box	-	**50.9**
Square	black-box	-	52.7
PGD-20	white-box	-	51.1
CW-20	white-box	-	51.8
PGD-100	white-box	-	**50.6**
CW-100	white-box	-	51.5

Table 6: Comparison of different adversarial attack methods on TRADES [46] for CIFAR-10 dataset (WideResNet, $\epsilon = 0.031$, natural accuracy: 85.4%).

Methods	Att. Type	ADBD	Rob. Acc (%)
SignOPT	hard-label	0.196	84.0
HSJA	hard-label	0.064	71.6
RayS	hard-label	**0.040**	**57.3**
SignHunter	black-box	-	**56.1**
Square	black-box	-	**56.1**
PGD-20	white-box	-	56.5
CW-20	white-box	-	55.6
PGD-100	white-box	-	56.3
CW-100	white-box	-	**55.3**

Table 7: Comparison of different adversarial attack methods on SENSE [21] for CIFAR-10 dataset (WideResNet, $\epsilon = 0.031$, natural accuracy: 91.9%).

Methods	Att. Type	ADBD	Rob. Acc (%)
SignOPT	hard-label	0.170	88.2
HSJA	hard-label	0.044	66.6
RayS	hard-label	**0.029**	**42.5**
SignHunter	black-box	-	61.9
Square	black-box	-	**58.2**
PGD-20	white-box	-	62.1
CW-20	white-box	-	59.7
PGD-100	white-box	-	60.1
CW-100	white-box	-	**57.9**

Note also that even though Sensible Adversarial Training, Feature Scattering-based Adversarial Training and Adversarial Interpolation Training have quite different robust accuracy results under RayS attack, their ADBD results are quite similar.

5 DISCUSSIONS AND CONCLUSIONS

In this paper, we proposed the Ray Searching attack, which only requires the hard-label output of the target model. The proposed

Table 8: Comparison of different adversarial attack methods on Feature-Scattering [44] for CIFAR-10 dataset (WideResNet, $\epsilon = 0.031$, natural accuracy: 91.3%).

Methods	Att. Type	ADBD	Rob. Acc (%)
SignOPT	hard-label	0.175	87.1
HSJA	hard-label	0.048	70.0
RayS	hard-label	**0.030**	**44.5**
SignHunter	black-box	-	67.3
Square	black-box	-	**55.3**
PGD-20	white-box	-	72.8
CW-20	white-box	-	57.2
PGD-100	white-box	-	70.4
CW-100	white-box	-	**54.8**

Table 9: Comparison of different adversarial attack methods on Adversarial Interpolation Training [45] for CIFAR-10 dataset (WideResNet, $\epsilon = 0.031$, natural accuracy: 91.0%).

Methods	Att. Type	ADBD	Rob. Acc (%)
SignOPT	hard-label	0.169	84.2
HSJA	hard-label	0.049	70.5
RayS	hard-label	**0.031**	**46.9**
SignHunter	black-box	-	73.6
Square	black-box	-	**69.0**
PGD-20	white-box	-	75.6
CW-20	white-box	-	69.2
PGD-100	white-box	-	75.3
CW-100	white-box	-	**68.9**

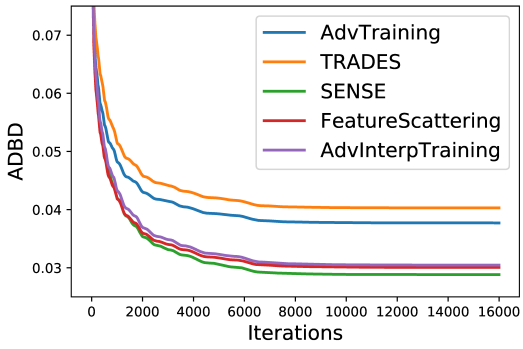


Figure 3: Average Decision Boundary Distance (ADBD) against RayS attack iterations plot for several robust models.

Ray Searching attack is much more effective in attack success rate and efficient in terms of query complexity, compared with other hard-label attacks. Moreover, it can be used as a sanity check tool for possible "falsely robust" models that deceive current white-box and black-box attacks.

In the following discussions, we try to analyze the key ingredients for the success of the proposed Ray Searching attack.

Why RayS attack is more effective and efficient than the other hard-label baselines?

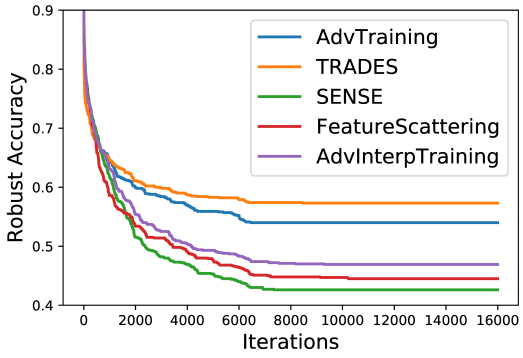


Figure 4: Robust accuracy against RayS attack iterations plot for several robust models.

As we mentioned before, traditional hard-label attacks are more focused on the L_2 norm threat model with only a few extensions to the L_∞ norm threat model. While for our RayS attack, we reformulate the continuous problem of finding the closest decision boundary into a discrete problem based on empirical findings in L_∞ norm threat model, which leads to a more effective hard-label attack. On the other hand, the strategy of directly searching for the closest decision boundary together with a fast check step eliminates unnecessary searches and significantly improves the attack efficiency.

Why RayS attack can detect possible "false" robust models while traditional white-box and black-box attacks cannot?

One thing we observe from Section 4 is that although different attacks lead to different robust accuracy results, their attack performances are correlated with the choice of attack loss functions, e.g., both PGD attack and SignHunter attack utilize CrossEntropy loss and their attack performances are similar in most cases. A similar effect can also be seen for the CW attack and Square attack, both of which utilize the CW loss function. However, these loss functions were used as surrogate losses to problem (3.1), and they may not be able to truly reflect the quality/potential of an intermediate example (an example near the original clean example that is not yet a valid adversarial example). For instance, consider the case where two intermediate examples share the same log probability at ground truth class y, but vary drastically on other classes. Their CrossEntropy losses are the same in such cases, but one may have larger potential to develop into a valid adversarial example than the other one (e.g., the second-largest probability is close to the largest probability). Therefore, CrossEntropy loss does not really reflect the true quality/potential of the intermediate examples. Similar instances can also be constructed for CW loss. In sharp contrast, our RayS attack consider the decision boundary radius as the search criterion[9]. When we compare two examples on the decision boundary, it is clear that the closer one is better. In cases where the attack problem is hard to solve and the attacker could easily get stuck at intermediate examples (e.g., attacking robust training models), it is easy to see that the RayS attack stands a better chance of finding a successful attack. This partially explains the superiority of RayS attack in detecting "falsely robust" models.

[9] Actually it is a criterion for all hard-label attack.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and senior PC for their helpful comments. This research was sponsored in part by the National Science Foundation SaTC-1717950 and CAREER Award 1906169. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] Abdullah Al-Dujaili and Una-May O'Reilly. 2020. Sign Bits Are All You Need for Black-Box Attacks. In *International Conference on Learning Representations*.

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2019. Square Attack: a query-efficient black-box adversarial attack via random search. *arXiv preprint arXiv:1912.00049* (2019).

[3] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *International Conference on Machine Learning*.

[4] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *International Conference on Learning Representations*.

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

[6] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. 2019. Hopskipjumpattack: A query-efficient decision-based attack. *arXiv preprint arXiv:1904.02144* 3 (2019).

[7] Jinghui Chen, Dongruo Zhou, Jinfeng Yi, and Quanquan Gu. 2020. A Frank-Wolfe framework for efficient and effective adversarial attacks. *AAAI* (2020).

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

[9] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, JinFeng Yi, and Cho-Jui Hsieh. 2019. Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach. In *International Conference on Learning Representations*.

[10] Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. 2020. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack. In *International Conference on Learning Representations*.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 248–255.

[12] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. 2018. Stochastic activation pruning for robust adversarial defense. *International Conference on Learning Representations* (2018).

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *International Conference on Learning Representations* (2015).

[14] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. 2018. Countering adversarial images using input transformations. *International Conference on Learning Representations* (2018).

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European Conference on Computer Vision*. Springer, 630–645.

[17] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. 2012. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine* 29 (2012).

[18] Weiwei Hu and Ying Tan. 2017. Generating adversarial malware examples for black-box attacks based on GAN. *arXiv preprint arXiv:1702.05983* (2017).

[19] Andrew Ilyas, Logan Engstrom, Anish Athalye, Jessy Lin, Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Black-box Adversarial Attacks with Limited Queries and Information. In *Proceedings of the 35th International Conference on Machine Learning*.

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

[21] Jungeum Kim and Xiao Wang. 2020. Sensible adversarial learning. https://openreview.net/forum?id=rJlf_RVKwr

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

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).

[24] Yann LeCun, Corinna Cortes, and CJ Burges. 2010. MNIST handwritten digit database. (2010).

[25] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. 2019. NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks. In *International Conference on Machine Learning*. 3866–3876.

[26] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. *International Conference on Learning Representations* (2018).

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

[28] Seungyong Moon, Gaon An, and Hyun Oh Song. 2019. Parsimonious Black-Box Adversarial Attacks via Efficient Combinatorial Optimization. In *International Conference on Machine Learning*. 4636–4645.

[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2574–2582.

[30] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).

[31] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 506–519.

[32] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 372–387.

[33] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 582–597.

[34] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *International Conference on Learning Representations* (2018).

[35] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. 2018. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *International Conference on Learning Representations* (2018).

[36] Ilya Sutskever, Geoffrey E Hinton, and A Krizhevsky. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (2012), 1097–1105.

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

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[39] Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron Oord. 2018. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. In *International Conference on Machine Learning*. 5025–5034.

[40] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. 2019. On the Convergence and Robustness of Adversarial Training. In *International Conference on Machine Learning*. 6586–6595.

[41] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. 2020. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In *International Conference on Learning Representations*.

[42] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2018. Mitigating adversarial effects through randomization. *International Conference on Learning Representations* (2018).

[43] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).

[44] Haichao Zhang and Jianyu Wang. 2019. Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems*. 1829–1839.

[45] Haichao Zhang and Wei Xu. 2020. Adversarial Interpolation Training: A Simple Approach for Improving Model Robustness. https://openreview.net/forum?id=Syejj0NYvr

[46] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *International Conference on Machine Learning*. 7472–7482.