# A Frank-Wolfe Framework for Efficient and Effective Adversarial Attacks

**Jinghui Chen,**[1] **Dongruo Zhou,**[1] **Jinfeng Yi,**[2] **Quanquan Gu**[1]
[1]Department of Computer Science, University of California, Los Angeles
[2]JD AI Research
{jhchen,drzhou,qgu}@cs.ucla.edu, yijinfeng@jd.com

**Abstract**

Depending on how much information an adversary can access to, adversarial attacks can be classified as white-box attack and black-box attack. For white-box attack, optimization-based attack algorithms such as projected gradient descent (PGD) can achieve relatively high attack success rates within moderate iterates. However, they tend to generate adversarial examples near or upon the boundary of the perturbation set, resulting in large distortion. Furthermore, their corresponding black-box attack algorithms also suffer from high query complexities, thereby limiting their practical usefulness. In this paper, we focus on the problem of developing efficient and effective optimization-based adversarial attack algorithms. In particular, we propose a novel adversarial attack framework for both white-box and black-box settings based on a variant of Frank-Wolfe algorithm. We show in theory that the proposed attack algorithms are efficient with an $O(1/\sqrt{T})$ convergence rate. The empirical results of attacking the ImageNet and MNIST datasets also verify the efficiency and effectiveness of the proposed algorithms. More specifically, our proposed algorithms attain the best attack performances in both white-box and black-box attacks among all baselines, and are more time and query efficient than the state-of-the-art.

## 1 Introduction

Deep Neural Networks (DNNs) have made many breakthroughs in different areas of artificial intelligence such as image classification (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016), object detection (Ren et al. 2015; Girshick 2015), and speech recognition (Mohamed et al. 2012; Bahdanau et al. 2016). However, recent studies show that deep neural networks are vulnerable to adversarial examples (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2015) – a tiny perturbation on an image that is almost invisible to human eyes could mislead a well-trained image classifier towards misclassification. Soon later this is proved to be not a coincidence in image classification: similar phenomena have been observed in other problems such as speech recognition (Carlini et al. 2016), visual QA (Xu et al. 2017), image captioning (Chen et al. 2017a), machine translation (Cheng et al. 2018), reinforcement learning (Pattanaik et al. 2018), and even on systems that operate in the physical world (Kurakin, Goodfellow, and Bengio 2016).



Depending on how much information an adversary can access to, adversarial attacks can be classified into two classes: white-box attack (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2015) and black-box attack (Papernot, McDaniel, and Goodfellow 2016; Chen et al. 2017c). In the white-box setting, the adversary has full access to the target model, while in the black-box setting, the adversary can only access the input and output of the target model but not its internal configurations.

Several optimization-based methods have been proposed for the white-box attack. One of the first successful attempt is the FGSM method (Goodfellow, Shlens, and Szegedy 2015), which works by linearizing the network loss function. CW method (Carlini and Wagner 2017) further improves the attack effectiveness by designing a regularized loss function based on the logit-layer output of the network and optimizing the loss by Adam (Kingma and Ba 2015). Even though CW largely improves the effectiveness, it requires a large number of gradient iterations to optimize the distortion of the adversarial examples. Iterative gradient (steepest) descent based methods such as PGD (Madry et al. 2018) and I-FGSM (Kurakin, Goodfellow, and Bengio 2016) can achieve relatively high attack success rates within a moderate number of iterations. However, they tend to generate adversarial examples near or upon the boundary of the perturbation set, due to the projection nature of the algorithm. This leads to large distortion in the resulting adversarial examples.

In the black-box attack, since one needs to make gradient estimations in such setting, a large number of queries are required to perform a successful black-box attack, especially when the data dimension is high. A naive way to estimate gradient direction is to perform finite difference approximation on each dimension (Chen et al. 2017c). This would take $O(d)$ queries to performance one full gradient estimation where $d$ is the data dimension and therefore result in inefficient attacks. For example, attacking a $299 \times 299 \times 3$ ImageNet (Deng et al. 2009) image may take hundreds of thousands of queries. This significantly limits the practical usefulness of such algorithms since they can be easily defeated by limiting the number of queries that an adver-

sary can make to the target model. Although recent studies (Ilyas et al. 2018; Ilyas, Engstrom, and Madry 2018) have improved the query complexity by using Gaussian sensing vectors or gradient priors, due to the inefficiencies of PGD framework, there is still room for improvements.

In this paper, we propose efficient and effective optimization-based adversarial attack algorithms based on a variant of Frank-Wolfe algorithm. We show in theory that the proposed attack algorithms are efficient with guaranteed convergence rate. The empirical results also verify the efficiency and effectiveness of our proposed algorithms.

In summary, we make the following main contributions:

1. We develop a new Frank-Wolfe based projection-free attack framework with momentum mechanism. The framework contains an iterative first-order white-box attack algorithm which admits the fast gradient sign method (FGSM) as a one-step special case, and also a corresponding black-box attack algorithm which adopts zeroth-order optimization with two sensing vector options (either from the Euclidean unit sphere or from the standard Gaussian distribution).
2. We prove that the proposed white-box and black-box attack algorithms with momentum mechanism enjoy an $O(1/\sqrt{T})$ convergence rate in the nonconvex setting. Compared with existing analyses of Frank-Wolfe for nonconvex optimization (Lacoste-Julien 2016; Reddi et al. 2016; Balasubramanian and Ghadimi 2018), we use momentum in our algorithm for both white-box and black-box attacks and therefore our analysis is more involved. To the best of our knowledge, the convergence of Frank-Wolfe with momentum in the nonconvex setting has never been established before, which is of independent interest. We also show that the query complexity of the proposed black-box attack algorithm is linear in data dimension $d$.
3. Our experiments on MNIST and ImageNet datasets show that (i) the proposed white-box attack algorithm has better distortion and is more efficient than all the state-of-the-art white-box attack baseline algorithms, and (ii) the proposed black-box attack algorithm is highly query efficient and achieves the highest attack success rate among other baselines.

The remainder of this paper is organized as follows: in Section 2, we briefly review existing literature on adversarial examples and Frank-Wolfe algorithm. We present our proposed Frank-Wolfe framework in Section 3, and the main theory in Section 4. In Section 5, we compare the proposed algorithms with state-of-the-art adversarial attack algorithms on ImageNet and MNIST datasets. Finally, we conclude this paper in Section 6.

## 2 Related Work

There is a large body of work on adversarial attacks. In this section, we review the most relevant work in both white-box and black-box attack settings, as well as the non-convex Frank-Wolfe optimization.

**White-box Attacks:** (Szegedy et al. 2013) proposed to use box-constrained L-BFGS algorithm for conducting white-box attacks. (Goodfellow, Shlens, and Szegedy 2015) proposed the Fast Gradient Sign Method (FGSM) based on linearization of the network as a simple alternative to L-BFGS. (Kurakin, Goodfellow, and Bengio 2016) proposed to iteratively perform one-step FGSM (Goodfellow, Shlens, and Szegedy 2015) algorithm and clips the adversarial point back to the distortion limit after every iteration. It is called Basic Iterative Method (BIM) or I-FGM in the literature. (Madry et al. 2018) showed that for the $L_\infty$ norm case, BIM/I-FGM is almost[1] equivalent to Projected Gradient Descent (PGD), which is a standard tool for constrained optimization. (Papernot et al. 2016) proposed JSMA to greedily attack the most significant pixel based on the Jacobian-based saliency map. (Moosavi-Dezfooli, Fawzi, and Frossard 2016) proposed attack methods by projecting the data to the closest separating hyperplane. (Carlini and Wagner 2017) introduced the so-called CW attack by proposing multiple new loss functions for generating adversarial examples. (Chen et al. 2017b) followed CW's framework and use an Elastic Net term as the distortion penalty. (Dong et al. 2018) proposed MI-FGSM to boost the attack performances using momentum.

**Black-box Attacks:** One popular family of black-box attacks (Hu and Tan 2017; Papernot, McDaniel, and Goodfellow 2016; Papernot et al. 2017) is based on the transferability of adversarial examples (Liu et al. 2018; Bhagoji et al. 2017), where an adversarial example generated for one DNN may be reused to attack other neural networks. This allows the adversary to construct a substitute model that mimics the targeted DNN, and then attack the constructed substitute model using white-box attack methods. However, this type of attack algorithms usually suffer from large distortions and relatively low success rates (Chen et al. 2017c). To address this issue, (Chen et al. 2017c) proposed the Zeroth-Order Optimization (ZOO) algorithm that extends the CW attack to the black-box setting and uses a zeroth-order optimization approach to conduct the attack. Although ZOO achieves much higher attack success rates than the substitute model-based black-box attacks, it suffers from a poor query complexity since its naive implementation requires to estimate the gradients of all the coordinates (pixels) of the image. To improve its query complexity, several approaches have been proposed. For example, (Tu et al. 2018) introduces an adaptive random gradient estimation algorithm and a well-trained Autoencoder to speed up the attack process. (Ilyas et al. 2018) and (Liu et al. 2018) improved ZOO's query complexity by using Natural Evolutionary Strategies (NES) (Wierstra et al. 2014; Salimans et al. 2017) and active learning, respectively. (Ilyas, Engstrom, and Madry 2018) further improve the performance by considering the gradient priors. (Li et al. 2019) proposed to learn the distributions of adversarial examples to achieve better black-box attack performance. (Moon, An, and Song 2019) re-formulated the black-box attack problem as a discrete surrogate optimiza-

[1] Standard PGD in the optimization literature uses the exact gradient to perform the update step while PGD (Madry et al. 2018) is actually the steepest descent (Boyd and Vandenberghe 2004) with respect to $L_\infty$ norm.

tion problem and used combinatorial search algorithm to improve the query efficiency.

**Non-convex Frank-Wolfe Algorithms:** The Frank-Wolfe algorithm (Frank and Wolfe 1956), also known as the conditional gradient method, is an iterative optimization method for constrained optimization problem. (Jaggi 2013) revisited Frank-Wolfe algorithm in 2013 and provided a stronger and more general convergence analysis in the convex setting. (Yu, Zhang, and Schuurmans 2017) proved the first convergence rate for Frank-Wolfe type algorithm in the non-convex setting. (Lacoste-Julien 2016) provided the convergence guarantee for Frank-Wolfe algorithm in the non-convex setting with adaptive step sizes. (Reddi et al. 2016) further studied the convergence rate of non-convex stochastic Frank-Wolfe algorithm in the finite-sum optimization setting. Very recently, (Staib and Jegelka 2017) proposed to use Frank-Wolfe for distributionally robust training (Sinha, Namkoong, and Duchi 2018). (Balasubramanian and Ghadimi 2018) proved the convergence rate for zeroth-order nonconvex Frank-Wolfe algorithm using one-side finite difference gradient estimator with standard Gaussian sensing vectors.

# 3 Methodology

## 3.1 Notation

Throughout the paper, scalars are denoted by lower case letters, vectors by lower case bold face letters and sets by calligraphy upper cae letters. For a vector $\mathbf{x} \in \mathbb{R}^d$, we denote the $L_p$ norm of $\mathbf{x}$ by $\|\mathbf{x}\|_p = (\sum_{i=1}^d x_i^p)^{1/p}$. Specially, for $p = \infty$, the $L_\infty$ norm of $\mathbf{x}$ by $\|\mathbf{x}\|_\infty = \max_{i=1}^d |\theta_i|$. We denote $\mathcal{P}_\mathcal{X}(\mathbf{x})$ as the projection operation of projecting vector $\mathbf{x}$ into the set $\mathcal{X}$.

## 3.2 Problem Formulation

According to the attack purposes, attacks can be divided into two categories: *untargeted attack* and *targeted attack*.

In particular, untargeted attack aims to turn the prediction into any incorrect label, while the targeted attack, requires to mislead the classifier to a specific target class. In this work, we focus on the strictly harder targeted attack setting (Carlini and Wagner 2017; Ilyas et al. 2018). It is worth noting that our proposed algorithm can be extended to untargeted attack straightforwardly. To be more specific, let us define $\ell(\mathbf{x}, y)$ as the classification loss function of the targeted DNN with an input $\mathbf{x} \in \mathbb{R}^d$ and a corresponding label $y$. For targeted attacks, we aim to minimize $\ell(\mathbf{x}, y_{\text{tar}})$ to learn an adversarial example that will be misclassified to the target class $y_{\text{tar}}$. In the rest of this paper, let $f(\mathbf{x}) = \ell(\mathbf{x}, y_{\text{tar}})$ be the attack loss function for simplicity, and the corresponding targeted attack problem [2] can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \|\mathbf{x} - \mathbf{x}_{\text{ori}}\|_p \le \epsilon. \end{aligned} \tag{3.1}$$

[2]Note that there is usually an additional constraint on the input variable $\mathbf{x}$, e.g., $\mathbf{x} \in [0, 1]^n$ for normalized image inputs.

Evidently, the constraint set $\mathcal{X} := \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_{\text{ori}}\|_p \le \epsilon\}$ is a bounded convex set when $p \ge 1$. Note that even though we mainly focus on the most popular $L_\infty$ attack case in this paper, our proposed methods can easily extend to general $p \ge 1$ case.

## 3.3 Frank-Wolfe vs. PGD

Although PGD can achieve relatively high attack success rate within moderate iterates, the multi-step update formula requires an additional projection step at each iteration to keep the iterates within the constraint set. This tends to cause the generated adversarial examples near or upon the boundary of the constraint set, and leads to relatively large distortion. This motivates us to use Frank-Wolfe based optimization algorithm (Frank and Wolfe 1956). Different from PGD, Frank-Wolfe algorithm is projection-free as it calls a Linear Minimization Oracle (LMO) over the constraint set $\mathcal{X}$ at each iteration, i.e.,

$$\text{LMO} \in \operatorname*{argmin}_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x}, \nabla f(\mathbf{x}_t) \rangle.$$

The LMO can be seen as the minimization of the first-order Taylor expansion of $f(\cdot)$ at point $\mathbf{x}_t$:

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}_t) + \langle \mathbf{x} - \mathbf{x}_t, \nabla f(\mathbf{x}_t) \rangle.$$

By calling LMO, Frank Wolfe solves the linear problem in $\mathcal{X}$ and then perform weighted average with previous iterate to obtain the final update formula.

Comparing the two methods, PGD is a more "aggressive" approach. It first takes a step towards the negative gradient direction while ignoring the constraint to get a new point (often outside the constraint set), and then correct the new point by projecting it back into the constraint set. In sharp contrast, Frank-Wolfe is more "conservative" as it always keeps the iterates within the constraint set. Therefore, it avoids projection and can lead to better distortion.

## 3.4 Frank-Wolfe White-box Attacks

The proposed Frank-Wolfe based white-box attack algorithm is shown in Algorithm 1, which is built upon the classic Frank-Wolfe algorithm. The key difference between Algorithm 1 and the classic Frank-Wolfe algorithm is in Line 4, where an additional momentum term $\mathbf{m}_t$ is introduced. The momentum term $\mathbf{m}_t$ will help stabilize the LMO direction and leads to empirically accelerated convergence of Algorithm 1.

**Algorithm 1** Frank-Wolfe White-box Attack Algorithm

1: **input:** number of iterations $T$, step sizes $\{\gamma_t\}$;
2: $\mathbf{x}_0 = \mathbf{x}_{\text{ori}}, \mathbf{m}_{-1} = \nabla f(\mathbf{x}_0)$
3: **for** $t = 0, \ldots, T-1$ **do**
4: $\quad \mathbf{m}_t = \beta \cdot \mathbf{m}_{t-1} + (1-\beta) \cdot \nabla f(\mathbf{x}_t)$
5: $\quad \mathbf{v}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x}, \mathbf{m}_t \rangle$ // LMO
6: $\quad \mathbf{d}_t = \mathbf{v}_t - \mathbf{x}_t$
7: $\quad \mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_t \mathbf{d}_t$
8: **end for**
9: **output:** $\mathbf{x}_T$

The LMO solution itself can be expensive to obtain in general. Fortunately, for the constraint set $\mathcal{X}$ defined in (3.1), the corresponding LMO has a closed-form solution. Here we provide the closed-form solution of LMO (Line 5 in Algorithm 1) for $L_\infty$ norm case [3]:

$$\mathbf{v}_t = -\epsilon \cdot \text{sign}(\mathbf{m}_t) + \mathbf{x}_{\text{ori}}.$$

Note that if we write down the full update formula at each iteration in Algorithm 1, it becomes

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \epsilon \cdot \text{sign}(\mathbf{m}_t) - \gamma_t(\mathbf{x}_t - \mathbf{x}_{\text{ori}}). \quad (3.2)$$

Intuitively speaking, the term $-\gamma_t(\mathbf{x}_t - \mathbf{x}_{\text{ori}})$ enforces $\mathbf{x}_t$ to be close to $\mathbf{x}_{\text{ori}}$ for all $t = 1, \ldots, T$, which encourages the adversarial example to have a small distortion. This is the key advantage of Algorithm 1.

**Comparison with FGSM:** When $T = 1$, substituting the above LMO solutions into Algorithm 1 yields the final update of $\mathbf{x}_1 = \mathbf{x}_0 - \gamma_t \epsilon \cdot \text{sign}(\nabla f(\mathbf{x}_0))$, which reduces to FGSM [4] when $\gamma_t = 1$. Therefore, our proposed Frank-Wolfe white-box attack also includes FGSM as a one-step special instance.

### 3.5 Frank-Wolfe Black-box Attacks

Next we consider the black-box setting, where we cannot perform back-propagation to calculate the gradient of the loss function anymore. Instead, we can only query the DNN system's outputs with specific inputs. To clarify, here the output refers to the logit layer's output (confidence scores for classification), not the final prediction label.

We propose a zeroth-order Frank-Wolfe based algorithm to solve this problem in Algorithm 2. The key difference between our proposed black-box attack and white-box attack is one extra gradient estimation step, which is presented in Line 4 in Algorithm 2. Also, the momentum term $\mathbf{m}_t$ is now defined as the exponential average of previous gradient estimations $\{\mathbf{q}_t\}_{t=0}^{T-1}$. This will help reduce the variance in zeroth-order gradient estimation and empirically accelerate the convergence of Algorithm 2.

**Algorithm 2** Frank-Wolfe Black-box Attack Algorithm

1: **input:** number of iterations $T$, step sizes $\{\gamma_t\}$, sample size for gradient estimation $b$, sampling parameter $\delta$;
2: $\mathbf{x}_0 = \mathbf{x}_{\text{ori}}$, $\mathbf{m}_{-1} = \text{GRAD\_EST}(\mathbf{x}_0, b, \delta)$
3: **for** $t = 0, \ldots, T-1$ **do**
4: $\quad \mathbf{q}_t = \text{GRAD\_EST}(\mathbf{x}_t, b, \delta)$ // Alg 3
5: $\quad \mathbf{m}_t = \beta \cdot \mathbf{m}_{t-1} + (1-\beta) \cdot \mathbf{q}_t$
6: $\quad \mathbf{v}_t = \text{argmin}_{\mathbf{v} \in \mathcal{X}} \langle \mathbf{v}, \mathbf{m}_t \rangle$
7: $\quad \mathbf{d}_t = \mathbf{v}_t - \mathbf{x}_t$
8: $\quad \mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_t \mathbf{d}_t$
9: **end for**
10: **output:** $\mathbf{x}_T$

[3] The derivation can be found in the Appendix.

[4] The extra clipping operation in FGSM is to project to the additional box constraint for image classification task. We will also need this clipping operation at the end of each iteration for specific tasks such as image classification.

As in many other zeroth-order optimization algorithms (Shamir 2017; Flaxman, Kalai, and McMahan 2005), Algorithm 3 uses symmetric finite differences to estimate the gradient and therefore, gets rid of the dependence on back-propagation in white-box setting. Different from (Chen et al. 2017c), here we do not utilize natural basis as our sensing vectors, instead, we provide two options: one is to use vectors uniformly sampled from Euclidean unit sphere and the other is to use vectors uniformly sampled from standard multivarite Gaussian distribution. This will greatly improve the gradient estimation efficiency comparing to sensing with natural basis as such option will only be able to estimate one coordinate of the gradient vector per query. In practice, both options here provide us competitive experimental results. It is worth noting that NES method (Wierstra et al. 2014) with antithetic sampling (Salimans et al. 2017) used in (Ilyas et al. 2018) yields similar formula as our option II in Algorithm 3.

**Algorithm 3** GRAD_EST($\mathbf{x}, b, \delta$)

$\mathbf{q} = \mathbf{0}$
**for** $i = 1, \ldots, b$ **do**
$\quad$ **option I:** Sample $\mathbf{u}_i$ uniformly from the Euclidean unit sphere with $\|\mathbf{u}_i\|_2 = 1$
$\quad\quad \mathbf{q} = \mathbf{q} + \frac{d}{2\delta b}\big(f(\mathbf{x} + \delta \mathbf{u}_i) - f(\mathbf{x} - \delta \mathbf{u}_i)\big)\mathbf{u}_i$
$\quad$ **option II:** Sample $\mathbf{u}_i$ uniformly from the standard Gaussian distribution $N(\mathbf{0}, \mathbf{I})$
$\quad\quad \mathbf{q} = \mathbf{q} + \frac{1}{2\delta b}\big(f(\mathbf{x} + \delta \mathbf{u}_i) - f(\mathbf{x} - \delta \mathbf{u}_i)\big)\mathbf{u}_i$
**end for**
**return** $\mathbf{q}$

## 4 Main Theory

In this section, we establish the convergence guarantees for our proposed Frank-Wolfe adversarial attack algorithms described in Section 3. The omitted proofs can be found in the Appendix. First, we introduce the convergence criterion for our Frank-Wolfe adversarial attack framework.

### 4.1 Convergence Criterion

The loss function for common DNN models are generally nonconvex. In addition, (3.1) is a constrained optimization. For such general nonconvex constrained optimization, we typically adopt the Frank-Wolfe gap as the convergence criterion (since gradient norm of $f$ is no longer a proper criterion for constrained optimization problems):

$$g(\mathbf{x}_t) = \max_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x} - \mathbf{x}_t, -\nabla f(\mathbf{x}_t) \rangle.$$

Note that we always have $g(\mathbf{x}_t) \geq 0$ and $\mathbf{x}_t$ is a stationary point for the constrained optimization problem if and only if $g(\mathbf{x}_t) = 0$, which makes $g(\mathbf{x}_t)$ a perfect convergence criterion for Frank-Wolfe based algorithms.

### 4.2 Convergence Guarantee for Frank-Wolfe White-box Attack

Before we are going to provide the convergence guarantee of Frank-Wolfe white-box attack (Algorithm 1), we introduce

the following assumptions that are essential to the convergence analysis.

**Assumption 4.1.** Function $f(\cdot)$ is $L$-smooth with respect to $\mathbf{x}$, i.e., for any $\mathbf{x}, \mathbf{x}'$, it holds that

$$f(\mathbf{x}') \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{x}' - \mathbf{x}) + \frac{L}{2}\|\mathbf{x}' - \mathbf{x}\|_2^2.$$

Assumption 4.1 is a standard assumption in nonconvex optimization, and is also adopted in other Frank-Wolfe literature such as (Lacoste-Julien 2016; Reddi et al. 2016). Note that even though the smoothness assumption does not hold for general DNN models, a recent study (Santurkar et al. 2018) shows that batch normalization that is used in many modern DNNs such as Inception V3 model, actually makes the optimization landscape significantly smoother [5]. In addition, recent studies (Allen-Zhu, Li, and Song 2019; Du et al. 2019; Zou et al. 2019) also showed that the loss function of overparameterized deep neural networks is semi-smooth. This justifies the validity of Assumption 4.1.

**Assumption 4.2.** Set $\mathcal{X}$ is bounded with diameter $D$, i.e., $\|\mathbf{x} - \mathbf{x}'\|_2 \leq D$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$.

Assumption 4.2 implies that the input space is bounded. For common tasks such as image classification, given the fact that images have bounded pixel range and $\epsilon$ is a small constant, this assumption trivially holds. Given the above assumptions, the following lemma shows that the momentum term $\mathbf{m}_t$ will not deviate from the gradient direction significantly.

**Lemma 4.3.** Under Assumptions 4.1 and 4.2, for $\mathbf{m}_t$ in Algorithm 1, it holds that

$$\|\nabla f(\mathbf{x}_t) - \mathbf{m}_t\|_2 \leq \frac{\gamma\beta LD}{1-\beta}.$$

Now we present the theorem, which characterizes the convergence rate of our proposed Frank-Wolfe white-box adversarial attack algorithm presented in Algorithm 1.

**Theorem 4.4.** Under Assumptions 4.1 and 4.2, let $\gamma_t = \gamma = \sqrt{2(f(\mathbf{x}_0) - f(\mathbf{x}^*))/(C_\beta LD^2 T)}$, the output of Algorithm 1 satisfies

$$\widetilde{g}_T \leq \sqrt{\frac{2C_\beta LD^2(f(\mathbf{x}_0) - f(\mathbf{x}^*))}{T}},$$

where $\widetilde{g}_T = \min_{1\leq k\leq T} g(\mathbf{x}_k)$, $\mathbf{x}^*$ is the optimal solution to (3.1) and $C_\beta = (1+\beta)/(1-\beta)$.

**Remark 4.5.** Theorem 4.4 suggests that our proposed Frank-Wolfe white-box attack algorithm achieves a $O(1/\sqrt{T})$ rate of convergence. Unlike previous work (Lacoste-Julien 2016) which focuses on the convergence rate of classic Frank-Wolfe method, our analysis shows the convergence rate of the Frank-Wolfe method with momentum mechanism.

[5] The original argument in (Santurkar et al. 2018) refers to the smoothness with respect to each layer's parameters. Note that the first layer's parameters are in the mirror position (in terms of back-propagation) as the network inputs. Therefore, the argument in (Santurkar et al. 2018) can also be applied here with respect to the network inputs.

### 4.3 Convergence Guarantee for Frank-Wolfe Black-box Attack

Next we analyze the convergence of our proposed Frank-Wolfe black-box adversarial attack algorithm presented in Algorithm 2.

In order to prove the convergence of our proposed Frank-Wolfe black-box attack algorithm, we need the following additional assumption that $\|\nabla f(\mathbf{0})\|_2$ is bounded.

**Assumption 4.6.** Gradient of $f(\cdot)$ at zero point $\nabla f(\mathbf{0})$ satisfies $\max_y \|\nabla f(\mathbf{0})\|_2 \leq G$.

Following the analysis in (Shamir 2017), let $f_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u}}[f(\mathbf{x} + \delta\mathbf{u})]$, which is the smoothed version of $f(\mathbf{x})$. This smoothed function value plays a central role in our theoretical analysis, since it bridges the finite difference gradient approximation with the actual gradient. The following lemma shows this relationship.

**Lemma 4.7.** For any $\mathbf{x}$ and the gradient estimator $\mathbf{q}$ of $\nabla f(\mathbf{x})$ in Algorithm 3, its expectation and variance satisfy

$$\mathbb{E}[\mathbf{q}] = \nabla f_\delta(\mathbf{x}),$$
$$\mathbb{E}\|\mathbf{q} - \mathbb{E}[\mathbf{q}]\|_2^2 \leq \frac{1}{b}\left(2d(G + LD)^2 + \frac{1}{2}\delta^2 L^2 d^2\right).$$

And also we have

$$\mathbb{E}\|\nabla f(\mathbf{x}) - \mathbf{q}\|_2 \leq \frac{\delta Ld}{2} + \frac{2\sqrt{d}(G + LD) + \delta Ld}{\sqrt{2b}}.$$

Now we are going to present the theorem, which characterizes the convergence rate of Algorithm 2.

**Theorem 4.8.** Under Assumptions 4.1, 4.2 and 4.6, let $\gamma_t = \gamma = \sqrt{(f(\mathbf{x}_0) - f(\mathbf{x}^*))/(C_\beta LD^2 T)}$, $b = Td$ and $\delta = \sqrt{1/(Td^2)}$, the output of Algorithm 2 satisfies

$$\mathbb{E}[\widetilde{g}_T] \leq \frac{D}{\sqrt{T}}\left(\sqrt{2C_\beta L(f(\mathbf{x}_0) - f(\mathbf{x}^*))} + C_\beta(L + G + LD)\right),$$

where $\widetilde{g}_T = \min_{1\leq k\leq T} g(\mathbf{x}_k)$, the expectation of $\widetilde{g}_T$ is over the randomness of the gradient estimator, $\mathbf{x}^*$ is the optimal solution to (3.1) and $C_\beta = (1+\beta)/(1-\beta)$.

**Remark 4.9.** Theorem 4.8 suggests that Algorithm 2 also enjoys a $O(1/\sqrt{T})$ rate of convergence. Note that (Balasubramanian and Ghadimi 2018) proves the convergence rate for classic zeroth-order Frank-Wolfe algorithm. Our result is different in several aspects. First, we prove the convergence rate of zeroth-order Frank-Wolfe with momentum. Second, we use symmetric finite difference gradient estimator with two types of sensing vectors while they (Balasubramanian and Ghadimi 2018) use one-side finite difference gradient estimator with Gaussian sensing vectors. In terms of query complexity, the total number of queries needed in Algorithm 2 is $Tb = T^2 d$, which is linear in the data dimension $d$. In fact, in the experiment part, we observe that this number can be substantially smaller than $d$, e.g., $b = 25$.

# 5 Experiments

In this section, we present the experimental results for our proposed Frank-Wolfe attack framework against other state-of-the-art adversarial attack algorithms in both white-box and black-box settings. All of our experiments are conducted on Amazon AWS p3.2xlarge servers which come with Intel Xeon E5 CPU and one NVIDIA Tesla V100 GPU (16G RAM). All experiments are implemented in Tensorflow platform version $1.10.0$ within Python $3.6.4$.

## 5.1 Evaluation Setup

We compare the performance of all attack algorithms by evaluating on both MNIST (LeCun 1998) and ImageNet (Deng et al. 2009) datasets. For MNIST dataset, we attack a pre-trained 6-layer CNN: 4 convolutional layers followed by 2 dense layers with max-pooling and Relu activations applied after each convolutional layer. The pre-trained model achieves $99.3\%$ accuracy on MNIST test set. For ImageNet experiments, we attack a pre-trained Inception V3 model (Szegedy et al. 2016). The pre-trained Inception V3 model is reported to have a $78.0\%$ top-1 accuracy and a $93.9\%$ top-5 accuracy. For MNIST dataset, we randomly choose $1000$ images from its test set that are verified to be correctly classified by the pre-trained model and also randomly choose a target class for each image. Similarly, for ImageNet dataset, we randomly choose $250$ images from its validation set as our attack examples. For our proposed black-box attack, we test both options in Algorithm 3. We performed grid search to tune the hyper-parameters for all algorithm to ensure a fair comparison. Detailed description on hyperparameter tuning and parameter settings can be found in the Appendix.

## 5.2 Baseline Methods

We compare the proposed algorithms with several state-of-the-art baseline algorithms. Specifically, we compare the proposed white-box attack algorithm with (i) FGSM (Goodfellow, Shlens, and Szegedy 2015) (ii) PGD (Madry et al. 2018) (normalized steepest descent[6]) (iii) MI-FGSM (Dong et al. 2018). We compare the proposed black-box attack algorithm with (i) NES-PGD attack (Ilyas et al. 2018) and (ii) Bandit attack (Ilyas, Engstrom, and Madry 2018). We did not report the comparison with ZOO (Chen et al. 2017c) here because it consistently underperforms NES-PGD and Bandit attacks according to our experiments and prior work. We also compare with (Li et al. 2019) on attacking the robust model trained by adversarial training.

## 5.3 White-box Attack Experiments

In this subsection, we present the white-box attack experiments on both MNIST and ImageNet datasets. We choose $\epsilon = 0.3$ for MNIST dataset and $\epsilon = 0.05$ for ImageNet dataset. For comparison, we report the attack success rate, average number of iterations to complete the attack, as well as average distortion for each method.

Tables 1 and 2 present our experimental results for the white-box attack experiments. For experiments on both datasets, while FGSM only needs $1$ gradient update per attack, it only achieves $21.5\%$ attack success rate on MNIST and $1.2\%$ attack success rate on ImageNet in the targeted attack setting. All the other methods achieve $100\%$ attack success rate. PGD needs in average $6.2$ and $8.7$ gradient iterations per attack on MNIST and ImageNet respectively. MI-FGSM improves it to around $4.0$ and $5.0$ iterations per attack on MNIST and ImageNet. However, the distortion of both PGD and MI-FGSM is very close to the perturbation limit $\epsilon$, which indicates that their generated adversarial examples are near or upon the boundary of the constraint set. On the other hand, our proposed Frank-Wolfe white-box attack algorithm achieves not only the smallest average number of iterations per attack, but also the smallest distortion among the baselines. This suggests the advantage of Frank-Wolfe based projection-free algorithms for white-box attack.

Table 1: Comparison of targeted $L_\infty$ norm based white-box attacks on MNIST dataset with $\epsilon = 0.3$.

| Methods | ASR(%) | # Iterations | Distortion |
|---|---|---|---|
| FGSM | 21.5 | - | 0.300 |
| PGD | 100.0 | 6.2 | 0.277 |
| MI-FGSM | 100.0 | 4.0 | 0.279 |
| FW-white | 100.0 | **3.3** | **0.256** |

Table 2: Comparison of targeted $L_\infty$ norm based white-box attacks on ImageNet dataset with $\epsilon = 0.05$.

| Methods | ASR(%) | # Iterations | Distortion |
|---|---|---|---|
| FGSM | 1.2 | - | 0.050 |
| PGD | 100.0 | 8.7 | 0.049 |
| MI-FGSM | 100.0 | 5.0 | 0.049 |
| FW-white | 100.0 | **4.8** | **0.019** |

## 5.4 Black-box Attack Experiments

In this subsection, we present the black-box attack experiments on both MNIST and ImageNet datasets. The maximum query limit is set to be $50,000$ per attack. We choose $\epsilon = 0.3$ for MNIST dataset and $\epsilon = 0.05$ for ImageNet dataset. For comparison, we report the attack success rate, average attack time, average number of queries needed, as well as average number of queries needed on successfully attacked samples for each method.

Table 3 presents our experimental results for targeted black-box attacks on both ImageNet and MNIST datasets. We can see that on MNIST, NES-PGD method achieves a relatively high attack success rate, but still takes quite a lot queries per (successful) attack. Bandit method improves the query complexity for successfully attacked samples but has lower attack success rate in this setting and

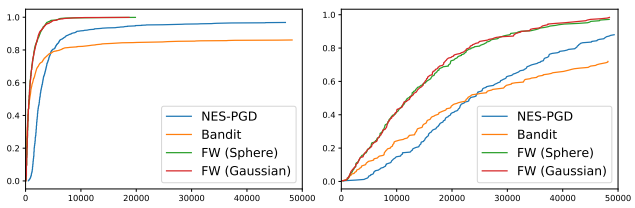
[6] standard PGD will need large step size to go anywhere since the gradient around the true example is relatively small. On the other hand, the large step size will cause the algorithm go out of the constraint set quickly and basically stop moving since then because of the projection step.

Table 3: Comparison of targeted $L_\infty$ norm based black-box attacks on MNIST and ImageNet datasets in terms of attack success rate, average time and average number of queries (QUERIES: for all images including both successfully and unsuccessfully attacked ones; QUERIES(SUCC): for successfully attacked ones only) needed per image.

| METHODS | MNIST ($\epsilon = 0.3$) | | | | IMAGENET ($\epsilon = 0.05$) | | | |
|---|---|---|---|---|---|---|---|---|
| | ASR(%) | TIME(S) | QUERIES | QUERIES(SUCC) | ASR(%) | TIME(S) | QUERIES | QUERIES(SUCC) |
| NES-PGD | 96.8 | 0.2 | 5349.0 | 3871.3 | 88.0 | 85.1 | 26302.8 | 23064.5 |
| BANDIT | 86.1 | 4.8 | 8688.9 | 2019.7 | 72.0 | 148.7 | 27172.5 | 18295.2 |
| FW (SPHERE) | **99.9** | **0.1** | **1132.6** | **1083.6** | 97.2 | 62.1 | 15424.0 | **14430.8** |
| FW (GAUSSIAN) | **99.9** | **0.1** | 1144.4 | 1095.4 | **98.4** | **50.6** | **15099.4** | 14532.3 |

takes longer time to complete the attack. In sharp contrast, our proposed Frank-Wolfe black-box attack algorithms (both sphere and Gaussian sensing vector options) achieve the highest success rate in the targeted black-box attack setting while greatly improve the query complexity by around $50\%$ over the best baseline. On ImageNet, similar patterns can be observed: our proposed Frank-Wolfe black-box attack algorithms achieve the highest attack success rate and further significantly improve the query efficiency against the baselines. This suggests the advantage of Frank-Wolfe based projection-free algorithms for black-box attack.

To provide more intuitive demonstrations, we also plot the attack success rate against the number of queries for our black-box experiments. Figure 1 shows the plots of the attack success rate against the number of queries for different algorithms on MNIST and ImageNet datasets respectively. As we can see from the plots, Bandit attack achieves better query efficiency for easy-to-attack examples (require less queries to attack) compared with NES-PGD or even FW at the early stages, but falls behind even to NES-PGD on hard-to-attack examples (require more queries to attack). We conjecture that in targeted attack setting, the gradient/data priors are not as accurate as in untargeted attack case, which makes Bandit attack less effective especially on hard-to-attack examples. On the other hand, our proposed Frank-Wolfe black-box attack algorithms achieve the highest attack success rate and the best efficiency (least queries needed for achieving the same success rate). This again confirm the advantage of Frank-Wolfe based projection-free algorithms for black-box attack.




(a) MNIST (b) ImageNet

Figure 1: Attack success rate against the number of queries plot for targeted black-box attacks on MNIST and ImageNet datasets.

### 5.5 Experiments on Adversarially Trained Model

In this subsection, we further present the white-box and black-box attack experiments on more challenging robust CIFAR10 model. Specifically, we apply the proposed Frank-Wolfe white-box and black-box attack algorithms to adversarially trained WideResNet model using adversarial training (Madry et al. 2018) . Following (Madry et al. 2018), we choose $\epsilon = 8/255$. For black-box case, the maximum query limit is set to be $20,000$ per attack. Table 4 presents our experimental results for targeted white-box attacks on robust CIFAR10 model. Specifically, in white-box case, the proposed Frank-Wolfe attack achieves $24.3\%$ attack success rate [7] with the smallest $L_\infty$ distortion, while PGD and MI-FGSM can only achieve lower attack success rates and also larger distortions. Table 5 presents our experimental results for targeted black-box attacks on robust CIFAR10 model. In black-box setting, our algorithm achieves $19.0\%$ attack success rate with the smallest overall queries (also relatively small number of queries for successful attempts) while NES needs larger number of queries but achieves only $9.4\%$ attack success rate. Bandit improves the number of average queries needed for successful attempts, yet its attack success rate is only $9.6\%$. Nattack achieves an attack success rate slightly better than Frank-Wolfe but requires the largest number queries for successful attempts.

Table 4: Comparison of targeted $L_\infty$ norm based while-box attacks on adversarially trained WideResNet on CIFAR10 with $\epsilon = 8/255$.

| Methods | ASR(%) | # Iterations | Distortion |
|---|---|---|---|
| FGSM | 21.5 | - | 8.00 |
| PGD | 24.0 | **15.6** | 7.49 |
| MI-FGSM | 24.1 | 15.8 | 7.60 |
| FW-white | **24.3** | 15.8 | **7.48** |

## 6 Conclusions and Future Work

In this work, we propose a Frank-Wolfe framework for efficient and effective adversarial attacks. Our proposed white-box and black-box attack algorithms enjoy an $O(1/\sqrt{T})$

[7] note that it is targeted attack, so the number is much lower than the original paper of (Madry et al. 2018)

Table 5: Comparison of targeted $L_\infty$ norm based black-box attacks on adversarially trained WideResNet on CIFAR10 with $\epsilon = 8/255$ in terms of attack success rate and average number of queries (QUERIES: for all images including both successfully and unsuccessfully attacked ones; QUERIES(SUCC): for successfully attacked ones only) needed per image.

| Methods | ASR(%) | # Queries | Queries(SUCC) |
|---|---|---|---|
| NES-PGD | 9.4 | 18541.1 | 4480.1 |
| Bandit | 9.6 | 18174.2 | **981.5** |
| Nattack | **20.0** | 17135.0 | 5675.0 |
| FW (Opt I) | 19.0 | **16735.2** | 2816.8 |
| FW (Opt II) | 16.8 | 16748.2 | 2703.2 |

rate of convergence, and the query complexity of the proposed black-box attack algorithm is linear in data dimension $d$. Finally, our empirical study on attacking both ImageNet dataset and MNIST dataset yield the best distortion in white-box setting and highest attack success rate/query complexity in black-box setting.

It would also be interesting to see the whether the performance of our Frank-Wolfe adversarial framework can be further improved by incorporating the idea of gradient/data priors (Ilyas, Engstrom, and Madry 2018). We leave it as a future work.

## Acknowledgement

We thank the anonymous reviewers and senior PC for their helpful comments. This research was sponsored in part by the National Science Foundation 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

Allen-Zhu, Z.; Li, Y.; and Song, Z. 2019. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, 242–252.

Bahdanau, D.; Chorowski, J.; Serdyuk, D.; Brakel, P.; and Bengio, Y. 2016. End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 4945–4949. IEEE.

Balasubramanian, K., and Ghadimi, S. 2018. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. *arXiv preprint arXiv:1809.06474*.

Bhagoji, A. N.; He, W.; Li, B.; and Song, D. 2017. Exploring the space of black-box attacks on deep neural networks. *arXiv preprint arXiv:1712.09491*.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Carlini, N., and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.

Carlini, N.; Mishra, P.; Vaidya, T.; Zhang, Y.; Sherr, M.; Shields, C.; Wagner, D.; and Zhou, W. 2016. Hidden voice commands. In *USENIX Security Symposium*, 513–530.

Chen, H.; Zhang, H.; Chen, P.-Y.; Yi, J.; and Hsieh, C.-J. 2017a. Show-and-fool: Crafting adversarial examples for neural image captioning. *arXiv preprint arXiv:1712.02051*.

Chen, P.-Y.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C.-J. 2017b. Ead: elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114*.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017c. 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*, 15–26. ACM.

Cheng, M.; Yi, J.; Zhang, H.; Chen, P.-Y.; and Hsieh, C.-J. 2018. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*.

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

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.

Du, S.; Lee, J.; Li, H.; Wang, L.; and Zhai, X. 2019. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, 1675–1685.

Flaxman, A. D.; Kalai, A. T.; and McMahan, H. B. 2005. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, 385–394. Society for Industrial and Applied Mathematics.

Frank, M., and Wolfe, P. 1956. An algorithm for quadratic programming. *Naval research logistics quarterly* 3(1-2):95–110.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hu, W., and Tan, Y. 2017. Generating adversarial malware examples for black-box attacks based on gan. *arXiv preprint arXiv:1702.05983*.

Ilyas, A.; Engstrom, L.; Athalye, A.; Lin, J.; Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2018. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning,{ICML} 2018*.

Ilyas, A.; Engstrom, L.; and Madry, A. 2018. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*.

Jaggi, M. 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, 427–435.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Lacoste-Julien, S. 2016. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*.

LeCun, Y. 1998. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*.

Li, Y.; Li, L.; Wang, L.; Zhang, T.; and Gong, B. 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.

Liu, Y.; Chen, X.; Liu, C.; and Song, D. 2018. Delving into transferable adversarial examples and black-box attacks. *International Conference on Data Mining (ICDM)*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*.

Mohamed, A.-r.; Dahl, G. E.; Hinton, G.; et al. 2012. Acoustic modeling using deep belief networks. *IEEE Trans. Audio, Speech & Language Processing* 20(1):14–22.

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.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 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.

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, 372–387. IEEE.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 506–519. ACM.

Papernot, N.; McDaniel, P.; and Goodfellow, I. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

Pattanaik, A.; Tang, Z.; Liu, S.; Bommannan, G.; and Chowdhary, G. 2018. Robust deep reinforcement learning with adversarial attacks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2040–2042. International Foundation for Autonomous Agents and Multiagent Systems.

Reddi, S. J.; Sra, S.; Póczos, B.; and Smola, A. 2016. Stochastic frank-wolfe methods for nonconvex optimization. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, 1244–1251. IEEE.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; and Sutskever, I. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.

Santurkar, S.; Tsipras, D.; Ilyas, A.; and Madry, A. 2018. How does batch normalization help optimization?(no, it is not about internal covariate shift). *arXiv preprint arXiv:1805.11604*.

Shamir, O. 2017. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research* 18(52):1–11.

Sinha, A.; Namkoong, H.; and Duchi, J. 2018. Certifying some distributional robustness with principled adversarial training. *International Conference on Learning Representations*.

Staib, M., and Jegelka, S. 2017. Distributionally robust deep learning as a generalization of adversarial training. *Machine Learning and Computer Security Workshop*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

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.

Tu, C.; Ting, P.; Chen, P.; Liu, S.; Zhang, H.; Yi, J.; Hsieh, C.; and Cheng, S. 2018. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *CoRR* abs/1805.11770.

Wierstra, D.; Schaul, T.; Glasmachers, T.; Sun, Y.; Peters, J.; and Schmidhuber, J. 2014. Natural evolution strategies. *The Journal of Machine Learning Research* 15(1):949–980.

Xu, X.; Chen, X.; Liu, C.; Rohrbach, A.; Darell, T.; and Song, D. 2017. Can you fool ai with adversarial examples on a visual turing test? *arXiv preprint arXiv:1709.08693*.

Yu, Y.; Zhang, X.; and Schuurmans, D. 2017. Generalized conditional gradient for sparse estimation. *The Journal of Machine Learning Research* 18(1):5279–5324.

Zou, D.; Cao, Y.; Zhou, D.; and Gu, Q. 2019. Stochastic gradient descent optimizes over-parameterized deep relu networks. *Machine Learning Journal*.